In: Wilhelm, R. (ed.): GI-10. Jahrestagung Saarbrücken. Berlin, Heidelberg, N.Y.: Springer, pp. 459-473.

THE NATURAL LANGUAGE SYSTEM HAM-RPM AS A HOTEL MANAGER: SOME REPRESENTATIONAL PREREQUISITES

Anthony Jameson, Wolfgang Hoeppner, Wolfgang Wahlster

Research Group 'Simulation of Language Understanding'
Universität Hamburg
Germanisches Seminar
Von-Melie-Park 6
D-2000 Hamburg 13
Federal Republic of Germany

ABSTRACT. The dialogue system HAM-RPM simulates a person conversing about one of several interchangeable scenes in colloquial German. Attention is currently being focused on a hotel reservation situation in which the system is required to construct and make use of a model of its dialogue partner's goals and beliefs. This situation calls for a semantic representation language which is declarative and which permits the representation, among other things, of various numerical and natural language quantifiers, conjoined noun phrases and belief modifiers. This paper describes this language and sketches the processes in the course of which expressions of the language are analysed, constructed, transformed, or evaluated. The system's behavior is described in detail to illustrate the relationships among these processes.

1. Introduction

Research on the natural language dialogue system HAM-RPM (Hamburg Dialogue Partner Model) is guided by the principle that many basic phenomena which have traditionally been subjected to purely linguistic or philosophical analysis can be fully understood only if investigated within a larger context which includes, among other things, the beliefs the dialogue partners have about their domain of discourse and about each other, the goals which motivate their participation in the dialogue, and the algorithms and heuristics they employ in attempting to attain these goals (cf. WINOGRAD 1976, BOBROW et al. 1977). The design of a system which simulates a natural dialogue partner is a useful way of approaching such phenomena in that it makes it both necessary and possible to do justice systematically to the roles played by these factors and to their interactions with one another.

One of the relevant factors, the knowledge relevant to the domain about which the dialogue partners converse, has been represented in HAM-RPM to date for three different scenes: two living-room interiors and a traffic scene (cf. v.HAHN et al. 1980). This knowledge is represented in two semantic networks: The referential semantic network contains extensional information concerning the scene being discussed, while the conceptual semantic network contains relevant intensional information. Both of these networks are implemented with the help of FUZZY's assertional data base (LEFAIVRE 1977). On the whole, however, HAM-RPM's knowledge base is heterogeneous, integrating several procedural and declarative representation languages, including production rules, semantic nets, ATN grammars, FUZZY 'Deduce Procedures' and a logic-oriented representation language. The system's knowledge base is distributed over nine files, which contain: a word lexicon, an idiom lexicon, a set of morphological rules, an ATN grammar for syntactic analysis, a coordinate-system representation of spatial data, a set of inference rules, an ATN grammar for generation and the referential and conceptual semantic networks. In addition, the following internal knowledge sources are built up during the dialogue: a record of references to objects, a syntactic and a semantic dialogue memory, an inference memory, and a partner model.

The following excerpt from a dialogue concerning the traffic scene may provide an impression of the behavior of the system. (The system's utterances are typed in capital letters.)

(1) Wo befindet sich das gelbe Auto?
(Where is the yellow car?)
WELCHES GELBE AUTO MEINST DU?
(Which yellow car do you mean?)
Wieso fragst Du mich das?
(Why do you ask?)
WEIL ES HIER DREI GELBE AUTOS GIBT.
(Because there are three yellow cars.)
WELCHES GELBE AUTO MEINTEST DU?
(Which yellow car did you mean?)
Das, das in etwa neben dem Kombi parkt.
(The one parked more or less next to the station wagon.)
VOR DEM FACHBEREICH FÜR INFORMATIK.
(In front of the department of computer science.)

(2) Ist dieses Gebäude groß?
(Is this building large?)
JA.
(Yes.)

(3) Kannst du bitte ausführlicher antworten! Was steht neben der Ampel, die rot ist?
(Could you please give more detailed answers! What's next to the traffic light which is red?)
DAS GRÜNE AUTO UND DER GROSSE BAUM.
(The green car and the yellow tree.)

The three domains of discourse mentioned above have permitted the investigation of a variety of interesting phenomena, but they have the disadvantage that it is unclear what motivation the user might have for asking questions about such a domain; this means that the system in turn can have no other overall goal than the usual goal of question-answering systems, namely that of providing accurate information in an easily understandable form. But participants in natural dialogues often have goals of a quite different sort, and their behavior is determined to a large degree both by their own goals and by the perceived goals of their partner. So that these factors could be studied in a natural manner within HAM-RPM, a fourth domain of discourse has been introduced: The system takes the role of a hotel manager who is talking on the telephone with a caller who is interested in renting a room. The caller is assumed to have the overall goal of determining whether the room offered meets his requirements, which are not known to the manager in advance and which may be quite vague. The hotel manager is assumed to want to present the room in question in a particularly favorable light without saying anything which might later be revealed to be an outright untruth. The system must thus attempt to recognize the caller's specific desires concerning the room, as they are revealed -- usually indirectly -- in his utterances; to recognize when a specific question is asked, what desire of the caller's underlies it; and to make use of the various devices offered by the natural language which permit a particular state of affairs to be presented in a way which appears favorable with respect to a particular desired state of affairs.

The core of the dialogues simulated in this hotel domain consists of exchanges in which the caller expresses, directly or indirectly, some desire concerning the hotel room and the manager attempts to convince him that this wish would in fact be fulfilled. The beginning and the end of the telephone call deviate from this pattern, consisting largely of standard phrases and amenities. These parts of the dialogue are to be handled by an analysis based on a relatively simple dialogue grammar, though even here some important information about the caller may be obtained, for example if he introduces himself as a senior manager.

The dialogue situation described requires that the semantic representation language used by the system be declarative in nature, since it is not only questions which must be represented (other semantic representation languages for German are KS in PLIDIS (cf. ZIFONUN 1979) and MSRL in BACON (cf. HABEL et al. 1978)). For example, the system must generate more complex answers if it is to attempt to sell the hotel room than it would have to if it merely attempted to provide adequate factual answers (cf. COHEN 1978). Such answers cannot be generated directly in natural language, but must first be represented in an internal language. A declarative semantic representation language is also required for the model of the beliefs and desires of the caller which the system builds up in the course of the dialogue (cf. RICH 1979).

In addition to requiring a declarative semantic representation language the hotel situation poses several more specific problems for the representation language and the associated processes, problems which are not faced by more typical question-answering systems:

Numerical quantifiers (cf. SHAPIRO 1979) and other *natural language quantifiers* occur frequently in such dialogues, as do utterances expressing the *goodness* of fit between a quantifier and a given state of affairs, e.g.:

(4) *Caller:* Are all of the pictures oil-paintings?
Manager: Well, not quite, but most of them are.
One or two are etchings.

The main challenge here lies in the fact that the differences in the ways in which such quantifiers are used are often subtle and difficult to express within a single general schema. For example, consider the conditions under which it might be appropriate to give an unqualified affirmative answer to each of the following questions:

(5) *Caller:* Are all of the pictures framed?
(6) *Caller:* Is each of the pictures framed?

In (5) the number of pictures in the room would have to be at least three, since in colloquial English 'all' is not normally applied to sets consisting of two or fewer objects. In (6) the manager would have to have specific knowledge concerning each of the pictures, and not, e.g., merely general knowledge of the policy of the hotel concerning pictures.

Conjunctions of noun phrases are often called for in answers to questions about the room, as in example (7). The fact that such conjunctions are seldom used in questions themselves may incidentally explain why they have received little attention in the AI systems developed to date.

(7) *Manager:* One of the tables and all of the chairs are antiques.

A general treatment of such noun phrases is complicated by the fact that, as this example shows, the conjoined noun phrases may contain different quantifiers.

Particles which reveal the beliefs and desires of the speaker concerning the room represent for the hotel manager a valuable means of recognizing and influencing the caller's beliefs concerning the room (cf. example (8)).

(8) *Caller:* Does the room by any chance have a colour TV?
(9) *Manager:* Yes, of course it does.

Since such phenomena as these could not be handled in a general way by the representation language used in connection with HAM-RPM's first three domains of discourse -- and in fact have attracted little attention within other natural language systems -- a new language has been developed, together with procedures for constructing, analysing, transforming, and evaluating expressions in it. The main purpose of the present paper is to describe the distinguishing features of this language and to sketch the associated processes (section 2). The relationships among these processes and their role within the system as a whole will be illustrated in section 3 with a detailed desription of one example of the system's behavior.

2. Semantic Representation Language: Definition and Interpretation

2.1 Syntactic Definition

The definition of the new semantic representation language in extended BNF notation is given in Fig. 1, in which the nonterminals of the language are listed in alphabetical order. The periods and the one-word comments preceded by a semicolon are not part of the syntactic definition, but are used internally by the system to make the manipulations of language structures more abstract and efficient. Capitalized symbols without the semicolon are terminal elements of the language. Where the right-hand side of an expansion contains only terminal elements, the list of symbols is subject to extension. The two most important elements are the top-level 'FORMULA', onto which, roughly speaking, sentences and clauses are mapped, and 'TERM', onto which both noun phrases and predicates are mapped. 2.2 The Sublanguages SURF and DEEP.

```
<arc>                   ::=   <arc2>
                        ::=   <arc3>
<arc2>                  ::=   ISA | REF | MODUS | RELATOR | AGENS | OBJEKTIV | INSTRUMENTAL |
                              BENEFAKTIV | PROFORM
<arc3>                  ::=   ZEIT | RAEUMLICH
<atomic-formula>        ::=   (<arc2> <term>. ;TERM1 <term>. ;TERM2)                                    ;Arc2
                        ::=   (<arc3> <term>. ;TERM1 <term>. ;TERM2 <term>. ;TERM3)                     ;Arc3
                        ::=   (<arc> <term>.2-3 ;TERMS)                                                  ;Arc
                        ::=   (<d-arc> <term>. <description>.)                                         ;D-Arc
                        ::=   (<f-arc> <term>. <formula>.)                                             ;F-Arc
<constant>              ::=   (*R ?C (ATOM !C))
<d-arc>                 ::=   IS
<description>           ::=   (D: <operator2><description>. ;DESCRIPTION1 <description>.;DESCRIPTION2)  ;Operator2
                        ::=   (D: <operator1><description>.)                                     ;Operator1
                        ::=   (D: <operator><description>. 1-2 ; DESCRIPTIONS)                   ;Operator
                        ::=   (LAMBDA: <variable>. <formula>.)                                   ;Lambda:
<f-arc>                 ::=   SAG | GLAUB | HOFF
<formula>               ::=   (F: <term> <formula> ;PREDICATE)                                        ;Deep
                        ::=   (F: <operator1> <formula>.)                                        ;Operator1
                        ::=   (F: <operator2> <formula>.;FORMULA1 <formula>. ;FORMULA2)          ;Operator2
                        ::=   (F: <operator> <formula>. 1-2 ;FORMULAS)                            ;Operator
                        ::=   <atomic-formula>                                                    ;Atomic
                        ::=   T                                                                     ;True
<hedge>                 ::=   ETWA | GENAU | FAST | UNGEFAEHR | DIREKT | IN/ ETUA | EINIGERMASSEN
<number>                ::=   (*R ?N NUMBERP !N))
                        ::=   T                                                                 ;Infinite
<number-word>           ::=   <number>
                        ::=   VIEL | EINIG | MEHRER | ETLICH | WENIG | E- | KEIN
<operator>              ::=   <operator1>
                        ::=   <operator2
<operator1>             ::=   NOT | LEIDER | SELBSTVERSTAENDLICH | MOEGLICH | WUENSCHENSWERT
<operator2>             ::=   AND | OR
<quantifier>            ::=   (FOR: <quantifier-operator>.<variable>.)                                ;Deep
<quantifier—operator>
                        ::=   (Q: <quantifier-word>)
                        ::=   (Q: <quantifier-word> <quantity>. ;PRESUPPOSITION?)          ;PRESUPPOSITION?
                        ::=   (Q: <quantity>)                                                   ;Quantity
                        ::=   (Q: <constant>)                                                   ;Constant
<quantifier-word>       ::=   (<hedge> <quantifier-word>)                                          ;Hedge
                        ::=   ALL | MEIST | WELCH | WIEVIEL | D- | JED | DIES | JEN | BEID
<quantity>              ::=   (<hedge> <number-word>)                                              ;Hedge
                        ::=   <number-word>
                        ::=   <range>
<range>                 ::=   (<number-word> FROM <number-word> ;TO)
<term>                  ::=   <constant>
                        ::=   <variable>
                        ::=   (T: <quantifier-operator> <description>.                           ;Surface
                        ::=   (T: <quantifier> <formula>.)                                    ;Quantifier
                        ::=   (NT: <term>. ;TERMl <term>. ;TERM2)                                 ;Nested
                        ::=   (T: <hedge> <term>.)                                                 ;Hedge
                        ::=   (T: <operator1><term>.)                                          ;Operator1
                        ::=   (T: <operator2><term> ;TERM1 <term> ;TERM2)                      ;Operator2
                        ::=   (T: <operator><term> 1-2 ;TERMS)                                  ;Operator
<variable>              ::=   (*R 7AT0M (AND (ATOM !AT0M) (EQ (NTHCHAR !AT0M 1) 'X)))
```

Fig. 1

It is helpful, when describing the use of the language in HAM-RPM, to speak of two sublanguages, SURF and DEEP. Most of the productions in Fig. 1 are common to both sublanguages, but the element

'DESCRIPTION' occurs only in SURF, and the productions bearing the labels ;SURFACE and ;DEEP are possible only in the respective sublanguage. SURF is used as a target and source language by the two ATN grammars which analyse and generate natural language utterances, respectively. It is well suited for this purpose in that its expressions are roughly parallel to those of natural language. For example, the scope of quantifiers is not expressed explicitly. Example (10) is the SURF representation of sentence (9).

(9) Mehrere Blumen stehen auf jedem Tisch.
(Several flowers are on each table.)

```
(10) (IS   (T: (Q: MEHRER) (LAMBDA: X71 (ISA X71 BLUME)))
           (LAMBDA: X72   (RAEUMLICH AUF X72
                          (T: (Q: JED) (LAMBDA: X74 (ISA X74 TISCH)))))
```

DEEP, on the other hand, is better suited for the analysis and construction of expressions by the internal processes which draw inferences on the basis of the partner's utterances, update the partner model accordingly, and derive answers to the partner's explicit or implicit questions. Example (11) is the DEEP representation corresponding to the SURF expression in example (10).

```
(11) (F: ((FOR: (Q: JED) X76) (ISA X76 TISCH))
           (F: ((FOR: (Q:MEHRER) X75)(ISA X75 BLUME)) (RAEUMLICH AUF X75 X7)))
```

Although lexical and syntactic ambiguities in the input utterance are resolved when the corresponding SURF representation is constructed, the SURF expression itself may still be ambiguous. Referential or semantic ambiguities, e.g., in connection with the resolution of noun phrases, are not removed until the DEEP representation is generated. Even the DEEP expression, of course, may contain certain ambiguities concerning its interpretation as a speech act.

Vagueness, which, unlike ambiguity, is a typical feature of colloquial language, also differs from ambiguity in that it cannot be removed during the translation process, but must be reflected directly in the DEEP constructions. This is why DEEP provides for vague and explicitly hedged quantifiers as well as for hedged predications (see Fig. 1).

The way in which syntactic or lexical ambiguities are removed during the construction of SURF expressions becomes evident when the SURF expression corresponding to an ambiguous utterance is verbalized by HAM-RPM's ATN-based language generator via the procedure PARAPHRASE. For example, the input (12) would be paraphrased as (13). This

(12) Is the razor socket above the mirror in the bathroom?
(13) Is the razor socket above the mirror which is located in the bathroom?

paraphrase capability is used on the one hand in clarification dialogues (cf. v.HAHN et al. 1980) to check a given interpretation when an ambiguity is not easily resolvable on the basis of context information. On the other hand, it enables the system to give helpful answers to follow-up questions such as '*Did I ask you that?*', which the user may ask when he is puzzled by the system's answer. Incidentally, it is the SURF representation, not the surface string, which is stored in the system's dialogue memory, so that in general the verbalization of a question as it is recalled is not identical to the original verbal input.

The translation from SURF to DEEP is performed by the procedure NORMALIZE. This procedure works top-down and recursive, and can be passed any well-formed structure as an argument. It first normalizes the structure's immediate constituents and then transforms it in a way which varies from one type of structure to another. For example, when passed an 'ATOMIC-FORMULA' of type ;ARC2, it first normalizes each of the two 'TERMS' and then extracts any quantifiers they may contain in the appropriate order.

As a comparison of the last two examples will show, one of the main tasks of NORMALIZE is the determination of the scope of quantifiers. The algorithm used for this purpose is modeled after the one described by Hendrix (cf. HENDRIX 1976); it takes into account the intrinsic strenghts of the various natural language quantifier expressions and their relative positions within the SURFACE structure. A more sophisticated algorithm would, of course, take into account other factors.

The translation from DEEP to SURF is performed by the inverse procedure NORMALIZE-1 whose basic control structure is similar to that of NORMALIZE. Fig. 2 sketches the translations which occur when an input utterance is analysed and evaluated and leads to the generation of a natural language response.

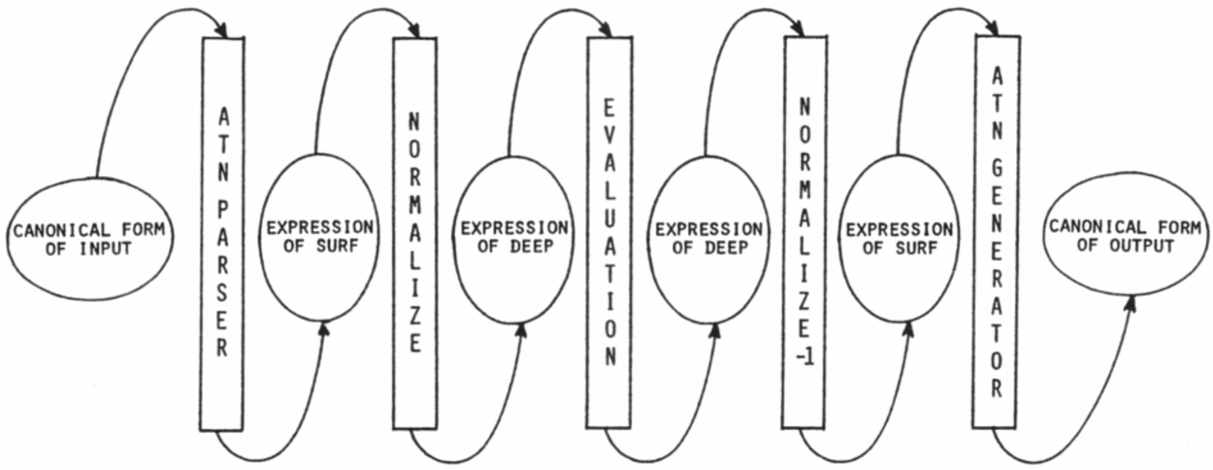

Fig. 2

Although Fig. 2 represents normalization as a separate pass, the fact that NORMALIZE can be applied to any well-formed structure of the language makes it possible to integrate this translation with the parsing of the natural language input: Substructures can be normalized as soon as they have been constructed by the parser, so that they can be further processed before the parsing proceeds further. This possibility is particular important in those cases in which it is useful to resolve a pronoun or definite noun phrase during the course of parsing.

2.3. The Evaluation Component

The evaluation component in HAM-RPM is, among other things, the procedural counterpart of what in logic is called the 'interpretation function'. Its tasks include the determination of the truth value of a given expression within a given context and the derivation of specific conclusions from an expression.

Evaluation is pattern-directed, being performed by two 'teams' of FUZZY 'Deduce procedures' called TEST-PROCs and GEN-PROCs. Each procedure has a characteristic pattern which matches some specified class of 'FORMULAS', though the patterns differ in their degree of specificity. GEN-PROCs generate constants which, when subsituted for a given variable, make a given formula true; TEST-PROCs determine the truth value of a given formula when a given variable is assigned a given value. For example, one GEN-PROC specializes in generating constants which fulfill a given conjunction of formulas, and one TEST-PROC checks the truth of assertions concerning spatial relations. The two teams call each other recursively, processing the input expression in a top-down manner.

The major advantage of this control structure for evaluation is that it relieves the designer of the temptation to handle syntactically similar structures according to a uniform scheme even when they in fact exhibit differences worth taking into account: Structures with special properties can be handled by separate specialized procedures (cf. section 2.4).

2.4 Selected Problems of Representation

We are now in a position to examine the way in which HAM-RPM deals with the representational problems mentioned in section 1. Inspection of the definition of the element 'QUANTIFIER-OPERATOR' shows that a relatively large and in fact open-ended-repertoire of quantifiers is provided. 'QUANTIFIER-WORDS' represent variants of the universal quantifier, including, e.g. the definite article and a quantifier corresponding to the expression most of the. Since the pattern-directed evaluation component just described allows nearly arbitrary control regimes to be associated with each of these quantifiers their idiosyncrasies can be taken into account in a natural manner.

The other major category of quantifiers, 'QUANTITIES', generalises the existential quantifier; it includes numerical quantifiers and in particular vague expressions such as several which, like the 'QUANTIFIER-WORDS', have special definitions associated with them.

The necessity of representing conjoined noun phrases which may have different quantifiers is responsible for the unusual first production in the definition of the element 'FORMULA': The quantifiers do not occupy the first position in the formula, as is usual in logic-based formalisms, but rather are contained within the first 'TERM', and not even on the top level, as can be seen from the definition of 'TERM'. Several specialized TEST-PROCs are required to locate quantifiers during the evaluation and to supervise their interpretation. Incidentally, conjoined noun phrases present some interesting problems in connection with normalization, as their quantifiers must have the same scope Modifiers such as of course and unfortunately have the same syntactic role as the unary 'OPERATOR1' NOT, i.e., they can modify 'FORMULAs', 'TERMs', or 'DESCRIPTIONs'. Leaving the interpretation of the particles to the corresponding GEN-PROCs and TEST-PROCs has the advantage of leaving open the exact use to which they will be put during evaluation. This is important because such particles do not always serve to express information about the speaker's beliefs or desires but may instead simply help to disambiguate referring expressions, as in (14) when the two possible antecedents for *it* are *the bathtub* and *the hole in the wall*.

(14) *Manager:* Unfortunately, it is rather large.

3. The Processing of an Example Dialogue Segment

All of the processing phases which can occur when HAM-RPM responds to an utterance of the user's are presented in linear order in Fig. 3. The actual flow of control in any particular case is in general quite different, involving the repetition or omission of various processing phases. This is the case in particular when clarification dialogues occur.

In the remainder of this section a possible response of the system to the question (15) in the hotel situation will be traced. An example was chosen in which the representational problems discussed above and the system's construction and use of a partner model play a relatively minor role, as work on some aspects of these problems has not yet been completed.

(15) Was fuer Sitzgelegenheiten, die einigermassen bequem sind, befinden sich in diesem Zimmer?
(What chairs that are reasonably comfortable are there in this room?)

The following discussion can best be understood if continual reference is made to Fig.3.

The task of the component *Lexical Analysis* is to create a canonical representation of the input string. This requires a nontrivial control structure, as the results of one process may in turn establish preconditions for other processes (cf. HOEPPNER/ JAMESON 1979). A first set of processes tries to find the canonical form of each word in the input utterance using declarative and procedural lexical knowledge. In the example inflectional suffixes and punctuation marks are separated from the words to which they are appended. A second set of processes identifies and transforms special syntag-matic groups of words. The reflexive verb *sich befinden*, whose two words in general needn't even be adjacent within the sentence, is replaced by the nonreflexive verb *stehen* (to stand). The construction *was fuer* is replaced by the synonymous word *welch* (which). The final string which leaves *Lexical Analysis* is:

(16) WELCH SITZGELEGENHEIT /, D- EINIGERMASSEN BEQUEM SEIN /,
STEH IN DIES ZIMMER /?

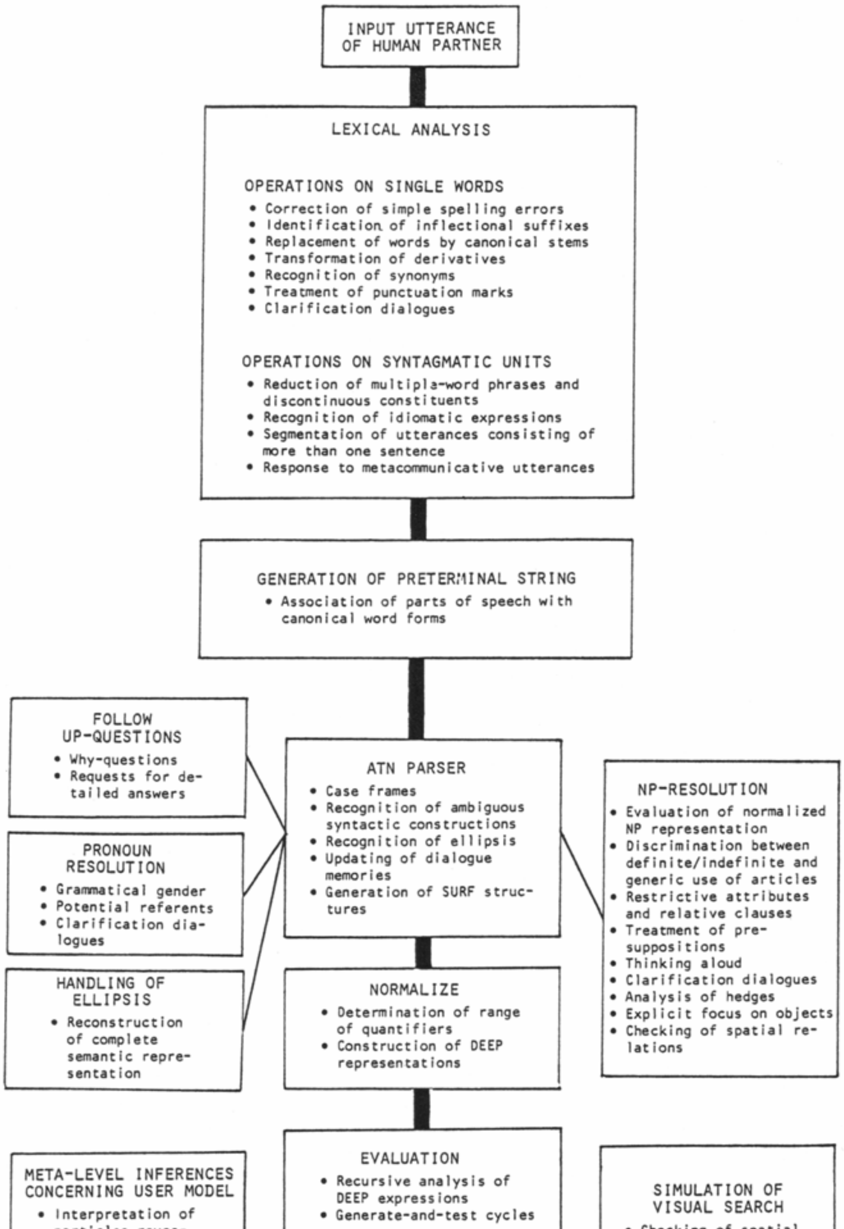
INPUT UTTERANCE OF HUMAN PARTNER
LEXICAL ANALYSIS
OPERATIONS ON SINGLE WORDS
• Correction of simple spelling errors
• Identification of inflectional suffixes
• Replacement of words by canonical stems
• Transformation of derivatives
• Recognition of synonyms
• Treatment of punctuation marks
• Clarification dialogues
OPERATIONS ON SYNTAGMATIC UNITS
• Reduction of multipla-word phrases and discontinuous constituents
• Recognition of idiomatic expressions
• Segmentation of utterances consisting of more than one sentence
• Response to metacommunicative utterances
GENERATION OF PRETERMINAL STRING
• Association of parts of speech with canonical word forms
FOLLOW UP-QUESTIONS
• Why-questions
• Requests for detailed answers
PRONOUN RESOLUTION
• Grammatical gender
• Potential referents
• Clarification dialogues
HANDLING OF ELLIPSIS
• Reconstruction of complete semantic representation
ATN PARSER
• Case frames
• Recognition of ambiguous syntactic constructions
• Recognition of ellipsis
• Updating of dialogue memories
• Generation of SURF structures
NP-RESOLUTION
• Evaluation of normalized NP representation
• Discrimination between definite/indefinite and generic use of articles
• Restrictive attributes and relative clauses
• Treatment of presuppositions
• Thinking aloud
• Clarification dialogues
• Analysis of hedges
• Explicit focus on objects
• Checking of spatial relations
NORMALIZE
• Determination of range of quantifiers
• Construction of DEEP representations
META-LEVEL INFERENCES CONCERNING USER MODEL
• Interpretation of particles revealing user's beliefs and desires
• Performing assumed inferences of the user
• Hypotheses about user's intentions and expectations
EVALUATION
• Recursive analysis of DEEP expressions
• Generate-and-test cycles
GENERATION OF RESPONSE IN DEEP
• Vague inferences
• Network search
• Selectional restrictions
• Construction of inference memory
SIMULATION OF VISUAL SEARCH
• Checking of spatial relations
• Search for possible referents
• Rating of referents
• Consideration of dialogue focus

Fig. 3

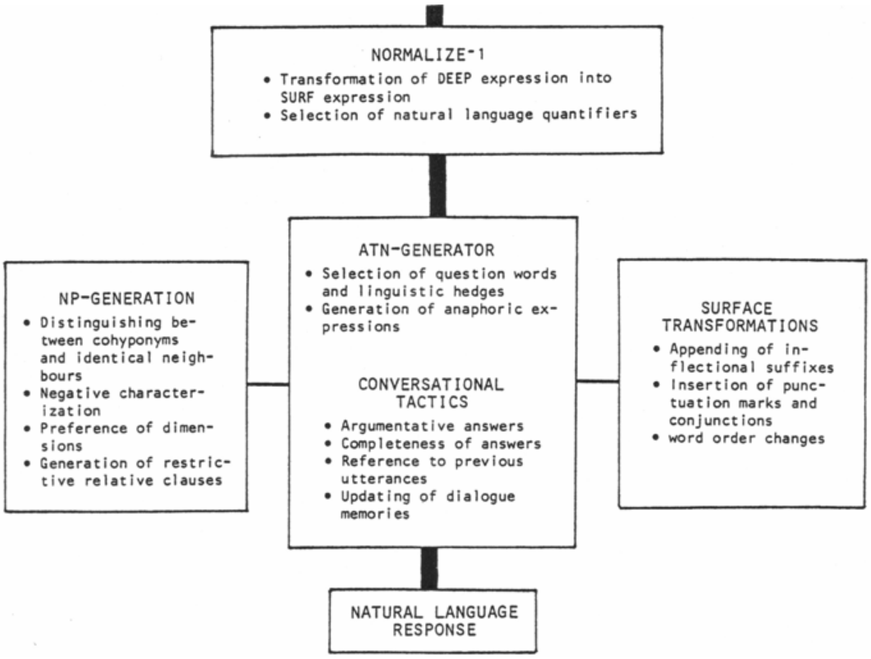
NORMALIZE-1
• Transformation of DEEP expression into SURF expression
• Selection of natural language quantifiers
NP-GENERATION
• Distinguishing between cohyponyms and identical neighbours
• Negative characterization
• Preference of dimensions
• Generation of restrictive relative clauses
ATN-GENERATOR
• Selection of question words and linguistic hedges
• Generation of anaphoric expressions
CONVERSATIONAL TACTICS
• Argumentative answers
• Completeness of answers
• Reference to previous utterances
• Updating of dialogue memories
SURFACE TRANSFORMATIONS
• Appending of inflectional suffixes
• Insertion of punctuation marks and conjunctions
• word order changes
NATURAL LANGUAGE RESPONSE

Fig. 3 (cont'd)

In the system's lexicon each canonical word form is associated with its part of speech (or parts of speech, if it is lexically ambiguous) and other syntactic markers such as gender or article type. The preterminal string generated: by the next component consists of the word forms and their parts of speech:

```
(17) ((WELCH IPRN) (SITZGELEGENHEITEN NOM) (/, SATZZ) (D- DET RELPRN)
(EINIGERMASSEN ADV) (BEQUEM ADV) (SEIN VRB) (/, SATZZ) (STEH VRB)
(IN PRP) (DIES DEM PRN) (ZIMMER NOM) (/? SATZZ))
```

This preterminal string forms the input to the ATN parser (cf. FININ 1977), a component which on the one hand generates the SURF structure of the utterance and on the other hand activates complex semantic-pragmatic processing phases. (18) is the SURF expression for the preterminal string (17).

```
(18) (IS  (T:  (Q: WELCH)
               (D: AND  (LAMBDA: X1 (ISA X1 SITZGELEGENHEIT))
                        (LAMBDA: X2 (REF X2 (T: EINIGERMASSEN BEQUEM)))))
          (LAMBDA: X3 (RAEUMLICH IN X3
                         (T: (Q DIES) (LAMBDA: X4 (ISA X4 ZIMMER))))))
```

This SURF representation contains only one structure -- the embedded term for the noun phrase *dies Zimmer* -- which causes the activation of a reference-semantic processing component. SURF structures which represent noun phrases with the definite article are, when appropriate, NORMALIZEd, in this case yielding the DEEP 'TERM' (19), and passed to the NP-resolution component.

```
(19) (T: (FOR: (Q: DIES) X40) (ISA X40 ZIMMER))
```

In such cases this component looks for a unique referent in the domain of discourse. The system refers to the knowledge base, in particular to the conceptual and referential semantic networks and the dialogue memories, to try to identify the internal name of the object the NP was intended to refer to. The analyis of the noun phrase in the example described here is particularly simple: The room in question is assumed to be the one the hotel manager has offered. The SURF expression corresponding to the noun phrase in now replaced in (18) by the constant ZIMMER4, the internal name of the room. The resulting structure, an 'ATOMIC-FORMULA', is now passed to NORMALIZE, which transforms it into a 'FORMULA' of the type ;DEEP:

```
(20) (F:   (T:   (FOR: (Q WELCH) X31)
                 (F: AND (ISA X31 SITZGELEGENHEIT)
                         (F:   (T: EINIGERMASSEN (T:(FOR: (Q: BEQUEM) X33) T) )
                               (REF X31 X33))))
           (RAEUMLICH IN X31 ZIMMER4))
```

The semantic-pragmatic evaluation of these DEEP structures makes heavy use of the model of the caller that the hotel manager has built up during the dialogue. Let us assume that he believes the caller to be a salesman. The present question might then lead to the assumption that he is interested in meeting with customers in his hotel room. This new hypothesis will then be added to the system's user model so that it can be referenced again during the dialogue.

This structure is then passed to the evaluation component so that an answer can be determined. The evaluation begins with an application of the team TEST-PROCs, just as if the structure represented a yes-no-question, because *WELCH* is defined as a quantifier which causes a description of a set of objects to be returned (instead of a truth value).

A uniform evaluation procedure might begin by generating arbitrary chairs from the system's knowledge base -- i.e. chairs in all rooms of the hotel with which the manager is familiar -- testing to see if they are '*reasonably comfortable*', and finally checking whether they are located in the room in question. It is obviously more efficient and psychologically more realistic if the system immediately takes the spatial restriction into account by restricting its attention to the room in question. This is easily accomplished within the general control structure outlined in section 2.3: A TEST-PROC is defined which, when recognizing certain sorts of spatial restrictions in certain parts of a 'FORMULA', simply switches to the corresponding CONTEXT of FUZZY's data base and applies TEST-PROCs to the rest of the structure.

The following procedures are activated during the remainder of the evaluation: a TEST-PROC capable of handling the quantifier WELCH; a GEN-PROC specialized in conjunctions; a GEN-PROC which generates names of objects corresponding to a general concept (*SITZGELEGENHEIT* applies to all kinds of chairs, as well as couches, stools, etc.); a TEST-PROC specialized in hedges such as *EINIGERMASSEN* and a TEST-PROC which can determine how '*comfortable*' a given object is, using, if necessary, general knowledge about the properties of various types of furniture. The TEST-PROC responsible for WELCH collects the internal names of the objects which fulfill the stated conditions, eventually outputting the DEEP expression (21) which represents their conjunction

```
(21) (T: AND   (T: (FOR: (Q: SOFA3) X31) T)
               (T: AND      (T: (FOR: (Q: SESSEL1) X31) T)
                            (T: (FOR: (Q: SESSEL5) X31) T)))
```

If the system had any reason to generate a syntactically complete answer, it would substitute (21) for the corresponding structure in (20) and verbalize the entire structure, but we will assume that it simply passes (21) to NORMALIZE-1, which returns the SURF structure (22), and then calls the ATN-based generator to verbalize this structure. The generator calls the NP-generation component as a subroutine when it encounters the internal names of the three objects.

```
(22) (T: AND SOFA3 (T: AND SESSEL1 SESSEL5))
```

If any of the three objects has already been mentioned during the dialogue, it will be described with a definite noun phrase; we will assume this not to be the case, so that indefinite descriptions are appropriate. Noticing that two of the descriptions returned by the NP-generator are identical ('*E-SESSEL*'), the ATN-based generator combines them before outputting the canonical string (23).

(23) (E- SOFA 2 SESSEL)

The last component in the generation process is the *Surface Transformations*, the complement to *Lexical Analysis*. Here inflectional suffixes which mark case and number are appended, punctuation marks and conjunctions are inserted, and numbers are replaced by words. The natural language response to (15) is therefore:

(24) EIN SOFA UND ZWEI SESSEL.

Before HAM-RPM will be able to respond appropriately in cases where the user's utterance is less transparent and a straightforward response would be less encouraging, much work must be devoted to enabling it to construct and make use of an explicit model of its dialogue partner. But we believe that the representation language and associated processes described here have sufficient expressive power and generality to serve as an adequate foundation for these further developments. (See v.HAHN et al. 1980 for a detailed description of HAM-RPM's processing phases, including those which were not involved in this example, e.g., the handling of pronouns, ellipsis and follow-up questions.)

4. Technical Data

A noncompiled version of HAM-RPM is running under the TOPS 10 Operating system on the DECsystem 1070 (PDP-10) of the Fachbereich für Informatik at the University of Hamburg. Compromising approximately 600 LISP/FUZZY procedures, the current version requires from one to fifteen seconds for a response. The ATTACH package (Jameson 1980) was developed as a piece of auxiliary system software for HAM-RPM to allow direct access to programs and data on secondary storage. The complete system, which occupies at present 150k of 36-bit words thus only needs to be loaded when all components are being used and speed is important. The system of demand fetching provided by the ATTACH package makes it possible for records in the assertional data base, the lexica, and the ATN grammars to be loaded into core only when required. ATTACH permits to interface HAM-RPM to large data bases which are written in nearly arbitrary formats, which needn't conform to LISP syntax. An important prerequisite for the application of a version of HAM-RPM as an intelligent natural language interface to nonlinguistic AI systems and to documentation or information systems is thus fulfilled.

Acknowledgement

This research is currently being supported by the Deutsche Forschungsgemeinschaft (DFG) under contract Ha1026/2.

References

BOBROW, D.G., KAPLAN, R.M., KAY, M., NORMAN, D.A., THOMPSON, H., WINOGRAD, T. (1977): GUS - A frame-driven dialog system. In: Artificial Intelligence, 8, 1, 155-173

COHEN, P.R. (1978): On knowing what to say: Planning speech acts. Univ. of Toronto, Department of Computer Science, Technical Report No. 118

FININ, T.W. (1977): An interpreter and compiler for augmented transition networks. Thesis, Univ. of Illinois, Department of Computer Science

HABEL, CH., ROLLINGER, G-R., SCHMIDT, A. (1978): Eine semantische Reprasentationssprache zur Darstellung modaler Konzepte. Univ. Berlin, Fachbereich für Informatik, Technical Report SNR 6/78 v.HAHN, W., HOEPPNER, W., JAMESON, A., WAHLSTER, W. (1978): HAM-RPM: Natural Dialogues with an artificial partner. In: Proceedings of the AISB/GI Conference on Artificial Intelligence. Hamburg, 122-131

v.HAHN, W., HOEPPNER, W., JAMESON, A., WAHLSTER, W. (1980): The anatomy of the natural language dialogue system HAM-RPM. In: Bole, L. (ed.): Natural language based computer systems. Munich: Hanser/MacmilIan, 119-253

HENDRIX, G.G. (1976): Semantic aspects of translation. In: Walker, D.E. (ed.): Speech understanding research. Final technical Report, SRI, Menlo Park

HOEPPNER, W., JAMESON, A. (1979): Kooperatives Dialogverhalten im Simulationssystem HAM-RPM. In: Proceedings of the 4th Workshop on Artificial Intelligence. Bad Honnef, 21-31

JAMESON, A. (1980): ATTACH: A package for accessing LISP programs and data from disc. HAM-RPM Memo 12

LEFAIVRE, R.A. (1977): FUZZY reference manual. Rutgers University, Computer Science Department

RICH, E. (1979): Building and exploiting user models. Carnegie Mellon University, Computer Science Department, CMU-CS-79-119

SHAPIRO, S.C. (1979): Numerical quantifiers and their use in reasoning with negative information. In: Proceedings of the 6th International Joint Conference on Artificial Intelligence, Tokyo, 791-796

WAHLSTER, W., JAMESON, A., HOEPPNER, W. (1978): Glancing, referring and explaining in the dialogue system HAM-RPM. In: American Journal of Computational Linguistics, Microfiche 77, 53-67

WINOGRAD, T. (1976): Towards a procedural understanding of semantics. Stanford Artificial Intelligence Laboratory, Memo AIM 292

ZIFONUN, G. (1979): Formale Reprasentation natürlichsprachlicher Äußerungen. In: Kolvenbach, M., Lötscher, A., Lutz, H.D. (eds.): Künstliche Intelligenz und natürliche Sprache. Sprachverstehen und Problemlösen mit dem Computer. Tübingen: Narr, 93-134