A KNOWLEDGE ENGINEERING APPROACH TO NATURAL LANGUAGE UNDERSTANDING

Stuart C. Shapiro & Jeannette G. Neal
Department of Computer Science
State University of New York at Buffalo
Amherst, New York 14226

ABSTRACT

This paper describes the results of a preliminary study of a Knowledge Engineering approach to Natural Language Understanding. A computer system is being developed to handle the acquisition, representation, and use of linguistic knowledge. The computer system is rule-based and utilizes a semantic network for knowledge storage and representation. In order to facilitate the interaction between user and system, input of linguistic knowledge and computer responses are in natural language. Knowledge of various types can be entered and utilized: syntactic and semantic; assertions and rules. The inference tracing facility is also being developed as a part of the rule-based system with output in natural language. A detailed example is presented to illustrate the current capabilities and features of the system.

I INTRODUCTION

This paper describes the results of a preliminary study of a Knowledge Engineering (KE) approach to Natural Language Understanding (NLU). The KE approach to an Artificial Intelligence task involves a close association with an expert in the task domain. This requires making it easy for the expert to add new knowledge to the computer system, to understand what knowledge is in the system, and to understand how the system is accomplishing the task so that needed changes and corrections are easy to recognize and to make. It should be noted that our task domain is NLU. That is, the knowledge in the system is knowledge about NLU and the intended expert is an expert in NLU.

The KE system we are using is the SNePS semantic network processing system [11]. This system includes a semantic network system in which all knowledge, including rules, is represented as nodes in a semantic network, an inference system that performs reasoning according to the rules stored in the network, and a tracing package that allows the user to follow the system's reasoning. A major portion of this study involves the design and implementation of a SNePS-based system, called the NL-system, to enable the NLU expert to enter linguistic knowledge into the network in natural language, to have this knowledge available to query and reason about, and to use this knowledge for processing text including additional NLU knowledge. These features distinguish our system from other rule-based natural language processing systems such as that of Pereira and Warren [9] and Robinson [10].

One of the major concerns of our study is the acquisition of knowledge, both factual assertions and rules of inference. Since both types of knowledge are stored in similar form in the semantic network, our NL-system is being developed with the ability to handle the input of both types of knowledge, with this new knowledge immediately available for use. Our concern with the acquisition of both types of knowledge differs from the approach of Haas and Hendrix [1], who are pursuing only the acquisition of large aggregations of individual facts.

The benefit of our KE approach may be seen by considering the work of Lehnert [5]. She compiled an extensive list of rules concerning how questions should be answered. For example, when asked, "Do you know what time it is?", one should instead answer the question "What time is it?". Lehnert only implemented and tested some of her rules, and those required a programming effort. If a system like the one being proposed here had been available to her, Lehnert could have tested all her rules with relative ease.

Our ultimate goal is a KE system with all its linguistic knowledge as available to the language expert as domain knowledge is in other expert systems. In this preliminary study we explore the feasibility of our approach as implemented in our representations and NL-system.

** This work was supported in part by the National Science Foundation under Grants MCS80-06314 and SPI-8019895.

II OVERVIEW OF THE NL-SYSTEM

A major goal of this study is the design and implementation of a user-friendly system for experimentation in KE applied to Natural Language Understanding.

The NL-system consists of two logical components:

a) A facility for the input of linguistic knowledge into the semantic network in natural language. This linguistic knowledge primarily consists of rules about NLU and a lexicon. The NL-system contains a core of network rules which parse a user's natural language rule and build the corresponding structure in the form of a network rule. This NL-system facility enables the user to manipulate both the syntactic and semantic aspects of surface strings.

b) A facility for phrase/sentence generation and question answering via rules in the network. The user can pose a limited number of types of queries to the system in natural language, and the system utilizes rules to parse the query and generate a reply. An inference tracing facility is also being developed which uses this phrase/sentence generation capability. This will enable the user to trace the inference processes, which result from the activation of his rules, in natural language.

When a person uses this NL-system for experimentation, there are two task domains co-resident in the semantic network. These domains are: (1) The NLU-domain which consists of the collection of propositions and rules concerning Natural Language Understanding, including both the NL-system core rules and assertions and the user-specified rules and assertions; and (2) the domain of knowledge which the user enters and interacts with via the NLU domain. For this study, a limited "Bottle Domain" is used as the domain of type (2). This domain was chosen to let us experiment with the use of semantic knowledge to clarify, during parsing, the way one noun madifies another in a noun-noun construction, viz. "milk bottle" vs. "glass bottle".

In a sense, the task domain (2) is a sub-domain of the NLU-domain since task domain (2) is built and used via the NLU-domain. However, the two domains interact when, for example, knowledge from both domains is used in understanding a sentence being "read" by the system.

The system is dynamic and new knowledge, relevant to either or both domains, can be added at any time.

III PRELIMINARIES FOR ENTERING RULES

The basic tools that the language expert will need to enter into the system are a lexicon of words and a set of processing rules. This system enables them to be input in natural language.

The system initially uses five "undefined terms": L-CAT, S-CAT, L-REL, S-REL, and VARIABLE. L-CAT is a term which represents the category of all lexical categories such as VERB and NOUN. S-CAT represents the category of all string categories such as NOUN PHRASE or VERB PHRASE. L-REL is a term which represents the category of relations between a string and its lexical constituents. Examples of L-RELs might be MOD NOUN and HEAD NOUN (of a NOUN NOUN PHRASE). S-REL represents the category of relations between a string and its sub-string constituents, such as FIRST NP and SECOND NP (to distinguish between two NPs within one sentence). VARIABLE is a term which represents the class of identifiers which the user will use as variables in his natural language rules.

Before entering his rules into the system, the user must inform the system of all members of the L-CAT and VARIABLE categories which he will use. Words in the S-CAT, L-REL and S-REL categories are introduced by the context of their use in user-specified rules. The choice of all linguistic names is totally at the discretion of the user.

A list of the initial entries for the example of this paper are given below. The single quote mark indicates that the following word is mentioned rather than used. Throughout this paper, lines beginning with the symbol ** are entered by the user and the following line(s) are the computer response. In response to a declarative input statement, if the system has been able to parse the statement and build a structure in the semantic network to represent the input statement, then the computer replies with an echo of the user's statement prefaced by the phrase "I UNDERSTAND THAT". In other words, the building of a network structure is the system's "representation" of understanding.

```
** 'NOUN IS AN L-CAT.
I UNDERSTAND THAT ' NOUN IS AN L-CAT
** 'G-DETERMINER IS AN L-CAT.
(NOTE: Computer responses are omitted for
these input statements due to space
constraints of this paper. Responses are all
similar to the one shown above.)
** 'RELATION IS AN L-CAT.
** 'E IS A VARIABLE.
** 'X IS A VARIABLE.
```

```
** 'Y IS A VARIABLE.
** 'ON IS A RELATION.
** 'A IS A G-DETERMINER.
** 'BOTTLE IS A NOUN.
** 'CONTAINER IS A NOUN.
** 'TABLE IS A NOUN.
** 'DESK IS A NOUN.
** 'BAR IS A NOUN.
** 'FLUID IS A MASS-NOUN.
** 'MATERIAL IS A MASS-NOUN.
** 'MILK IS A MASS-NOUN.
** 'WATER IS A MASS-NOUN.
** 'GLASS IS A MASS-NOUN.
```

IV THE CORE OF THE NL-SYSTEM

The core of the NL-system contains a collection of rules which accepts the language defined by the grammar listed in the Appendix. The core is responsible for parsing the user's natural language input statements and building the corresponding network structure.

It is also necessary to start with a set of semantic network structures representing the basic relations the system can use for knowledge representation. Currently these relations are:

a) Word W is preceded by "connector point" P in a surface string; e.g. node M3 of figure 1 represents that word IS is preceded by connector point M2 in the string;
b) Lexeme L is a member of category C; e.g. this is used to represent the concept that 'BOTTLE IS A NOUN, which was input in Section 3;
c) The string beginning at point P1 and ending at point P2 in a surface string is in category C; e.g. node M53 of figure 3 represents the concept that "a bottle" is a GNP;
d) Item X has the relation R to item Y; e.g. node M75 of figure 1 represents the concept that the class of bottles is a subset of the class of containers;
e) A class is characterized by its members participating in some relation; e.g. the class of glass bottles is characterized by its members being made of glass;
f) The rule structures of SNePS.

V SENTENTIAL COMPONENT REPRESENTATION

The representation of a surface string utilized in this study consists of a network version of the list structure used by Pereira and Warren [10] which eliminates the explicit "connecting" tags or markers of their alternate representation. This representation is also similar to Kay's charts [4] in that several structures may be built as alternative analyses of a single substring. The network structure built up by our top-level "reading" function, without any of the additional structure that would be added as a result of processing via rules of the network, is illustrated in figure 1.

As each word of an input string is read by the system, the network representation of the string is extended and relevant rules stored in the SNePS network are triggered. All applicable rules are started in parallel by processes of our MULTI package [8], are suspended if not all their antecedents are satisfied, and are resumed if more antecedents are satisfied as the string proceeds. The SNePS bidirectional inference capability [6] focuses attention towards the active parsing processes and cuts down the fan out of pure forward or backward chaining. The system has many of the attributes and benefits of Kaplan's producer-consumer model [3] which influenced the design of the inference system. The two SNePS subsystems, the MULTI inference system and the MATCH subsystem, provide the user with the pattern matching and parse suspension and continuation capability enjoyed by the Flexible Parser of Hayes & Mouradian [2].

VI INPUT AND PROCESSING OF THE USER'S RULES

After having entered a lexicon into the system as described above, the user will enter his natural language rules. These rules must be in the IF-THEN conditional form. A sample rule that the user might enter is:

IF A STRING CONSISTS OF A G-DETERMINER FOLLOWED BY A NOUN CALLED THE MOD-NOUN FOLLOWED BY ANOTHER NOUN CALLED THE HEAD-NOUN
THEN THE STRING IS AN NNP.

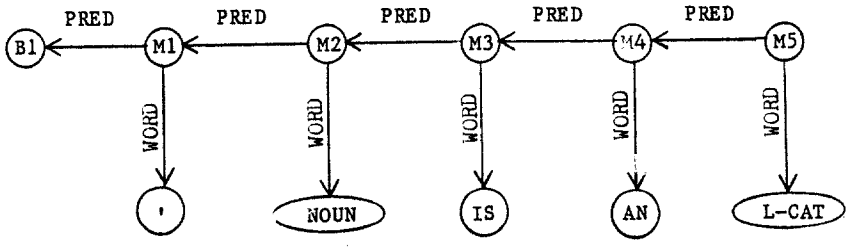

Figure 1. Network representation of a sentence.

The words which are underlined in the above rule are terms selected by the user for certain linguistic entities. The lexical category names such as G-DETERMINER and NOUN must be entered previously as discussed above. The words MOD-NOUN and HEAD-NOUN specify lexical constituents of a string and therefore the system adds them to the L-REL category. The string name NNP is added to the S-CAT category by the system.

The user's rule-statement is read by the system and processed by existing rules as described above. When it has been completely analyzed, a translation of the rule-statement is asserted in the form of a network rule structure. This rule is then available to analyze further user inputs.

The form of these user rules is determined by the design of our initial core of rules. We could, of course, have written rules which accept user rules of the form

NNP ---> G-DETERMINER NOUN NOUN.

Notice, however, that most of the user rules of this section contain more information than such simple phrase-structure rules.

Figure 2 contains the list of the user natural language rules which are used as input to the NL-system in the example developed for this paper. These rules illustrate the types of rules which the system can handle.

By adding the rules of figure 2 to the system, we have enhanced the ability of the NL-

```
1. ** IF A STRING CONSISTS OF A MASS-NOUN
   *  THEN THE STRING IS A GNP
   *    AND THE GNP EXPRESSES THE CONCEPT NAMED BY THE MASS-NOUN.
   I UNDERSTAND THAT IF A STRING CONSISTS OF A MASS-NOUN THEN THE STRING
   IS A GNP AND THE GNP EXPRESSES THE CONCEPT NAMED BY THE MASS-NOUN

2. ** IF A STRING CONSISTS OF A G-DETERMINER FOLLOWED BY A NOUN
   *  THEN THE STRING IS A GNP
   *    AND THE GNP EXPRESSES THE CONCEPT NAMED BY THE NOUN.

   (NOTE: Computer responses omitted for these rules due to space constraints of
   this paper. Responses are exemplified by the response to first rule above.)

3. ** IF A STRING CONSISTS OF A G-DETERMINER FOLLOWED BY A NOUN CALLED
   *  THE MOD-NOUN FOLLOWED BY ANOTHER NOUN CALLED THE HEAD-NOUN
   *    THEN THE STRING IS AN NNP.

4. ** IF A STRING CONSISTS OF AN NNP
   *    THEN THERE EXISTS A CLASS E SUCH THAT
   *    THE CLASS E IS A SUBSET OF THE CLASS NAMED BY THE HEAD-NOUN
   *    AND THE NNP EXPRESSES THE CLASS E.

5. ** IF A STRING CONSISTS OF AN NNP
   *  AND THE NNP EXPRESSES THE CLASS E
   *  AND THE CLASS NAMED BY THE MOD-NOUN IS A SUBSET OF MATERIAL
   *  AND THE CLASS NAMED BY THE HEAD-NOUN IS A SUBSET OF CONTAINER
   *  THEN THE CHARACTERISTIC OF E IS TO BE MADE-OF THE ITEM NAMED
   *    BY THE MOD-NOUN.

6. ** IF A STRING CONSISTS OF AN NNP
   *  AND THE NNP EXPRESSES THE CLASS E
   *  AND THE CLASS NAMED BY THE MOD-NOUN IS A SUBSET OF FLUID
   *  AND THE CLASS NAMED BY THE HEAD-NOUN IS A SUBSET OF CONTAINER
   *  THEN THE FUNCTION OF E IS TO BE CONTAINING THE ITEM NAMED BY THE
   *    MOD-NOUN.

7. ** IF A STRING CONSISTS OF A GNP CALLED THE FIRST-GNP FOLLOWED BY
   *  THE WORD 'IS FOLLOWED BY A GNP CALLED THE SECOND-GNP
   *  THEN THE STRING IS A DGNP-SNTC.

8. ** IF A STRING CONSISTS OF A DGNP-SNTC
   *  THEN THE CLASS NAMED BY THE FIRST-GNP IS A SUBSET OF THE CLASS
   *    NAMED BY THE SECOND-GNP
   *    AND THE DGNP-SNTC EXPRESSES THIS LAST PROPOSITION.

9. ** IF A STRING CONSISTS OF AN NNP FOLLOWED BY THE WORD 'IS
   *  FOLLOWED BY A RELATION FOLLOWED BY A GNP
   *  THEN THE STRING IS A SENTENCE
   *    AND THERE EXISTS AN ITEM X AND THERE EXISTS AN ITEM Y
   *    SUCH THAT THE ITEM X IS A MEMBER OF THE CLASS NAMED BY THE NNP
   *    AND THE ITEM Y IS A MEMBER OF THE CLASS NAMED BY THE GNP
   *    AND THE ITEM X HAS THE RELATION TO THE ITEM Y
   *    AND THE SENTENCE EXPRESSES THIS LAST PROPOSITION.

10.** IF THE FUNCTION OF E IS TO BE CONTAINING THE ITEM X
   *  AND Y IS A MEMBER OF E
   *    THEN THE FUNCTION OF Y IS TO BE CONTAINING THE ITEM X.

11.** IF THE CHARACTERISTIC OF E IS TO BE MADE OF THE ITEM X
   *  AND Y IS A MEMBER OF E
   *    THEN THE CHARACTERISTIC OF Y IS TO BE MADE OF THE ITEM X.
```

Figure 2. The rules used as input to the system.

system to "understand" surface strings when "read" into the network. If we examine rules 1 and 2, for example, we find they define a GNP (a generic noun phrase). Rules 4, 8, and 9 stipulate that a relationship exists between a surface string and the concept or proposition which is its intension. This relationship we denoted by "expresses". When these rules are triggered, they will not only build syntactic information into the network categorizing the particular string that is being "read" in, but will also build a semantic node representing the relationship "expresses" between the string and the node representing its intension. Thus, both semantic and syntactic concepts are built and linked in the network.

In contrast to rules 1 - 9, rules 10 and 11 are purely semantic, not syntactic. The user's rules may deal with syntax alone, semantics alone, or a combination of both.

All knowledge possessed by the system resides in the same semantic network and, therefore, both the rules of the NL-system core and the user's rules can be triggered if their antecedents are satisfied. Thus the user's rules can be used not only for the input of surface strings concerning the task domain (2) discussed in Section 2, but also for enhancing the NL-system's capability of "understanding" input information relative to the NLU domain.

VII PROCESSING ILLUSTRATION

Assuming that we have entered the lexicon via the statements shown in Section 3 and have entered the rules listed in Section 6, we can input a sentence such as "A bottle is a container". Figure 3 illustrates the network representation of the surface string "A bottle is a container" after having been processed by the user's rules listed in Section 6. Rule 2 would be triggered and would identify "a bottle" and "a container" as GNPs, building nodes M53, M55, M61, and M63 of figure 3. Then the antecedent of rule 7 would be satisfied by the sentence, since it consists of a GNP, namely "a bottle", followed by the word "is", followed by a GNP, namely "a container". Therefore the node M90 of figure 3 would be built identifying the sentence as a DGNP-SNTC. The addition of this knowledge would trigger rule 8 and node M75 of figure 3 would be built asserting that the class named "bottle" is a subset of the class named "container". Furthermore, node M91 would be built asserting that the sentence EXPRESSES the above stated subset proposition.

Let us now input additional statements to the system. As each sentence is added, node structures are built in the network concerning both the syntactic properties of the sentence and the underlying semantics of the sentence. Each of these structures is built into the system only, however, if it is the consequence of the triggering of one of the expert's rules.

We now add three sentences (preceded by the **) and the program response is shown for each.

**A BOTTLE IS A CONTAINER.
I UNDERSTAND THAT A BOTTLE IS A CONTAINER

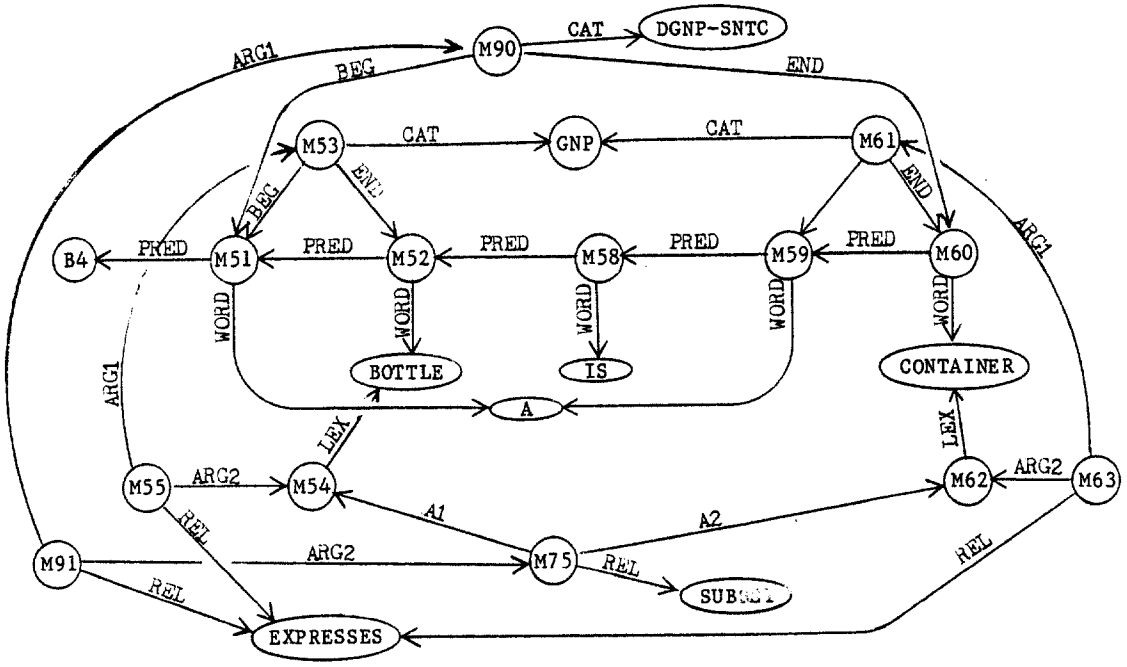

Figure 3. Network representation of processed surface string.

**MILK IS A FLUID.
I UNDERSTAND THAT MILK IS A FLUID

**GLASS IS A MATERIAL.
I UNDERSTAND THAT GLASS IS A MATERIAL

Each of the above input sentences is parsed by the rules of Section 6 identifying the various noun phrases and sentence structures, and a particular semantic subset relationship is built corresponding to each sentence.

We can now query the system concerning the information just added and the core rules will process the query. The query is parsed, an answer is deduced from the information now stored in the semantic network, and a reply is generated from the network structure which represents the assertion of the subset relationship built corresponding to each of the above input statements. The next section discusses the question-answering/generation facility in more detail.

** WHAT IS A BOTTLE?
A BOTTLE IS A CONTAINER

Now we input the sentence "A milk bottle is on a table". The rules involved are rules 2, 3, 4, 6, 9, and 10. The phrase "a milk bottle" triggers rule 3 which identifies it as a NNP (noun-noun phrase). Then since the string has been identified as an NNP, rule 4 is triggered and a new class is created and the new class is a subset of the class representing bottles. Rule 6 is also triggered by the addition of the instances of the consequents of rules 3 and 4 and by our previous input sentences asserting that "A bottle is a container" and "Milk is a fluid". As a result, additional knowledge is built into the network concerning the new sub-class of bottles: the function of this new class is to contain milk. Then since "a table" satisfies the conditions for rule 2, it is identified as a GNP, rule 9 is finally triggered, and a structure is built into the network representing the concept that a member of the set of bottles for containing milk is on a member of the set of tables. The antecedents of rule 10 are satisfied by this member of the set of bottles for containing milk, and an assertion is added to the effect that the function of this member is also to contain milk. The computer responds "I UNDERSTAND THAT . . ." only when a sructure has been built which the sentence EXPRESSES.

** A MILK BOTTLE IS ON A TABLE.
I UNDERSTAND THAT A MILK BOTTLE IS ON A TABLE

In order to further ascertain whether the system has understood the input sentence, we can query the system as follows. The system's core rules again parse the query, deduce the answer, and generate a phrase to express the answer.

** WHAT IS ON A TABLE?
A BOTTLE FOR CONTAINING MILK

We now input the sentence "A glass bottle is on a desk" to be parsed and processed by the rules of Section 6. Processing of this sentence is similar to that of the previous sentence, except that rule 5 will be triggered instead of rule 6 since the system has been informed that glass is a material. Since the string "a glass bottle" is a noun-noun phrase, glass is a subset of material, and bottle is a subset of container, a new class is created which is a subset of bottles and the characteristic of this class is to be made of glass. The remainder of the sentence is processed in the same way as the previous input sentence, until finally a structure is built to represent the proposition that a member of the set of bottles made of glass is on a member of the set of desks. Again, this proposition is linked to the input sentence by an EXPRESSES relation.

When we input the sentence (again preceded by the **) to the system, it responds with its conclusion as shown here.

** A GLASS BOTTLE IS ON A DESK.
I UNDERSTAND THAT A GLASS BOTTLE IS ON A DESK

To make sure that the system understands the difference between "glass bottle" and "milk bottle", we query the system relative to the item on the desk:

** WHAT IS ON A DESK?
A BOTTLE MADE OF GLASS

We now try "A water bottle is on a bar", but the system cannot fully understand this sentence since it has no knowledge about water. We have not told the system whether water is a fluid or a material. Therefore, rules 3 and 4 are triggered and a node is built to represent this new class of bottles, but no assertion is built concerning the properties of these bottles. Since only three of the four antecedents of rule 6 are satisfied, processing of this rule is suspended. Rule 9 is triggered, however, since all of its antecedents are satisfied, and therefore an assertion is built into the network representing the proposition that a member of a subset of bottles is on a member of the class of bars. Thus the system replies that it has understood the input sentence, but really has not fully understood the phrase "a water bottle" as we can see when we query the system. It does not respond that it is "a bottle for containing water".

** A WATER BOTTLE IS ON A BAR.
I UNDERSTAND THAT A WATER BOTTLE IS ON A BAR
** WHAT IS ON A BAR?
A BOTTLE

Essentially, the phrase "water bottle" is ambiguous for the system. It might mean "bottle for containing water", "bottle made of water", or something else. The system's "representation" of this ambiguity is the suspended rule processing. Meanwhile the parts of the sentence which are "comprehensible" to the system have been processed and stored. After we tell the system "water is a fluid", the system resumes its processing of rule 6 and an assertion is established in the network representing the concept that the function of this latest class of bottles is to contain water. The ambiguity is resolved by rule processing being completed in one of the ways which were previously possible. We can then query the system to show its understanding of what type of bottle is on the bar.

** WATER IS A FLUID.
I UNDERSTAND THAT WATER IS A FLUID

** WHAT IS ON A BAR?
A BOTTLE FOR CONTAINING WATER

This example demonstrates two features of the system: 1) The combined use of syntactic and semantic information in the processing of surface strings. This feature is one of the primary benefits of having not only syntactic and semantic, but also hybrid rules. 2) The use of bi-directional inference to use later information to process or disambiguate earlier strings, even across sentence boundaries.

VIII QUESTION-ANSWERING/GENERATION

The question-answering/generation facility of the NL-system, mentioned briefly in Section 2, is completely rule-based. When a query such as "What is a bottle?" is entered into the system, the sentence is parsed by rules of the core in conjunction with user-defined rules. That is, rule 2 of Section 6 would identify "a bottle" as a GNP, but the top level parse of the input string is accomplished by a core rule. The syntax and corresponding semantics designated by rules 7 and 8 of Section 6 form the basis of the core rule. Our current system does not enable the user to specify the syntax and semantics of questions, so the core rules which define the syntax and consequents of a question were coded specifically for the example of this paper. we intend to pursue this issue in the future. Currently, the two types of questions that our system can process are:

WHAT IS <NP> ?
WHAT IS <RELATION> <NP> ?

Upon successful parse of the query, the system engages in a deduction process to determine which set is a superset of the set of bottles. This process can either find an assertion in the network answering the query or, if necessary, the process can utilize bi-directional inference, initiated in backword-chaining mode, to deduce an answer. In this instance, the network structure dominated by node M75 of figure 3 is found as the answer to the query. This structure asserts that the set of bottles is a subset of the set of containers.

Another deduction process is now initiated to generate a surface string to express this structure. For the purpose of generation, we have deliberately not used the input strings which caused the semantic network structures to be built. If we had deduced a string which EXPRESSES node M75, the system would simply have found and repeated the sentence represented by node M90 of figure 3. We plan to make use of these surface strings in future work, but for this study, we have employed a second "expresses" relation, which we call EXPRESS-2, and rules of the core to

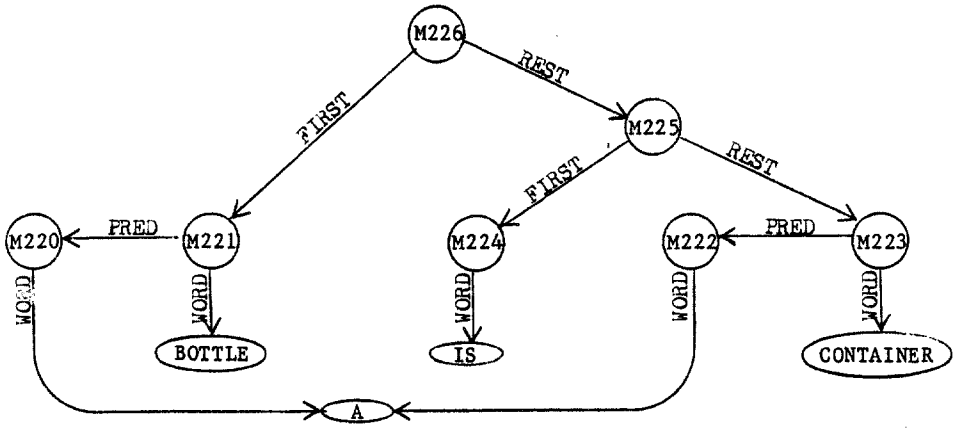

Figure 4. Network representation of a generated surface string.

generate surface strings to express semantic structures.

Figure 4 illustrates the network representation of the surface string generated for node M75. The string "A bottle", dominated by node M221, is generated for node M54 of figure 3, expressing an arbitrary member of the set of bottles. The string "a container", dominated by node M223, is generated to express the set of containers, represented by node M62 of figure 3. Finally, the surface string "A bottle is a container", represented by node M226, is established to express node M75 and the answer to the query. In general, a surface sentence is generated to EXPRESS-2 a given semantic structure by first generating strings to EXPRESS-2 the sub-structures of the semantic structure and by assembling these strings into a network version of a list. Thus the semantic structure is processed in a bottom-up fashion.

The structure of the generated string is a phrase-structured representation utilizing FIRST and REST pointers to the sub-phrases of a string. This representation reflects the subordinate relation of a phrase to its "parent" phrase. The structures pointed to by the FIRST and REST arcs can be a) another list structure with FIRST and REST pointers; b) a string represented by a node such as M90 of figure 3 with BEG, END, and CAT arcs; or c) a node with WORD arc to a word and an optional PRED arc to another node with PRED and WORD arcs. After the structure representing the surface string has been generated, the resulting list or tree is traversed and the leaf nodes printed as response.

IX CONCLUSIONS

Our goal is to design a NLU system for a linguistic theorist to use for language processing. The system's linguistic knowledge should be available to the theorist as domain knowledge. As a result of our preliminary study of a KE approach to Natural Language Understanding, we have gained valuable experience with the basic tools and concepts of such a system. All aspects of our NL-system have, of course, undergone many revisions and refinements during development and will most likely continue to do so.

During the course of our study, we have

a) developed two representations of a surface string: 1) a linear representation appropriate for input strings as shown in figure 1; and 2) a phrase-structured representation appropriate for generation, shown in figure 4;
b) designed a set of SNePS rules which are capable of analyzing the user's natural language input rules and building the corresponding network rules;
c) identified basic concepts essential for linguistic analysis: lexical category, phrase category, relation between a string and lexical constituent, relation between a string and sub-string, the expresses relations between syntactic structures and a semantic structures, and the concept of a variable that the user may wish to use in input rules;
d) designed a set of SNePS rules which can analyze some simple queries and generate a response.

X FUTURE DIRECTION

As our system has evolved, we have striven to reduce the amount of core knowledge which is essential for the system to function. We want to enable the user to define the language processing capabilities of the system, but a basic core of rules is essential to process the user's initial lexicon entries and rules. One of our high priority items for the immediate future is to pursue this issue. Our objective is to develop the NL-system into a boot-strap system to the greatest degree possible. That is, with a minimal core of pre-programmed knowledge, the user will input rules and assertions to enhance the system's capability to acquire both linguistic and non-linguistic knowledge. In other words, the user will define his own input language for entering knowledge into the system and conversing with the system.

Another topic of future investigation will be the feasibility of extending the user's control over the system's basic tools by enabling the user to define the network case frames for syntactic and semantic knowledge representation.

We also intend to extend the capability of the system so as to enable the user to define the syntax of questions and the nature of response.

XI SUMMARY

This study explores the realm of a Knowledge Engineering approach to Natural Language Understanding. A basic core of NL rules enable the NLU expert to input his natural language rules and his lexicon into the semantic network knowledge base <u>in natural language</u>. In this system, the rules and assertions concerning both semantic and syntactic knowledge are stored in the network and undergo interaction during the deduction processes.

An example was presented to illustrate: entry of the user's lexicon into the system; entry of the user's natural language rule statements

into the system; the types of rule statements which the user can utilize; how rules build conceptual structures from surface strings; the use of knowledge for disambiguating surface structure; the use of later information for disambiguating an earlier, partially understood sentence; the question-answering/generation facility of the NL-system.

REFERENCES

1. Haas,N. & Hendrix,G.G., "An Approach to Acquiring and Applying Knowledge", Proceedings of the AAAI, pp. 235-239, 1980.
2. Hayes, P. & Mouradian, G., "Flexible Parsing", Proceedings of the 18th Annual Meeting of the Association for Computational Linguistics, pp. 97-103, 1980.
3. Kaplan, R.M., "A Multi-processing Approach to Natural Language", Proceedings of the National Computer Conference, AFIPS Press, Montvale, NJ, pp. 435-440,1973.
4. Kay, M., "The Mind System", In R. Rustin, ed. Natural Language Processing, Algorithmics Press, New York, pp. 153-188, 1973.
5. Lehnert, W. G., The Process of Question Answering, Lawrence Erlbaum, Hillsdale, NJ, 1978.
6. Martins, J., McKay, D.P., & Shapiro, S.C., Bi-directional Inference, Technical Report No. 174, Department of Computer Science, SUNY at Buffalo, 1981.
7. McCord, M.C., Using Slots and Modifiers in Logic Grammars for Natural Language, Technical Report No. 69A-80, Univ. of Kentucky, rev. October, 1980,
8. McKay, D.P. & Shapiro, S.C., "MULTI - A LISP Based Multiprocessing System", Conference Record of the 1980 LISP Conference, Stanford Univ., pp. 29-37, 1980.
9. Pereira, F.C.N. & Warren, D.H.D., "Definite Clause Grammars for Language Analysis - A Survey of the Formalism and a Comparison with Augmented Transition Networks", Artificial Intelligence, pp. 231-278, 1980.
10. Robinson, J.J., "DIAGRAM, A Grammar for Dialogues", CACM, pp. 27-47, January, 1982.
11. Shapiro, S.C., "The SNePS Semantic Network Processing System". In N. Findler, ed. Associative Networks - The Representation and Use of Knowledge by Computers, Academic Press, New York, pp. 179a-203, 1979.
12. Shapiro, S.C., "Generalized Augmented Transition Network Grammars for Generation from Semantic Networks", Proceedings of the 17th Annual Meeting of the Association for Computational Linguistics, pp. 25-29, 1979.

XII APPENDIX - NL CORE GRAMMAR

The following grammar is a definitive description of the language in which the user can enter linguistic statements into the semantic network. The Backus-Naur Form (BNF) grammar is used in this language definition.

Notational conventions:
- Phrase in lower case letters explains the word required by the user
- Standard grammar metasymbols:
  - <> enclose nonterminal items
  - | for alternation
  - [] enclose optional items
  - () for grouping
  - Space represents concatenation
- Concatenation has priority over alternation

```
<LEX-STMT> ::=
     '<WORD> IS (A|AN) (L-CAT|<L-CAT-MEMBER>)
<RULE>  ::=     IF <ANT-STMT> THEN <CQ-STMT>
<ANT-STMT> ::=    <ANT-STMT> AND <ANT-STMT>
       | A STRING CONSISTS OF <STR-DESCRIPTION>
       | <STMT>
<CQ-STMT>  ::=    <CQ-STMT> AND <CQ-STMT>
       | THE STRING IS <G-DET> <STRING-NAME>
       | THERE EXISTS A <CONCEPT-WORD> <VAR>
       | <STMT>
<STMT>   ::=      <CL-REF> <REL-REF> <CL-REF>
       | THE <STRING-NAME> EXPRESSES <CL-REF>
       | THE <STRING-NAME> EXPRESSES THIS LAST
         PROPOSITION
       | THE <FUN-CHAR-WORD> OF <CL-REF> IS TO
         BE <FUN-CHAR-VERB> <CL-REF>
<STR-DESCRIPTION> ::=
  <STR-DESCRIPTION> FOLLOWED BY <STR-DESCRIPTION>
       | <G-DET> <LEX-NAME> [<LABEL-PHRASE>]
       | THE WORD '<LITERAL>
<LABEL-PHRASE>  ::=     CALLED <DET> <LABEL>
<LEX-NAME>  ::=  any lexical category name
<LABEL> ::=  any name or label
<STRING-NAME>  ::=  any string category name
<REL-REF>  ::=    IS A (SUBSET|MEMBER) OF
       | HAS THE <REL-WORD> TO
<CL-REF>   ::=    THE <CONCEPT-WORD> <VAR>
       | THE CLASS NAMED BY THE <NAME>
       | a member of an L-CAT category
<FUN-CHAR-WORD>  ::=  (FUNCTION|CHARACTERISTIC)
<FUN-CHAR-VERB>  ::=  any verb
<NAME> ::=  name of a string phrase or the
          constituent of a string phrase
<VAR> ::= any member of the category VARIABLE
<G-DET>   ::=  A  |  AN  |  ANOTHER
<DET>     ::=  <G-DET>  |  THE
<REL-WORD> ::= a member of L-CAT which should
          denote "relation"
<WORD> ::= any word
```