Structure from Anarchy: Meta Level Representation of Expert System Propositions for Natural Language Interfaces.[1]

Galina Datskovsky Moerdler
Department of Computer Science
Columbia University
New York, N.Y. 10027
phone: (212) 280-8177
Arpanet: galina@columbia.arpa.edu

Abstract:

In this paper we describe a meta level representation used for mapping natural language input into propositions of an expert system. This representation is based on verb classes that are structured hierarchically, with more general information encoded in the top level nodes and more specific information in the lower level nodes. Because of its structure, the representation is able to provide a detailed classification of the propositions, supplying a basis for defining semantics. It allows the system to answer questions about relationships between propositions without inferencing, as well as to answer questions the expert system could not previously handle.

1 Introduction

A great deal of work has been done in constructing natural language interfaces to well structured underlying systems, such as data base systems. These natural language interfaces generally make use of an assumed system structure, such as a schema, to define semantics [Martin, Appelt and Pereira 83; Grosz et. al. 85] [Woods et. al. 72; Woods 73] [Kaplan 79]. On the other hand, almost no effort has been made in constructing natural language interfaces to systems that do not have such an extensive description, e.g. expert systems[2]. The lack of such a schema means that there is no easy way to obtain information about propositions[3] of the underlying system. Thus, in order to build a natural language interface to expert systems the semantic interpreter must be able to provide the necessary structure. In an earlier paper [Datskovsky Moerdler et.al. 87] we briefly described a semantic interpreter that maps user statements into facts of an expert system, as well an inference engine for expert systems that can efficiently utilize this input. In this paper we discuss the meta level description of the expert system propositions, similar to a schema of a data base, utilized by the semantic interpreter and show how this structure is used in processing of user questions.

Our structure consists of a group of hierarchies which are formed from verb categories. The hierarchies provide a grouping of the propositions of an expert system by topic. For example, all propositions that deal with interpersonal relationships are grouped under one hierarchy, while those dealing with transfer of possession are grouped under another. The meaning of a proposition is specified step by step, as the hierarchy is traversed, thus allowing for mapping of various sentences, or parts of sentences into the propositions. To test our theories, the approach is currently being implemented as a front end to a small expert system that deals with personal income tax matters[4].

2 Expert Systems vs. Data Base Systems

Many techniques used in building natural language interfaces for data base systems can not carry over into the expert system domain because of the differences between the two underlying systems.

[2] Throughout this paper we are only concerned with expert systems that must communicate with their users in order to gather data before giving advice, such as Mycin [Shortliffe 76]

[3] In the rule of the form IF A and B then C, A, B and C are propositions. The terms proposition and fact are used interchangeably throughout this paper.

[4] In particular, we are implementing our ideas on one module of Taxpert [Ensor et. al. 85], an expert system designed in conjunction with AT&T Bell Laboratories. This module helps users determine whether they can or can not claim an individual as a dependent [Datskovsky Moerdler et.al. 87]

[1] This research was partially supported by Office of Naval Research grant N00014-82-K-0256.

A semantic interpreter for a data base system usually relies on the regular structure of the data base as encoded in the schema describing it. The schema usually describes the fields and tables of a given data base and provides such information as the key field of a table, the type of data found in each field, relationships between the fields (e.g. all the fields of a given table describe its key field), etc.. The relationships between tables are indicated by similarities and differences of their fields. A typical natural language interface associates semantics of nouns, adjectives and verb phrases of a natural language with fields of a data base. Verbs of the natural language are also associated with actions that can be performed on the tables of a given data base, such as *Find* and *Join*.

In contrast, no schema or description is available for expert systems. The propositions of an expert system may have arbitrary meanings. No relationship between the propositions is clearly defined. Although meta level structures have been built by systems such as Theresias [Davis 78], these structures are inadequate for defining semantics. Theresias provides such information as the relationships between antecedents and consequents of rules, groupings of rules by their left hand sides, etc.. Only one type of representation, schemata, actually gives a shallow (3 levels) description of propositions (as opposed to rules). However, this information is not sufficient for complete semantic definition and a more complex structure is required.

Another major difference is in the function of the two systems. A data base system is not expected to know or solve a user's problem, but only supply the information that the user requests. Consequently, an interface to a data base system must be able to retrieve information requested by the user. In figure 1 we present a typical interaction between Lifer-Ladder [Hendrix et. al. 78] and a user. The questions here involve retrieval of information from a naval data base.

user: What is the length and hull number of the
  Constellation?
system: ....
user: the home port?
system: ...

Figure 1: Interaction between a user and LIFER-LADDER (Taken from [Tennant 81])

On the other hand, an expert system is designed to be a problem solver. A user consults it about an issue and it must gather information in order to advise him. In figure 2 we present a typical interaction between a user and the Mycin [Shortliffe 76] system (taken from transcripts generated by the author). First, in questions 1-6 the system gathers information about the patient, such as age, sex, lab analysis, etc., and then, after many more questions not shown in the figure, makes a recommendation based on the gathered data. Note that the menu interface predefines the order in which information is entered into the system, whereas with a natural language interface, information is entered in no particular order, i.e it may be imbedded in every user input. The addition of new information with every user statement means that the expert system has to pose fewer questions and that the natural language interface must be responsible for managing all the new information. Further, the interface may have to derive information not only from user statements, but also from questions. This means that it has to derive the problem to solve, as well as facts that can be used for its solution from any given question, and add these facts to the data base (or working memory). The action of extracting a goal[5] and adding facts at the same time has no analogy in a data base system, but would be similar to allowing the user to query and update the data base at the same time.

```
  1) Patient's name:
  ** PT244
  2) Age:
  ** 80 YEARS
  3) Sex:
  ** MALE
...
  6) From what site was the specimen
     for CULTURE-1 taken?
    ** SPUTUM
```

More similar Questions Follow and finally a recommendation is made

```
[Determining which drugs are
desirable for use against the
(Klebsiella- pneumoniae...]

[REC-1] My preferred therapy
recommendation is as follows:
...
```

Figure 2: Mycin Transcript

[5] Throughout this paper goals refer to the goals the expert system must prove, not long term user goals.

3 The Structure in More Detail

To translate user input into facts and goals of an underlying expert system, a structure that is able to provide a foundation for the translation is necessary. This structure must provide the meaning of the expert system propositions, relationships between them and supply a means of mapping semantics of words and phrases into those propositions. It is also desirable that such a structure be general, and hence to some extent transportable from one system to another.

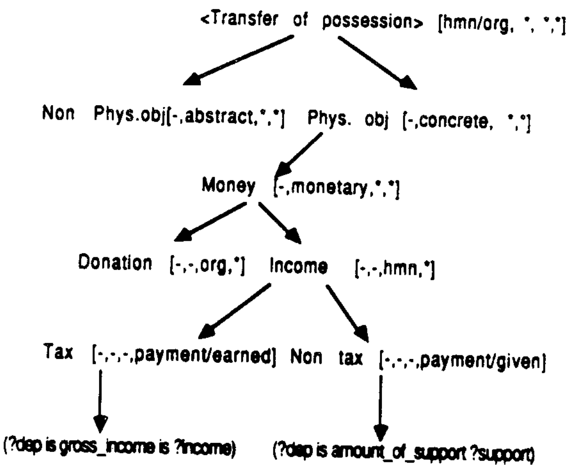
<Transfer of possession> [hmn/org, *, *,*]
Non Phys.obj[-,abstract,*,*]
Phys. obj [-,concrete, *,*]
Money [-,monetary,*,*]
Donation [-,-,org,*]
Income [-,-,hmn,*]
Tax [-,-,-,payment/earned]
Non tax [-,-,-,payment/given]
(?dep is gross_income is ?income)
(?dep is amount_of_support ?support)

Figure 3: Partial Tree formed for the *Transfer of possession* category[6]

Our structure consists of a group of hierarchies formed from classes of verbs. We have analyzed over 90 verbs most common to our domain and classified them into 13 categories[7]. These categories can be used in any domain that requires the verbs belonging to them, because they are derived from general properties of the verbs[8], thus allowing for a degree of transportability. Each verb category is organized hierarchically where each node of a hierarchy is derived from the meanings of one or more verbs. A number of selectional restrictions is attached to each node indicating constraints on the agent, patient, object and modifier of an input sentence (not all four restrictions are specified for every category). The hierarchies group propositions of an expert system by topic. The leaves of the hierarchies contain either expert system facts or pointers to other hierarchies, thus forming a connected forest. The top level nodes of the hierarchies provide general classes into which a group of propositions of an expert system might fall. At the lower levels of the hierarchies the propositions are separated into more specific subclasses of a given parent node, thus further specifying their meanings. At the lowest level, each node points to only one proposition thus uniquely defining it within its class.

For example, figure 3 shows the partial hierarchy for the Transfer of Possession category. The top level node of the hierarchy is derived from the properties of the verbs of the general class of Transfer of Possession. Verbs from that class have pointers to this node and all the propositions that deal with transfer of possession can be accommodated by this node and the nodes below it. The selectional restrictions on this node indicate that the transfer is initiated by either a human or an organization and that the beneficiary of the transfer, the object being transferred, as well as any modifiers can be unspecified until some lower level. The two nodes at the next level further divide the class of transfer of possession verbs and predicates into those dealing with physical object transfers and non physical object transfers. The [-] in the selectional restrictions indicate that the feature is inherited from the parent node. The restrictions on the two nodes also further specify that the object being transferred must be concrete in order to take the Phys Obj link and abstract in order to take the Non Phys Obj link. At the next level, the concept of physical object transfers (as embodied by the Phys Obj node) is further specified. In this example only one of its children, the *Money* node is shown[9]. Again, verbs dealing specifically with money transfers may point directly to this node. The restriction on the object of the transfer must be monetary in order for this node to be chosen during parsing. This node is further subdivided into Donation and Income, where the distinction is made based on the recipient of the transfer, since donations are normally given to organizations, and income to people. Next, Income can come in two forms, Taxable and Non Taxable, as indicated by the selectional restrictions of the objects

[6] In the figure, * stands for wild card, and - means that the feature is inherited from the parent node.

[7] This is not an absolute number. More categories may be needed in other domains where a greater number of verbs is necessary.

[8] These categories are based on works in linguistics, e.g. [Osgood 79] and on Roget's Thesaurus. For a more detailed description of the categories see [Datskovsky Moerdler et.al. 87]

[9] The node has 2 other children in the complete tree.

of the transfer, and finally, the bottom level of the hierarchy contains expert system propositions. The propositions (?dependent is gross_income ?income) and (?dependent is amount_of_support ?support) belong to a general class of *Transfer of Possession*, and a more specific class *Income*, indicating that both propositions describe a type of income that is generally transferred from one party to another. However, because one deals with taxable income and the other with non taxable income, these propositions are further subdivided into subclasses at the next level.

This kind of gradual division of propositions into subclasses not only provides a means for mapping user input into facts and goals of an expert system, but also allows the system to answer questions about relationships between the propositions, often without any inferencing. In addition, it allows the system to make meta level inferences it could not make without the structure. In the next section we present a brief description of the parsing algorithm and illustrate it with an example.

3.1 Parsing Algorithm: Overview and Example.

During parsing, an appropriate hierarchy is selected according to the definition of the verb in the system's dictionary, where each verb can point to any level in a hierarchy, and a selectional restriction based algorithm is used to traverse the hierarchy with the nouns of the sentence guiding the parser down the hierarchy, until an expert system proposition is reached. The information for this algorithm is encoded into each hierarchy, with the restrictions on the arguments of the verbs based on noun features derived from Roget's thesaurus. The system is currently being implemented in Common lisp on a Symbolics Lisp Machine. It uses an ATN parser which has been modified to call the semantics at various points before deciding which arc to take next. Syntax and semantics run in parallel, with syntax providing a deep structure of a sentence, and semantics supplying information for modifier attachment. Although the verb hierarchies are the primary source of facts, some facts are derived directly from the noun features.

As an example of how the natural language interface derives both propositions and goals from *Yes/No* questions posed by the user consider the question *Can I claim my son who earns a salary of $2000?*. A trace of the system execution of this sentence is shown in appendix I. The trace shows the nodes of the different hierarchies considered by the algorithm and where the interaction between syntax and semantics occurs. It also shows all the predicates derived by the system and a complete syntactic parse. In *yes/no* questions the goal is generally indicated by the main verb. The syntactic parser identifies *claim* as the main verb of the sentence. The verb *claim* is defined in the system's dictionary as *Classification* <+> *Dependency*[10], indicating that the verb belongs to the general category of *Classification* and a more specific subnode of that category, *Dependency*. The <+> indicates that the syntactic subject of the sentence is the semantic agent. Based on the definition of the verb the algorithm enters the *Classification* hierarchy at the *Dependency* node, as demonstrated in statements 1 and 2 of the system trace, thus limiting the choice of propositions that this input can map into to the general category of *Classification* and the subclass *Dependency* (see figure 4). Since only one proposition, (?user can_claim ?dependent), falls into this classification, it is derived as the goal, indicating that the user wants to know whether he can or can not claim a dependent (the variables of the proposition will later be instantiated with the appropriate values).

The additional information in the relative clause states that the dependent earns a salary of $2000, or (?dependent is gross_income ?income). To derive this additional information, the system selects a hierarchy based on the meaning of the verb of the relative clause. The verb *to earn* is defined in the dictionary as *Transfer of possession* <+>, so the algorithm enters the Transfer of Possession hierarchy (shown in figure 3). The choice of propositions that this input can map into is now limited to those in the general class of *Transfer of Possession*. Next, because of the feature *concrete* of the object (two thousand dollars) of the sentence the algorithm selects Phys Obj as the next node to consider. Based on the feature monetary of the word *dollars* the Money node is selected next. The Income node is chosen because the recipient of the money has the feature *human*, and finally, because *salary* is defined as *payment/earned*, the node *Tax* is selected, since earned payments are generally taxable. Finally (?dependent is gross_income ?income) is added to the working memory. The variables ?dependent and

[10] Although there are other meanings of the verb, this is the most frequently used meaning in the tax domain, so the system tries this category first.

?income are later instantiated with son and $2000 respectively. The derivation of this predicate can be seen in statement 5-13 of the system trace in appendix I.

Propositions can also be derived from certain noun phrases. In this example, the phrase *my son* indicates the existence of a child-parent relationship. The system then checks for agreement between the head pronoun *I* and the possessive *my* and once this agreement is verifies maps the representation of this relationship into the proposition (?dependent is son_of ?user), as shown in statement 4 of the trace.

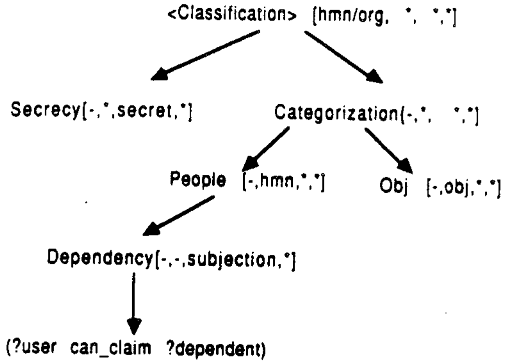
<Classification> [hmn/org, *, *,*]
Secrecy[-,*,secret,*]
Categorization(-,*, *,*]
People [-,hmn,*,*]
Obj [-,obj,*,*]
Dependency[-,-,subjection,*]
(?user can_claim ?dependent)

Figure 4: Partial Tree formed for the *Classification* category.

The mapping of natural language into propositions of the expert system as demonstrated above is possible because of the classification of propositions and descriptions of their meanings provided by the hierarchies. Note that the hierarchies are used to define semantics of words of the natural language e.g. the verb *to earn* is directly related to the meta level structure, or the *Transfer of Possession* hierarchy. The structure given by the hierarchies also provides a description of the propositions and gives similarities and differences between them. For example, both propositions (?dependent is gross_income ?income) and (?dependent is amount_of_support ?support) would have the general properties of the class *Income*, with unique features of their particular subclasses *Tax* and *Non Tax*. This unique classification allows for the mapping of the input in the above example into the appropriate proposition. It also allows the system to answer questions about the differences between the two propositions, as shown in the next section. Another benefit of this representation is that it provides the system with a way of dealing with input sentences like *My son earns $2000*, that do not completely specify a particular proposition. The sentence indicates that the desired proposition is in the class *Income*, and the system can proceed to specify the appropriate subclass by posing questions to the user without any additional inferencing on the part of the expert system. This particular capability of the algorithm will be discussed in greater detail in future work.

3.2 Other Questions that can be Answered from the Hierarchies

The hierarchies allow the system to handle a number of questions that could not be previously handled by the expert system, and answer other questions without invoking the inference process. In particular, these include questions that deal with relationships between facts and comparisons between sessions, as well as questions requiring general information.

User: My daughter receives a stipend of $5000, while my son gets a salary of $2000.
WHY is my daughter's tax situation different from my son's?

System: Your daughter's stipend is non taxable income.

[Answered by looking at the *Income* node of the *Transfer of Possession* hierarchy, where the two paths diverged.]

Figure 5: A Question Answered from the Transfer of Possession Hierarchy

As an example of questions that can be answered without invoking the inference process, consider the hypothetical example in figure 4 where the user tells the expert that his daughter receives a stipend of $5000, which translates into the proposition (daughter is amount_of_support 5000), since *stipend* is defined in the dictionary as *payment-given*. The fact that his son has a salary of $2000 translates into the proposition (son is gross_income 2000). To answer the *WHY* question the system could check where the derivation paths for the two sets of inputs diverged, and the difference between the two subclasses would constitutes the answer. In this example the paths diverge at the *Income* node of the *Transfer of possession* hierarchy, thus the answer can be supplied by simply examining the hierarchy.

The question in the first example required both a comparison between two derivation paths as well as the knowledge of the differences between two propositions. As a second example consider the question *What kinds of family relationships are recognized by the tax code?* This question is about general properties of the tax code and could not be handled by the expert system without the natural language interface, even though all the necessary information was already available in the system. To answer this question it is enough to search the hierarchies for a *Relationship* node with a child node that describes family relationships. Such a parent-child pair is found in the *Possession* hierarchy (see figure 6). The answer returned would consist of all the children found under this pair.

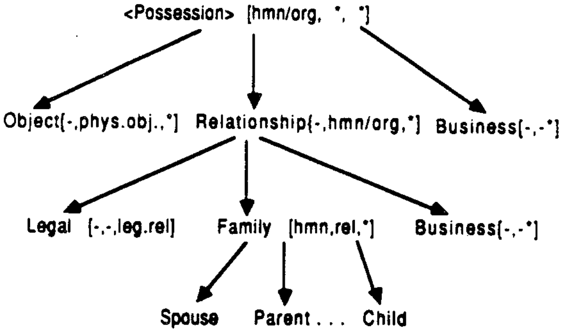

Figure 6: Partial Tree formed for the *Possession* category.

The question handling algorithm is currently under design. To process WH questions the system must first be able to determine whether it can be answered from the hierarchies, or whether the inference engine of the expert system should be invoked. Many of the necessary clues that indicate the question type have been identified, however there is still some more work to be done on this, as well as on the implementation of the module. It is clear, however, that the hierarchies give the system the ability to handle many more types of questions than the expert system alone could handle, and in many instances allow questions to be answered without invoking the inference process of the expert system.

4 Comparison with Previous Work: NLIs to Expert Systems and Other Work in Semantics

There has been some effort to construct natural language interfaces to expert systems, namely Prospector [Duda et. al. 79] and Xcalibur [Carbonell et.al. 83; Carbonell and Hayes 84]. Prospector is one of the first expert systems to communicate with its users in natural language. During the consultation the user simply describes what has been discovered at a given site by using patterns, built with the help of the Lifer [Hendrix et. al. 78] system, of the form "There is <deposit>", "There may be <deposit>", etc. There is not much published information that describes Prospector's natural language module. We can only hypothesize that a very simple and limited set of sentences is accepted by the system based on sample system sessions.

Xcalibur's interaction with the user greatly resembles that of a natural language interface to a data base system. Unlike systems such as Mycin, Xcalibur does not do most of the asking. It is not responsible for solving the user's problem, but rather the user has to know what he wants and query accordingly. Most expert systems are designed to solve a user's problem, and this property must be reflected in the interface. Xcalibur does not seem to be suitable as an interface for such systems because it is designed to retrieve information rather than solve a problem.

4.1 Other work in Semantics

Our work draws on Palmer's [Palmer 85], but is different from it in several ways. Palmer's Inference-driven semantic analysis is specifically designed for a finite, well-defined, i.e. limited domain. The main element of her approach is a set of partially instantiated logical terms, or semantic propositions, which capture the different relationships that can occur in a given domain. Unlike Palmer's work, our interpreter deals with a complex real world domain. It also makes a greater separation between domain specific and domain independent knowledge to allow for a degree of transportability. Also, while our semantics provides a hierarchical organization, Palmer's does not.

Other work that has influenced our own also includes that of Graeme Hirst [Hirst 83] and Steve Lytinen [Lytinen 84]. One of the main differences between our work and the work mentioned above (including Palmer's) is that our semantics imposes a structure on top of an unstructured underlying system, which is not the goal of the work mentioned above.

5 Possible Automation of Hierarchy Design

The lack of automatic construction of the hierarchies and automatic classification of propositions in them is currently a limitation in our system. If, for a given domain, a certain tree has to be extended, such extension will have to be done by hand. Also, propositions have to be hand encoded in the hierarchies. This makes transportability to other domains more difficult. After the top level categories are selected, the rest of the nodes of the hierarchies and the propositions, as well as the selectional restrictions can not be done interactively. However, we feel that the hierarchies lend themselves to automation construction by an Expert System Expert, because they are based on the linguistic properties of the verbs in the domain, as well as on the knowledge of the meanings of propositions.

In the future, we would like to design a customization phase similar to that of Team [Martin, Appelt and Pereira 83; Grosz et. al. 85] and Teli [Ballard 86]. With such a customization phase, a given expert, such as an Expert Systems Expert, can spend several hours automatically building up the necessary parse trees for a given domain. We feel that such a module would greatly enhance the system and make it much more usable.

6 Conclusions and Future Research

In this paper we presented a structure for expert systems, similar to a data base schema, that facilitates construction of natural language interfaces. This structure is based on verb classification and hierarchical structuring within each category. The hierarchies provide a grouping of expert system propositions into classes, thus capturing the similarities and differences between the propositions. This grouping provides a mapping between user input and the propositions of the expert system, as well as a mechanism for dealing with several types of questions without additional expert system inferencing. The structure provides a mechanism for answering questions that could not be previously handled by the expert system. It also provides a flexible and somewhat general mapping allowing for a degree of transportability.

One of our primary goals is to complete the implementation of our ideas. Processing of statements and yes/no questions has been fully implemented and the work on paragraph parsing and handling of semantically incomplete input is our current focus. In the future we plan to add such features as complete WH question processing and an automatic hierarchy construction algorithm.

7 Acknowledgments

I would like to thank my advisor, Kathleen McKeown for all her help and guidance in this work and Robert Ensor of AT&T for his helpful comments.

Appendix I

```
        (process '((can I claim my son who earns a salary of
twothousand dollars)))

        1. In Tree: CLASSIFY

        2. Considering the children of DEPENDENCY

        3. the proposition that was derived is

        ((?USER |CAN_CLAIM| ?DEPENDENT))

        back to syntax...

        4. the proposition derived from the noun phrase

        (MY SON) is

        (?DEPENDENT IS |SON_OF| ?USER)

        5. In Tree: TRANS_OF_POS

        6. Considering the children of TRANS_OF_POS

        back to syntax...

        7. Considering the children of TRANS_OF_POS

        back to syntax...

        8. Considering the children of TRANS_OF_POS

        9. Considering the children of |PHYS_OBJ|

        10. Considering the children of MONEY

        11. Considering the children of INCOME

        12. Considering the children of TAX

        13. the proposition that was derived is

        ((?DEPENDENT     IS     |GROSS_INCOME|
?INCOME))

        back to syntax...
        ((S (SUBJ (NP (DET NIL) (DESCRIBERS NIL) (HEAD ((PRON
I))) (NUMBER SING)(CONJ NIL) (SEM (HUMAN) NIL)) (QUALIFIERS
NIL) (QUESTION NO)(CASE OBJECTIVE))) (AUXS (CAN)) (TENSE
PRES) (MAINVERB CLAIM) (SEM-MVERB ((CLASSIFY +
DEPENDENCY))) (ADVERB NIL) (IND-OBJ NIL) (SUBCONJ NIL)
(D-OBJ (NP (DET MY) (DESCRIBERS NIL)(HEAD ((NOUN SON)))
(NUMBER SING)(CONJ NIL) (SEM ((HUMAN MALE RELATIVE
CHILD) NIL)) (QUALIFIERS ((S (SUBJ (NP (DET MY) (DESCRIBERS
NIL) (HEAD ((NOUN SON))) (SEM ((HUMAN MALE RELATIVE
CHILD) NIL)) (QUESTION NO)))(AUXS NIL) (TENSE PRES)
(MAINVERB EARN) (SEM-MVERB ((TRANS_OF_POS +))) (ADVERB
NIL) (IND-OBJ NIL) (SUBCONJ NIL)(D-OBJ (NP (DET A)
(DESCRIBERS NIL) (HEAD ((NOUN SALARY))) (NUMBER
SING)(CONJ NIL) (SEM ((CONCRETE MONETARY PAYMENT-
EARNED) NIL)) (QUALIFIERS ((PP (PREP OF) (PREP-OBJ (NP (DET
NIL) (DESCRIBERS (TWOTHOUSAND)) (HEAD ((NOUN DOLLAR)))
(NUMBER PLURAL)(CONJ NIL) (SEM ((MONETARY) (CONCRETE
NUMBER))) (QUALIFIERS NIL)(QUESTION NO) (CASE
OBJECTIVE)))))) (QUESTION NO) (CASE OBJECTIVE)))(ADJCOMP
NIL) (MODIF NIL) (TYPE WH-RELATIVE) (QUESTION-ELEMENT
NIL)(VOICE ACTIVE) (CONJ NIL) (CSENT NIL)))) (QUESTION NO)
(CASE POSSESSIVE))) (ADJCOMP NIL)(MODIF NIL) (TYPE
INTERROGATIVE) (QUESTION-ELEMENT (YES-NO)) (VOICE
ACTIVE)(CONJ NIL)(CSENT NIL)))

THE GOAL IS: (?USER |CAN_CLAIM| ?DEPENDENT)
```

References

[Ballard 86] Ballard B. Semantic Acquisition in TELI: A Transportable, User-Customized Natural Language Processor. In *Proceedings of twenty first ACL conference*. 1986.

[Carbonell and Hayes 84] Carbonell J., Hayes, P. *Recovery Strategies Parsing Extragrammatical Language*. Technical Report CMU-CS-84-107, Carngie-Mellon University, 1984.

[Carbonell et.al. 83] Carbonell J., Boggs W.M., Mauldin M., Anick, P. *Xcalibur Project Report 1*. Technical Report CMU-CS-83-143, Carngie-Mellon University, Digital Equipment Corp., 1983.

[Clancey 83] Clancey, W. The Epistimology of a Rule-Based Expert System - a Framework for Explanation. *Artificial Intelligence 20*, 1983.

[Datskovsky Moerdler et.al. 87] G. Datskovsky Moerdler, K. McKeown, J.R. Ensor. Building Natural Language Interface to Expert Systems. In *Proceedings of the IJCAI*. 1987.

[Davis 78] Davis, R. Knowledge Acquisition in Rule-Based Systems-Knowledge About Representation as a Basis for System Construction and Maintanance. *Pattern Directed Inference Systems*. Academic Press, 1978.

[Duda et. al. 79] Duda, R., Gasching, J., Hart, P. Model Design in the Prospector Consultant System for Mineral Exploration. In Michie, D. (editor), *Expert Systems in the micro-electronic age*. Edinburgh University Press, 1979.

[Ensor et. al. 85] Ensor, Gabbe and Blumenthal. Taxpert -- A Framework for Exploring Interactions Among Experts. 1985.in preparation.

[Grosz et. al. 85] Grosz, B., Martin, P., Appelt, D., Pereira, F.,*Team: An Experiment in the Design of Transportable Natural Language Interfaces*. Technical Report, SRI International, 1985.

[Hendrix et. al. 78] Hendrix, G., Sacerdoti, E., Sagalowicz D., Slocum J.*Developing a Natural Language Interface to Complex Data. ACM Transactions on Database Systems*, 1978.

[Hirst 83] Hirst, G. *Semantic Interpretation Against Ambiguity*. PhD thesis, Brown University, 1983.

[Kaplan 79] Kaplan, S.J. *Cooperative Responses From a Portable Natural Language Data Base Query System*. PhD thesis, University of Pennsylvania, 1979.

[Levin 85] Levin, B. Lexical Semantics in Review: An Introduction. In Levin, B. (editor), *Lexical Semantics in Review*. MIT, 1985.

[Lytinen 84] Lytinen S.L. *The Organization of Knowledge in a Multi-lingual Integrated Parser*. PhD thesis, Yale University, 1984.

[Martin, Appelt and Pereira 83] Martin, P., Appelt, D., Pereira, F. Transportability and Generality in a Natural Language Interface System. In *Proceedings of IJCAI*. 1983.

[Osgood 79] Osgood, Charles, E. *Focus on Meaning Volume I: Explorations in Semantic Space*. Mouton Publishers, 1979.

[Palmer 83] Palmer, M. Inference-Driven Semantic Analysis. In *Proceedings of the AAAI*. 1983.

[Palmer 85] Stone Palmer, M. *Driving Semantics for a Limited Domain*. PhD thesis, University of Edinburg, 1985.

[Pollack 83] Pollack, M.E. *Generating Expert Answers Through Goal Inference*. Technical Report, SRI International, 1983.

[Shortliffe 76] Shortliffe, E.H. *Mycin: A rule-based computer program for advising physicians regarding anitimicrobial therapy selection*. PhD thesis, Stanford University, 1976.

[Tennant 81] Tennant, H. *Natural Language Processing*. Petocelli Books, USA, 1981.

[Webber 71] Nash-Webber, B., Verbs of Composition. 1971.Harvard University, 1971.

[Woods 73] Woods, W.A. An Experimental parsing System for Transition Network Grammars. In Rustin (editor), *Natural Language Processing*. Algorithmic Press, 1973.

[Woods et. al. 72] Woods W., Kaplan R., Nash-Webber B. *The Lunar Sciences Natural Lnaguage Information System: Final Report*. Technical Report 2378, BBN, Cambridge, Mass, 1972.