August 1980

# Paraphrasing Using Given and New Information in a Question-Answer System

Kathleen R. McKeown
*University of Pennsylvania*

# Paraphrasing Using Given and New Information in a Question-Answer System

**Abstract**
The design and implementation of a paraphrase component for a natural language question-answer system (CO-OP) is presented. A major point made is the role of given and new information in formulating a paraphrase that differs in a meaningful way from the user's question. A description is also given of the transformational grammar used by the paraphraser to generate questions.

**Comments**
University of Pennsylvania Department of Computer and Information Science Technical Report No. MS-CIS-80-13.

MS-CIS-80-13

UNIVERSITY OF PENNSYLVANIA

THE MOORE SCHOOL

# Paraphrasing Using Given and New Information in a Question-Answer System

Kathleen R. McKeown

This work was partially supported by an IBM fellowship, NSF grant MCS 78-08401, and NSF grant MCS 79-19171.

Presented to the Faculty of the College of Engineering and Applied Science (Department of Computer and Information Science) in partial fulfillment of the requirements for the degree of Master of Science in Engineering.

Philadelphia, Pennsylvania

August, 1979

ABSTRACT

Paraphrasing Using Given and New Information
in a Question-Answer System

Kathleen R. McKeown

Supervisor: Dr. Aravind K. Joshi

The design and implementation of a paraphrase component for a natural language question-answer system (CO-OP) is presented. A major point made is the role of given and new information in formulating a paraphrase that differs in a meaningful way from the user's question. A description is also given of the transformational grammar used by the paraphraser to generate questions.

TABLE OF CONTENTS PAGE

I. INTRODUCTION . . . . . 2

II. OVERVIEW OF THE CO-OP SYSTEM . . . . . 4

III. THE CO-OP PARAPHRASER . . . . . 8

IV. LINGUISTIC BACKGROUND. . . . . 10

V. FORMULATION . . . . . 14

VI. IMPLEMENTATION OVERVIEW . . . . . 16

A. THE PHRASE STRUCTURE TREE . . . . . 17

B. DIVIDING THE TREE. . . . . 19

C. FLATTENING . . . . . 21

D. TRANSFORMATIONS . . . . . 24

E. CONJUNCTION AND DISJUNCTION . . . . . 29

F. NUMERICAL MODIFICATION . . . . . 35

G. Translation. . . . . 37

VII. GENERATION . . . . . 39

VIII. FUTURE RESEARCH . . . . . 44

IX. RELATED RESEARCH . . . . . 48

X. CONCLUSIONS . . . . . 51

XI. APPENDIX A . . . . . 53

XII. APPENDIX B . . . . . 58

XIII. APPENDIX C. . . . . 60

XIV. REFERENCES . . . . . 63

## ACKNOWLEDGEMENTS

This work was partially supported by an IBM fellowship, NSF grant MCS 78-08401, and NSF grant MCS 79-19171.

I would like to thank my advisor Dr. Aravind K. Joshi, for the tremendous amount of time and advice he has given me during the making of this thesis. I would also like to thank Jerry Kaplan for his advice during the implementation stages of the paraphraser and Dr. Bonnie Webber for her invaluable comments on the style and content of the thesis.

Some graduate students of the Moore School deserve special thanks for reading and commenting on various versions of the thesis and discussing critical points. They are: Tom Williams, David Bourne, Joe O'Rourke, and David Raab.

Finally, thanks goes to the members of the house at 4103 Chestnut St. for their support and friendship during the entire period of the project.

## I. INTRODUCTION

In a natural language interface to a database query system, a paraphraser can be used to ensure that the system has correctly understood the user. Such a paraphraser has been developed as part of the CO-OP system (KAPLAN 79). In CO-OP, an internal representation of the user's question is passed to the paraphraser which then generates a new version of the question for the user. Upon seeing the paraphrase, the user has the option of rephrasing her/his question before the system attempts to answer it. Thus, if the question was not interpreted correctly, the error can be caught before a possibly lengthy search of the database is initiated. Furthermore, the user is assured that the answer s/he receives is an answer to the question asked and not to a deviant interpretation of it.

The idea of using a paraphraser in the above way is not new. To date, other systems have used canned templates to form paraphrases, filling in empty slots in the pattern with information from the user's question (WALTZ 78; CODD 78). The CO-OP paraphraser differs from these earlier systems in that a systematic method to generate paraphrases has been adopted. In CO-OP, a transformational grammar is used to generate the paraphrase from an internal representation of the question. Moreover, the CO-OP paraphraser generates a question whose form differs in a meaningful way from that of

the original question. It makes use of a distinction between given and new information to indicate to the user the existential presuppositions made in her/his question.

## II. OVERVIEW OF THE CO-OP SYSTEM

The CO-OP system is aimed at infrequent users of database query systems. These casual users are likely to be unfamiliar with computer systems and unwilling to invest the time needed to learn a formal query language. Being able to converse naturally in English enables such persons to tap the information available in a database.

In order to allow the question-answer process to proceed naturally, CO-OP follows some of the "co-operative principles" of conversation (GRICE 75). In particular, by using these principles, the system attempts to find meaningful answers to questions having negative responses. The motivation for the approach was based on the observation that people expect a non-trivial response to their questions (i.e. - more informative than a simple "no"). When the correct direct response is negative, an indirect response can be more informative.

The CO-OP system was developed to provide cooperative responses by addressing any incorrect assumptions the questioner may have made in her/his question. When the direct response to a question would be simply "no" or "none", CO-OP gives a more informative indirect response by correcting the questioner's mistaken assumptions. For example, if question (A) below is posed, the speaker is assuming that projects in oceanography exist. If s/he is wrong, CO-OP gives the corrective indirect response (B) rather than the less cooperative direct response "none".

(A) Which users work on projects in oceanography?
(B) I don't know of any projects in oceanography.

The false assumptions that CO-OP corrects are the existential presuppositions of the question. For example, in question (A) above, the speaker makes the existential presupposition that there are projects in oceanography. Since these presuppositions can be computed from the surface structure of the question, a large store of semantic knowledge for inferencing purposes is not needed. In fact, a lexicon and database schema are the only items which contain domain-specific information. Although this means the CO-OP system is a portable one, it also means that the system does a minimum of semantic analysis.

The modules in the CO-OP system include a parser, a morphological analyzer, the paraphraser, an intermediate phase translator, and a control structure which does the presuppositional analysis when questions result in negative responses. The flow of control is initiated with the morphological analyzer which processes the input question and passes the result to the parser. In this stage, the morphological analyzer strips plural endings, determines the root form for verbs, etc.

The parser uses an Augmented Transition Network (ATN by (WOODS 73)) to parse the question and outputs a syntactic structure of the question in Meta Query Language (MQL). It is at this point that the paraphraser is invoked with the

MQL structure as input. The paraphraser rephrases the question in English and the result is presented to the user. If the user perceives that the system incorrectly interpreted her/his question, s/he can rephrase the question and try again before the database is searched for an answer.

Once the user is satisfied with the system's interpretation, the MQL version of the question is translated into Q, the formal query language used for interrogating the database. (The database for testing the CO-OP system was supplied by the National Center for Atmospheric Research (NCAR). It is in CODASYL format and is compatible with the commercially available database query system SEED (GERRITSEN 78) which was used.) If the database search results in a negative response, the control structure does a presuppositional analysis, checking to see if any of the presuppostions were false. If the database search results in an answer to the question, then the report formatter is called to comprehensibly present the results to the user.

As input to the paraphraser, the MQL structure contains the information available for its use in rephrasing the question. The MQL representation is composed of sets and arcs and encodes the surface structure of the question. The sets denote entities in the database while arcs denote binary relations between those entities. The lexical labels of the arcs and sets are drawn from words in the question. As such, sentences in the system are treated extensionally,

each word in the question pointing to its actual counterpart in the database. Figure 1 shows the MQL interpretation for the sample question (A) above.

Attached to the sets and arcs are properties which provide additional syntactic information about the question. For example, each set or arc has a property CAT which indicates the word's syntactic category. Other information available as properties includes the number of a noun or verb, the topic of the question, the main verb, the tense of a verb, etc. For a full description of MQL see the system documentation on the language and the macros which access it in Appendix A.

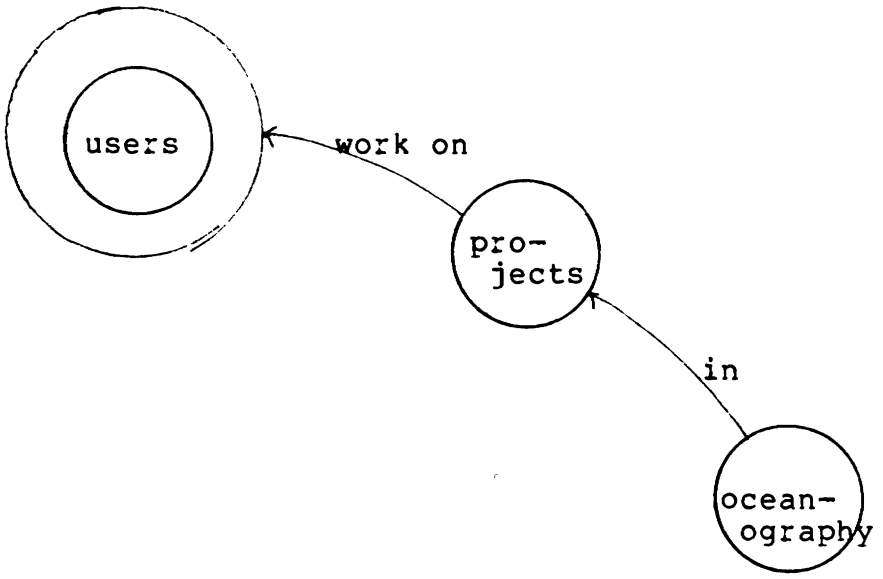



Figure 1

## III. THE CO-OP PARAPHRASER

CO-OP's paraphraser provides the only means of error-checking for the casual user. If the user is familiar with the system, s/he can ask to have the intermediate results printed, in which case the parser's output and the formal database query will be shown. The naive user however, is unlikely to understand these results. It is for this reason that the paraphraser was designed to respond in English.

The use of English to paraphrase queries creates several problems. The first is that natural language is inherently ambiguous. A paraphrase must clarify the system's interpretation of possible ambiguous phrases in the question without introducing additional ambiguity.

One particular type of ambiguity that a paraphraser must address is caused by the linear nature of sentences. A modifying relative clause, for example, frequently cannot be placed directly after the noun phrase it modifies. In such cases, the semantics of the sentence may indicate the correct choice of modified noun phrase, but occasionally, the sentence may be genuinely ambiguous. For example, question (C) below has two interpretations, both equally plausible. The speaker could be referring to books dating from the '60s or to computers dating from the '60s.

(C) Which students read books on computers dating from the '60s?

A second problem in paraphrasing English queries is the possibility of generating the exact question that was originally asked. If a grammar were developed to simply generate English from an underlying representation of the question this possibility could be realized. Instead, a method must be devised which can determine how the phrasing should differ from the original.

The CO-OP paraphraser addresses both the problem of ambiguity and the rephrasing of the question. It makes the system's interpretation of the question explicit by breaking down the clauses of the question and reordering them dependent upon their function in the sentence. Thus, question (C) above will result in either paraphrase (D) or (E), reflecting the interpretation the system has chosen.

(D) Assuming that there are books on computers (those computers date from the '60s), which students read those books?

(E) Assuming that there are books on computers (those books date from the '60s), which students read those books?

The method adopted guarantees that the paraphrase will differ from the original except in cases where no relative clauses or prepositional phrases were used. It was formulated on the basis of a distinction between given and new information and indicates to the user the presuppositions s/he has made in the question (in the "assuming that" clause), while focussing her/his attention on the attributes of the class s/he is interested in.

## IV. LINGUISTIC BACKGROUND

As mentioned earlier, the lexicon and the database are the sole sources of world knowledge for CO-OP. While this design increases CO-OP's portability, it means that little semantic information is available for the paraphraser's use. Contextual information is also limited since no running history or context is maintained for a user session in the current version. The input the paraphraser receives from the parser is basically a syntactic parse tree of the question. Using this information, the paraphraser must reconstruct the question to obtain a phrasing different from the original. The following question must therefore be addressed:

> What reasons are there for choosing one syntactic form of expression over another?

Some linguists maintain that word order is affected by functional roles elements play within the sentence.* Terminology used to describe the types of roles that can

---

* Some other influences on syntactic expression are discussed in (MORGAN and GREEN 73). They suggest that stylistic reasons, in addition to some of the functions discussed here, determine when different syntactic constructions are to be used. They point out, for example, that the passive tense is often used in academic prose to avoid identification of agent and to lend a scientific flavor to the text.

occur varies widely. Some of the distinctons that have been described include given/new, topic/comment, theme/rheme, and presupposition/focus. Definitions of these terms however, are not consistent (for example, see (PRINCE 79) for a discussion of various usages of "given/new").

Nevertheless, one influence on expression does appear to be the interaction of sentence content and the beliefs of the speaker concerning the knowledge of the listener. Some elements in the sentence function in conveying information which the speaker assumes is present in the "consciousness" of the listener (CHAFE 76). This information is said to be contextually dependent, either by virtue of its presence in the preceding discourse or because it is part of the shared world knowledge of the dialog participants. In a question-answer system, shared world knowledge refers to information which the speaker assumes is present in the database. Information functioning in the role just described has been termed "given".

"New" labels all information in the sentence which is presented as not retrievable from context. In the declarative, elements functioning in asserting information that the listener is presumed not to know are called new. In the question, elements functioning in conveying what the speaker wants to know (i.e.- what s/he doesn't know) represent information which the speaker presumes the listener is not already aware of. Firbas identifies additional functions in the question. Of these, (ii) is

used here to augment the interpretation of new information. He says:

> "(i) it indicates the want of knowledge on the part of the inquirer and appeals to the informant to satisfy this want.
>
> (ii) [a] it imparts knowledge to the informant in that it informs him what the inquirer is interested in (what is on his mind) and [b] from what particular angle the intimated want of knowledge is to be satisfied."
>
> (FIRBAS 74; p.31)

Although word order vis-a-vis these and related distinctions has been discussed in light of the declarative sentence, less has been said about the interrogative form. Halliday (HALLIDAY 67) and Krizkova* are among the few to have analyzed the question. Despite the fact that they arrive at different conclusions**, the two follow similar lines of reasoning. Krizkova argues that both the wh-item of the wh-question and the finite verb (e.g. - "do" or "be") of the yes/no question point to the new information to be disclosed in the response. These elements she claims,

---

* Summary by (FIRBAS 74) of the untranslated article "The Interrogative Sentence and Some Problems of the So-called Functional Sentence Perspective (Contextual Organization of the Sentence), Nasa rec 4, 1968.

** It should be noted that Halliday and Krizkova discuss the unknowns in the question in order to define the theme and rheme of a question. Appendix C contains a description of this concept and the analyses made by Halliday and Krizkova.

are the only unknowns to the questioner. Halliday, in discussing the yes/no question, also argues that the finite verb is the only unknown. The polarity of the text is in question and the finite element indicates this.

In this paper the interpretation of the unknown elements in the question as defined by Krizkova and Halliday is followed. The wh-items, in defining the questioner's lack of knowledge, act as new information. Firbas' analysis of the functions in questions is used to further elucidate the role of new information in questions. The remaining elements are given information. They represent information assumed by the questioner to be true of the database domain. This labeling of information within the question will allow the construction of a natural paraphrase, avoiding ambiguity.

## V. FORMULATION

Following the analysis described above, the CO-OP paraphraser breaks down questions into given and new information. More specifically, an input question is divided into three parts, of which (2) and (3) form the new information.

(1) given information
(2) Function ii[a] from Firbas above
(3) Function ii[b] from Firbas above

In terms of the question components, (2) comprises the question with no subclauses as it defines the lack of knowledge for the hearer. Part (3) comprises the direct and indirect modifiers of the interrogative words as they indicate the angle from which the question was asked. They define the attributes of the missing information for the hearer. Part (1) is formed from the remaining clauses.

As an example, consider question (F):

(F) Which division of the computing facility works on projects using oceanography research?

Following the outline above, part (2) of the paraphrase will be the question minus subclauses: "Which division works on projects?". Part (3), the modifiers of the interrogative words, will be "of the computing facility" which modifies "which division". The remaining clause "projects using oceanography research" is considered given information. The three parts can then be assembled into a natural sequence:

(G) Assuming that there are projects using oceanography research, which division works on those projects? Look for a division of the computing facility.

In question (F), information belonging to each of the three categories occurred in the question. If one of these types of information is missing, the question will be presented minus the initial or concluding clauses. Only part (2) of the paraphrase will invariably occur. If more than one clause occurs in a particular category, the question will be further splintered. Additional given information is parenthesized following the "assuming that ..." clause. Example (H) below illustrates the paraphrase for a question containing several clauses of given information and no clauses defining specific attributes of the missing information. Clauses containing information characterized by category (3) will be presented as separate sentences following the stripped-down question. (I) below demonstrates a paraphrase containing more than one clause of this type of information.

(H) Q: Which users work on projects in oceanography that are sponsored by NASA?

P: Assuming that there are projects in oceanography (those projects are sponsored by NASA), which users work on those projects?

(I) Q: Which programmers in superdivision 5000 from the ASD group are advised by Thomas Wirth?

P: Which programmers are advised by Thomas Wirth? Look for programmers in superdivision 5000. The programmers must be from the ASD group.

## VI. IMPLEMENTATION OVERVIEW

The paraphraser's first step in processing is to build a tree structure from the representation it is given. The tree is then divided into three separate trees reflecting the division of given and new information in the question. The design of the tree allows for a simple set of rules which flatten the tree. The final stage of processing in the paraphraser is translation. In the translation phase, labels in the parser's representation are translated into their corresponding words. During this process, necessary transformations of the grammar are performed upon the string.

## A. THE PHRASE STRUCTURE TREE

In its initial processing, the paraphraser transforms the parser's representation into one that is more convenient for generation purposes. The resultant structure is a tree that highlights certain syntactic features of the question. This initial processing gives the paraphraser some independence from the CO-OP system. Were the parser's representation changed or the component moved to a new system, only the initial processing phase need be modified.

The paraphraser's phrase structure tree uses the main verb of the question as the root node of the tree. The subject of the main verb is the root node of the left subtree, the object (if there is one) the root node of the right subtree. In the current system, the use of binary relations in the parser's representation (see (KAPLAN 79) for a description of Meta Query Language) creates the illusion that every verb or preposition has a subject and object. The paraphraser's tree does allow for the representation of other constructions should the incoming language use them.

Each of the subtrees represents other clauses in the question. Both the subject and the object of the main verb will have a subtree for each other clause it participates in. If a noun in one of these clauses also participates in another clause in the sentence, it will have subtrees too.

As an example, consider the question: "Which active users advised by Thomas Wirth work on projects in area 3?".

The phrase structure tree used in the paraphraser is shown in Figure 2. Since "work" is the main verb, it will be the root node of the tree. "users" is root of the left subtree, "projects" of the right. Each noun participates in one other clause and therefore has one subtree. Note that the adjective "active" does not appear as part of the tree structure. Instead, it is closely bound to the noun it modifies and is treated as a property of the noun.

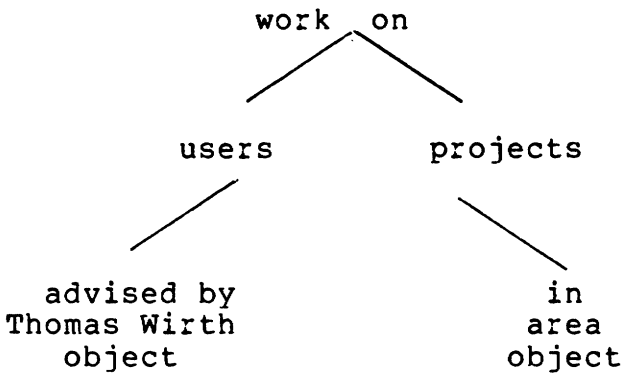
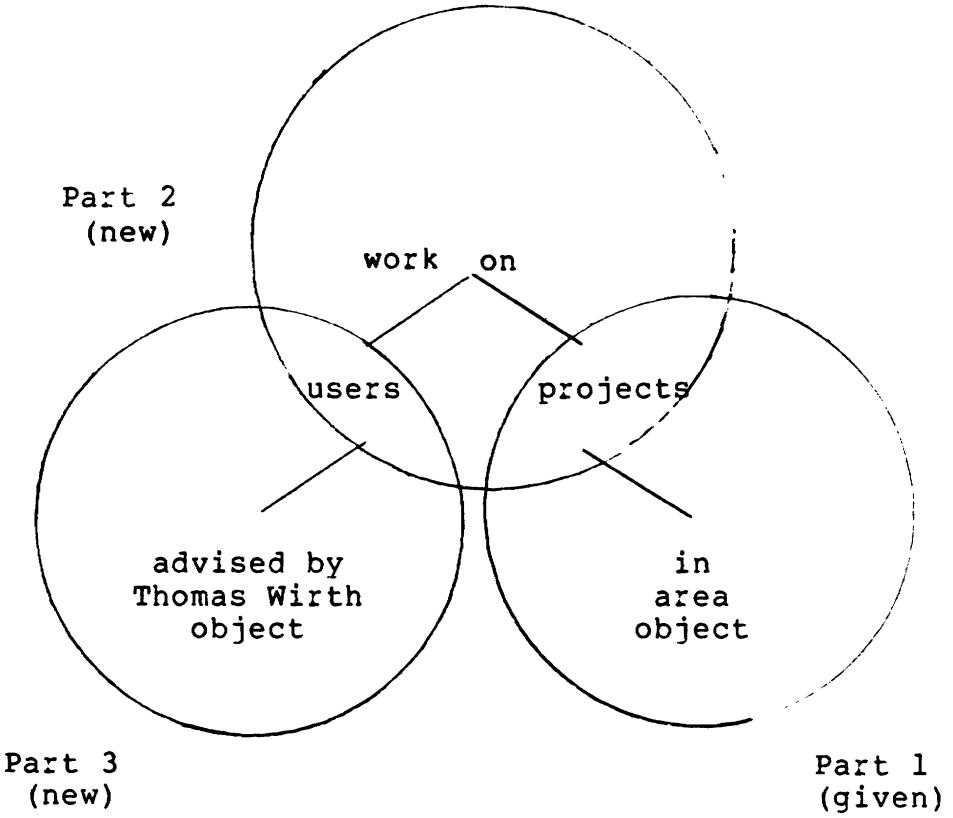



Figure 2

## B. DIVIDING THE TREE

The constructed tree is computationally suited for the three-part paraphrase. The tree is flattened after it has been divided into subtrees containing given information and the two types of new information. The splitting of the tree is accomplished by first extracting the topmost smallest portion of the tree containing the wh-item. At the very least, this will include the root node plus the left and right subtree root nodes. This portion of the tree is the stripped down question. The clauses which define the particular aspect from which the question is asked are found by searching the left and right subtrees for the wh-item or questioned noun. The subtree whose root node is the wh-item contains these clauses. Note that this may be the entire left or right subtree or may only be a subtree of one of these. The remainder of the tree represents given information. Figure 3 illustrates this division for the previous example.




Q: Which active users advised by Thomas Wirth work on projects in area 3?

P: Assuming that there are projects in area 3, which active users work on those projects? Look for users advised by Thomas Wirth.

Figure 3

C. FLATTENING

If the structure of the phrase structure tree is as shown in Figure 4, with A the left subtree and B the right, then the following rules define the flattening process:

```
TREE-> A R B
SUBTREE -> R' A' B'
```

In other words, each of the subtrees will be linearized by doing a pre-order traversal of that subtree. As a node in a subtree has three pieces of information associated with it, one more rule is required to expand a node. A node consists of:

(1) arc-label

(2) set-label

(3) subject/object

where arc-label is the label of the verb or preposition used in the parse tree and set-label the label of a noun phrase. Subject/object indicates whether the sub-node noun phrase functions as subject or object in the clause; it is used by the subject-aux transformation and does not apply to the expansion rule. The following rule expands a node:

```
NODE -> ARC-LABEL SET-LABEL
```

Two transformations are applied during the flattening process. They are wh-fronting and subject-aux inversion. They are further described in the section on

transformations.

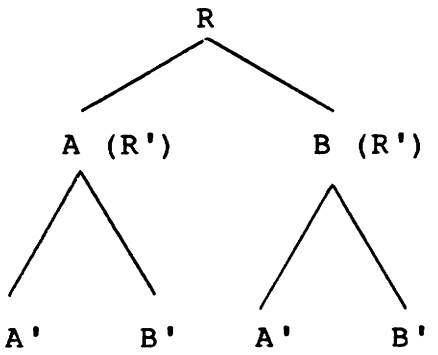
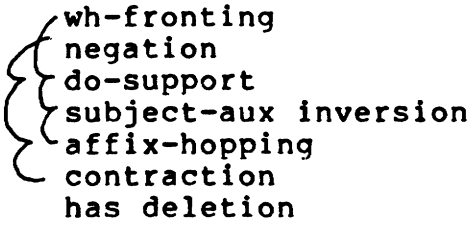



Figure 4

The tree of given information is flattened first. It is part of the left or right subtree of the phrase structure tree and therefore is flattened by a pre-order traversal. It is during the flattening stage that the words "Assuming that there (be) ... " are inserted to introduce the clause of given information. "Be" will agree with the subject of the clause. If there is more than one clause, parentheses are inserted around the additional ones. The tree representing the stripped down question is flattened next. It is followed by the modifiers of the questioned noun. The phrase "Look for" is inserted before the first clause of modifiers.

## D. TRANSFORMATIONS

The grammar used in the paraphraser is a transformational one. In addition to the basic flattening rules described above, the following transformations are used:




The curved lines indicate the ordering restrictions. There are two connected groups of transformations. If wh-fronting applies, then so will do-support, subject-aux inversion, and affix-hopping. The second group of transformations is invoked through the application of negation. It includes do-support, contraction, and affix-hopping. Has-deletion is not affected by the absence or presence of other transformations. A description of the transformation rules follows. The rules used here are based on analyses described by (AKMAJIAN and HENY 75) and analyses described by (CULLICOVER 76).

The rule for wh-fronting is specified as follows, where SD abbreviates structural description and SC, structural change:

```
SD:  X - NP - Y
     1   2    3
SC:  2+1 0    3
condition: 2 dominates wh
```

The first step in the implementation of wh-fronting is a search of the tree for the wh-item. A slightly different approach is used for paraphrasing than is used for generation. The difference occurs because in the original question, the NP to be fronted may be the head noun of some relative clauses or prepositional phrases. When generating, these clauses must be fronted along with the head noun. Since the clauses of the original question are broken down for the paraphrase, it will never be the case when paraphrasing that the NP to be fronted also dominates relative clauses or prepositional phrases. For this reason, when paraphrase mode is used, the applicability of wh-fronting is tested for and is applied in the flattening process of the stripped down question. If it applies, only one word need be moved to the initial position.

When generation is being done, the applicability of wh-fronting is tested for immediately before flattening. If the transformation applies, the tree is split. The subtree of which the wh-item is the root is flattened separately from the remainder of the tree and is attached in fronted position to the string resulting from flattening the other part.

After wh-fronting has been applied, do-support is invoked. In CO-OP, the underlying representation of the question does not contain modals or auxiliary verbs. Thus, fronting the wh-item necessitates supplying an auxiliary. The following rule is used for do-support:

```
SD:  NP - NP - tense - V - X
     1    2          3      4
SC:  1    do+2       3      4
condition: 1 dominates wh
```

Subject-aux inversion is activated immediately afterwards. Again, if wh-fronting applied, subject-aux inversion will apply also. The rule is:

```
SD:  NP - NP - AUX - X
     1    2     3    4
SC:  1    3+2   0    4
condition: 1 dominates wh
```

Affix-hopping follows subject-aux inversion. In the paraphraser it is a combination of what is commonly thought of as affix-hopping and number-agreement. Tense and number are attributes of all verbs in the parser's representation. When an auxiliary is generated, the tense and number are "hopped" from the verb to the auxiliary. Formally:

```
SD:  X - AUX - Y - tense-num-V - Z
     1    2    3         4    5   6
SC:  1   2+4   3         0    5   6
```

Some transformational analyses propose that wh-fronting and subject-aux inversion apply to the relative clause as well as the question. In the CO-OP paraphraser, the head-noun is properly positioned by the flattening process and wh-fronting need not be used. Subject-aux inversion however, may be applicable. In cases where the head noun of the clause is not its subject, subject-aux inversion results in the proper order.

The rule for negation is tested during the translation

phase of execution. It has been formalized as:

```
SD:  X - tense-V - NP - Y
     1       2     3    4
SC:  1      2+no   3    4
condition: 3 marked as negative
```

In the CO-OP representation, an indication of negation is carried on the object of a binary relation (see (KAPLAN 79)). When generating an English representation of the question, it is possible in some cases to express negation as modification of the noun (see question (J) below). In all cases however, negation can be indicated as part of the verb (see version (K) of question (J)). Therefore, when the object is marked as negative, the paraphraser moves the negation to become part of the verbal element.

(J) Which students have no advisors?
(K) Which students don't have advisors?

In English, the negative marker is attached to the auxiliary of the verbal element and therefore, as was the case for questions, an auxiliary must be generated. Do-support is used. The rule used for do-support after negation differs from the one used after wh-fronting. They are presented this way for clarity, but could have been combined into one rule.

```
SD:  X - tense-V-no - Y
     1       2      3
SC:  1     do+2     3
```

Affix-hopping, as described above, hops the tense, number, and negation from the verb to the auxiliary verb. The cycle of transformations invoked through application of negation is completed with the contraction transformation. The statement of the contraction transformation is:

```
SD:  X - do+tense -no - Y
     1     2       3    4
SC:  1    #2+n't#  0    4
```

where # indicates that the result must be treated as a unit for further transformations.

## E. CONJUNCTION AND DISJUNCTION

The use of conjunction and disjunction in questions affects both the design and the implementation of the three-part paraphrase. When conjunction or disjunction appears as part of the new information in the question, no changes need be made in the design of the paraphrase. When conjunction or disjunction appears as part of the given information however, the stripped-down question will refer to the already mentioned conjoined items. Some standard method of referring to given information which contains conjunction (or disjunction) must be adopted. In the CO-OP paraphraser, "some of each" is used to refer to conjoined plurals, "each" to conjoined singulars, and "any of the above" to disjoined entities. For example, (M) below is used to paraphrase question (L):

(L) Which users work on projects advised by Clayton-Paulsen and projects sponsored by NASA?

(M) Assuming that there are projects advised by Clayton-Paulsen and assuming that there are projects sponsored by NASA, which users work on some of each?

Note that since any number of items could potentially be conjoined, expressions which implicitly limit the number, like "both", had to be avoided. Furthermore, conjoined plurals do not necessarily imply that all of the entities are indicated. For example, (L) above does not imply that the speaker is interested only in users who worked on all projects advised by Clayton-Paulsen and all projects

sponsored by NASA. Again, a referring term that does not carry this connotation must be used.

The features of the system affected by the addition of conjunction and disjunction are the MQL representation, the paraphraser's phrase structure tree, and the flattening process. In MQL, the representation of conjunction is implicit except when it occurs around the main verb or one of its objects. When conjunction occurs around the wh-items in the question, the question is split into two, each of which is passed separately to the paraphraser (see question (N) and its paraphrase (O) below). The paraphraser currently does not provide for this type of conjunction since it does not occur in the input. Conjunction that occurs in relative clauses is treated by the parser as additional modification of the head noun and is so encoded in MQL. Question (P) below would be represented in the same way as if question (Q) had been asked.

(N) Which users and advisors are sponsored by NASA?

(O) Which users are sponsored by NASA? Which advisors are sponsored by NASA?

(P) Which users work on projects in oceanography and sponsored by NASA?

(Q) Which users work on projects in oceanography that are sponsored by NASA?

As was the case for conjoined wh-items, this type of conjunction in the user's question is invisible to the paraphraser.

Conjunction around the main verb is visible to the

paraphraser since more than one verb is marked as the main verb in the MQL representation. Although conjunction around the main verb is treated in the same way as other types of conjunction (i.e. - as modification of the subject), the additional modifying arc happens to be the main verb in this case. Conjunction around the objects of the main verb is also visible to the paraphraser. Conjunction around objects of verbs or prepositions results in duplication of the verb or preposition in the MQL representation. When conjunction occurs around the object of the main verb, the main verb is duplicated.

Disjunction is always explicitly represented. A special type of arc called a disjunct arc is used when verbs or nouns are disjoined in the question. Disjunction around nouns in the question results in the duplication in MQL of the relation the noun is part of. Figure 5 depicts the MQL representation of a question using disjunction.

The paraphraser's phrase structure tree currently handles any type of disjunction and the types of conjunction which are visible in the MQL. This is achieved by replicating node labels when conjunction or disjunction occurs in the question. Each node in the tree functions as a syntactic unit. The group of labels resulting from replication is also treated as a syntactic unit. However, each label in the group can have its own set of subtrees, thus allowing for the representation of questions such as (R) below (its representation is shown in Figure 6).

(R) Which users work on projects in oceanography and reports sponsored by NASA?

To distinguish between conjunction and disjunction at a node, the list of labels is nested with alternate levels corresponding to conjunction and disjunction. A node is labeled by a list of conjoined items. Each item in the list is a list of disjoined items. Nesting can occur to any level and ends when an item is a lexical label and not a list. If conjunction or disjunction does not appear at any particular level, only one item will occur in the list. In this manner, any combination of nested conjunction and disjunction can be represented. In fact, the paraphraser's representation allows for a greater range of possibilities than the MQL currently provides for, as it has limited nesting capabilities.

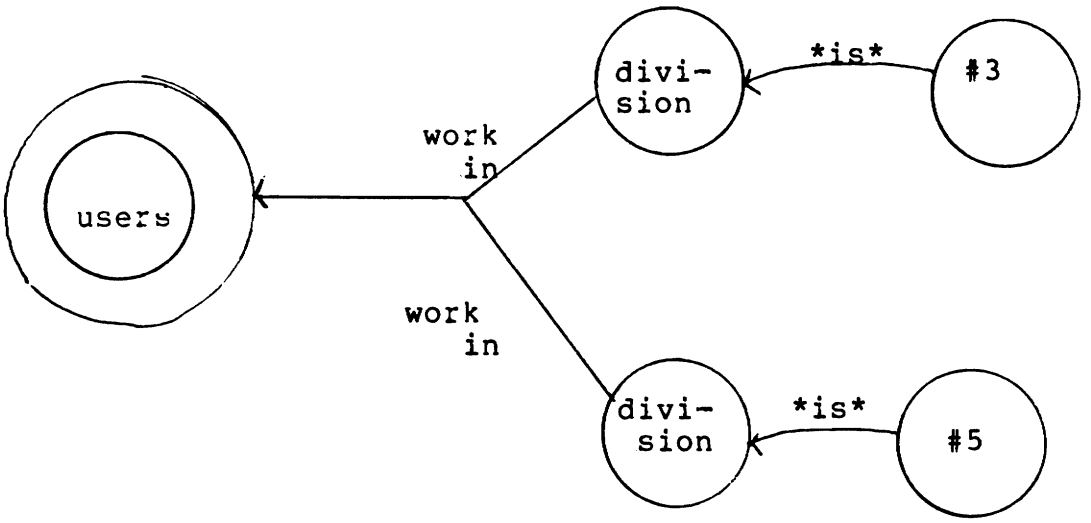
Flattening rules for expansion of nodes also need to be modified to accomodate conjunction and disjunction. Since a node is now a list of labels instead of a single label, ordering rules for expansion of a single node must be used. An item in a list and its subtrees are expanded and then conjoined or disjoined to its neighbors in the list. The procedure is recursive; it is applied to successive levels until a the bottom level is reached. A lexical label and its subtrees are expanded by the rules presented in Section VI. C above. The rule for expansion of a list of labels does not apply when the node has already been mentioned in the question. In such cases, the appropriate referring

phrase is used.

(1) (item-1 item-2 ... item-n)-> item-1 and item-2 and .... item-n

(2) If item-n = (item-n1 item-n2 .... item-nm)
item-n->item-n1 or item-n2 or .... item-nm
Else item-n is a lexical label

(3) if item-nm is not a lexical label, repeat




Which users work in division 3 or 5?

Figure 5

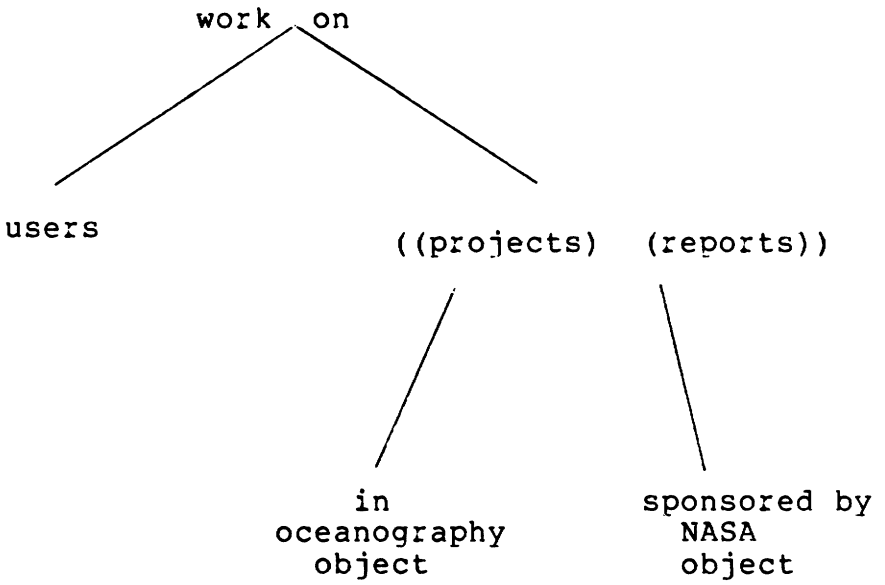



Figure 6

## F. NUMERICAL MODIFICATION

The CO-OP system incorporates a very limited treatment of quantification, that in fact, looks more like numerical modification of nouns. Any explicit quantifiers in the question (like "all", "every", "3 or more", etc.) are interpreted as modifiers of the nouns they precede. They are represented in MQL as properties of sets (see (KAPLAN 79) for details on the interpretation of quantified questions in parsing). Each property is a numerical range; two numbers are used, the first indicating the lower bound, the second indicating the upper. "u" is used to indicate all, or everything in the universe. For example, the pair (3 5) means "from 3 to 5", (0 0), "from 0 to 0" or "none" (indicating negation), (1 u) "from 1 to all", or "some", etc.

The paraphraser treats numerical modification in the same way it treats adjectives. Properties are translated from numerical representation to English during the third stage of processing, translation. As a set is translated from unique label to lexical entry, any numerical modifiers of the set are introduced into the string. Table 1 below shows how the numerical pairs are translated into words.

When numerically modified expressions occur as part of the given information in a question, a slightly different strategy is used. In such cases, the modification is not presented as part of the existential presupposition representing given information. When the noun appears for

the second time, the numerical modifier is used. Paraphrases (T) and (V) of (S) and (U) respectively, demonstrate this for two cases:

(S) Which students work on 5 or more projects advised by NASA?

(T) Assuming that there are projects advised by NASA, which students work on 5 or more of those projects?

(U) Which users work on every project advised by NASA?

(V) Assuming that there is at least one project advised by NASA, which user works on every such project?

| Number | Plural | Singular |
|---|---|---|
| (u u) | all | every |
| (1 u-1) | no all | no every |
| (0 0) | no | |
| (n m) | n to m | |
| (n n) | exactly n | |
| (n>1 u) | n or more | |

Table 1

G. Translation

In the translation phase, the final cosmetic changes are made to produce the paraphrase. The input to this module is a string of Lisp Gensyms, each Gensym being a unique label for a word in the original question. The string also includes the words which were introduced for the three-part paraphrase during the flattening process (e.g. - "Assuming that ... ", "Look for ...", etc.). The syntactic structure of the string is essentially that of the final paraphrase; in some cases, transformatons are performed upon the string during this stage.

The major bulk of work done during this stage is the translation of labels into their English counterparts. For nouns and adjectives, translation is a simple look-up procedure since the lexical entry for these items is stored as a property of the Gensym. The lexical property of a Verb Gensym contains the root form for the verb and some syntactic information. The tense, the number, and whether the verb has a regular conjugation are stored. The paraphraser calls the morphological routines with this information to conjugate the verb.

The nouns that were used in the question are translated first. A list of set-labels representing the nouns is maintained as part of the MQL. The input string is searched for each set-label in the list and the set-label in the string is replaced by the appropriate word. At this point, any adjectives that modify the noun are introduced into the

question. As mentioned earlier, during the tree-building phase, adjectives are stored as properties of nouns and do not appear as part of the tree structure. This is done in order to avoid splitting the adjective modification into clauses of given and new information. Instead, adjectives are closely bound to the nouns they modify. When translating, each set-label is checked for pre- or post-modification. Quantification on a set is also translated into words (see Section VI F for details).

The relations in a question are translated after the nouns have been translated. The relations include verbs and prepositions. A search of the input string is made for each arc-label and its occurrence in the string is replaced by the preposition or conjugated verb. When an arc-label is translated into a verb, the applicability of the negation transformation is tested. In order for the negation transformation to apply, "no" must appear directly after the verb (see Section VI. D). This is only possible if the quantification on the verb's object has already been translated and indicates negation. It is for this reason that the verbs are only translated once the nouns have been translated. Following translation of the verbs, the paraphrase is complete.

## VII. GENERATION

The paraphrase component has been given a dual function. It can generate an English version of the parser's representation as well as paraphrase in the three-part form. This function uses the same procedures and grammar as the three-part paraphraser, but the tree is not split into three separate trees before being flattened. The generation function could be used to produce the paraphrase, but if it were, there is no guarantee that the question would differ from the user's.

In CO-OP, generation is used to produce alternative suggestions and corrective responses. A corrective response is used to correct the user's false presuppositions. When an existential presupposition encoded in the question is incorrect, the portion of MQL representing the particular presupposition is passed to the paraphraser which generates the corrective response. For example, (X) below is a corrective response that could be generated by the paraphraser if (W) were asked:

(W) Which programmers in division 3 work on projects in oceanography?

(X) I don't know of any projects in oceanography.

Alternative suggestions are also used by the CO-OP system when the direct response to the user's question is negative. If an incorrect presupposition is removed from a question, the resulting question may no longer have a

negative response. In such cases, the system suggests the wider class question to the user as a possible interest. Alternative suggestions are only presented after a corrective response has been made. Thus, a sequence like (W),(X) above might be followed by the alternative suggestion (Y):

(Y) But you might be interested in programmers in division 3 that work on any projects.

For corrective responses, the paraphraser receives the portion of the MQL representation of the user's question which encodes the incorrect presupposition. For alternative suggestions, the paraphraser also receives a portion of the original MQL. In this case, it is the portion representing the user's question minus the incorrect presupposition. For both types of response, the paraphraser generates a question from the MQL representation it is given. The wh-item is then stripped from the front of the question and a phrase is attached to the front, converting the question to a statement. "I don't know of any ... " is used for corrective responses. Slight modifications are made, depending on the wh-item used. Table 2 shows the correspondence between particular wh-items and the phrases used. "But you might be interested in ... " is used for alternative suggestions. The adjective "any" is used before the noun which was restricted by the incorrect presupposition in the original question. Thus, in (Y) above, "any" modifies "projects" which, in the original

question, was restricted by "in oceanography".

The flattening process for generation differs from that used for paraphrasing. The tree is not divided into subtrees representing given and new information and therefore, the tree is flattened as a whole. The order of traversal is the same; the left and right subtrees are flattened by pre-order traversals, the total tree by an inorder traversal. The rule for expansion of sub-nodes however, is not identical. It introduces the word "that" into the question in order that each subtree be expanded as a relative clause, and not as a separate sentence. The rule is:

SUB-NODE->that ARC-LABEL SET-LABEL

The transformational grammar also applies to the generation process, with the one difference being the point at which the applicability of wh-fronting is tested for. Other than these changes and the flattening process, the generation process is the same as the paraphrase process. The generation function is general enough that it may eventually be used for other types of responses in cases when something other than a direct response is needed.

| WH-ITEM | PHRASE | EXAMPLE (question and corrective response) |
|---|---|---|
| which | I don't know of any | Which users work on projects?<br><br>I don't know of any users that work on projects. |
| who | I don't know of anyone | Who works on projects in oceanography?<br><br>I don't know of anyone that works on projects in oceanography. |

Table 2

| WH-ITEM | PHRASE | EXAMPLE (question and corrective response) |
|---|---|---|
| what | I don't know of anything | What is sponsored by NASA?<br><br>I don't know of anything that is sponsored by NASA. |
| where | I don't know of anyplace | Where do users have accounts?<br><br>I don't know of anyplace that users have accounts. |
| How many | I don't know of any | How many users have accounts?<br><br>I don't know of any users that have accounts. |

Table 2 (continued)

## VIII. FUTURE RESEARCH

The CO-OP paraphraser is lacking in the area of semantics. Although a good deal of attention has been given to the form of the question and reasons for using a different form than was used in the original, the words used in the paraphrase are the same words that occurred in the user's question (with the exception of words that are added in the paraphrase to introduce given or new information). The next step in paraphrasing in the database system is the development of a method to determine why and how the particular words in the paraphrase should differ from the user's.

A logical schema of the database is currently being developed (see (MAYS 79)) which will provide some semantic information for the system. The schema will be independent of the implementaion of any particular database and will contain knowledge about the structure of the data. This information could be used by the paraphraser in choosing words that reflect the structure and content of the database to replace words used in the original question.

The current version of the system does do some lexical disambiguation. Nouns in the user's question which do not occur in the lexicon are assigned to a specific database category. The word's syntactic function in the question and any semantic constraints placed upon it are used to determine its category (see (KAPLAN 79) for details). For example, in question (Z) below, the name Thomas Wirth does

not appear in the lexicon. (AA) is output to the user, indicating the category that has been assigned. The phrase is output before the paraphrase and is done by the control structure.

(Z) Which programmers advised by Thomas Wirth are in division 3?

(AA) I am assuming that Thomas Wirth is an advisor name.

Another change which would provide the user with additional semantic information would be to generate part of the paraphrase from the formal database query. This would provide two specific types of information for the user. The first of these has to do with the verbs used in the question and the relations in the database. When a verb which does not occur as a relation in the database is used in the question, a composite of relations from the database may be used to form a "new" relation corresponding to the verb. For example, "checks that bounced" might correspond to "checks whose amount is less than the account balance" in some fictitious banking database. In such cases, it may be helpful for the user to see how the verb was interpreted in terms of the database concepts.

A second type of information that could be generated from the formal database query concerns the method by which the database is to be searched. In some cases, the order in which sets are retrieved from the database and then restricted by modifying clauses makes a difference in the time taken to find the answer. Although such information is

not necessary for the user to understand the system's interpretation of her/his question, some users may be interested in the efficiency of the database search.

Additions to the paraphraser could also be made in the area of inferencing. The paraphraser could be used to indicate the system's interpretation of the user's intentions if the system were to address the question of why a particular question was asked. This type of paraphrase would contain more information than just the substantial content of the user's question. Some of the user's intentions or motives may be deduced on the basis of her/his question alone. For example, when a question is asked that requires a list for an answer, the questioner may not really be interested in all items on the list. Recognition of such motives is useful when the answer to a question becomes unmanageably large. The paraphraser could be used to ask the user about underlying motives which would restrict the list.

If a model were maintained of the user during her/his session with the system, more information would be available to aid the paraphraser in determining the user's intentions. It is often the case that a person asks a series of questions on one topic. S/he may have to ask several questions to get the information needed to ask a particular question. In such cases, the paraphraser may be able to deduce what the user is aiming at and include it as part of the paraphrase. Moreover, a question asked at the end of a

series of questions may take on a slightly different meaning if viewed in light of previous questions rather than taken in isolation. A running history of a user session would provide the paraphraser with the necessary information to generate these nuances of meaning.

## IX. RELATED RESEARCH

At the present time, two other paraphrasers that I am aware of exist for database question-answer systems. One was developed by David Waltz et. al. (WALTZ 78) for the PLANES system, the other by Ted Codd et. al. for the Rendezvous Version 1 System (CODD 78). The PLANES system generates the paraphrase from the formal database query using templates. The process involves three specific actions. English words are substituted for any abbreviations or code names which appeared in the database query. An appropriate paraphrase template is selected for use and the slots in the template are then filled with words and phrases from the query. The process is not generation per se. It involves the formation of templates which are suitable for the particular database and for the types of questions which can be asked.

The Rendezvous System also uses templates, although it is slightly more sophisticated than WALTZ's. There are three parts to generation and two types of templates are used. A header template which corresponds to the type of query is chosen first. There are three types of queries in the system (FIND, EXIST, and COUNT), of which FIND occurs most frequently. The header for FIND is PRINT THE ... EVERY ..., where the dots must be filled in. The second part of the paraphrase is the target list and it occurs only in PRINT type queries. The third part of the paraphrase is called the body. It is formed by extracting patterns from

tables that are associated with particular items in the database.

The goals of the Rendezvous generation component are important ones. The generated English must unambiguous, easy to understand, discriminating, and not misleading (CODD 78). Instead of developing a general solution to achieve these goals however, the research seems to be concentrated on particular examples which don't meet these criteria. This results in part from the use of patterns which are essentially fragments of English to be inserted in the sentence. The patterns must be constructed beforehand for a particular database and great care must be taken to choose phrases that can be easily patched together with a variety of other phrases. Such a solution necessitates looking at particular examples, instead of the general framework.

Goldman (GOLDMAN 75) has also developed a paraphraser, although it is not part of a question-answer system. The system, MARGIE, generates English from a conceptual dependency network and operates in either inference or paraphrase mode. In paraphrase mode, MARGIE outputs all possible ways it knows of expressing a particular concept. Unlike the CO-OP paraphraser, MARGIE is a semantic paraphraser; it uses different idioms and phrases to express the same idea.

Other work has been done in generation by Simmons and Slocum (SIMMONS and SLOCUM 72), Heidorn (HEIDORN 75), and

McDonald (MCDONALD 78). Simmmons and Slocum have developed a system to generate English from semantic networks using a transformational grammar. The formalism they use is similar to an ATN (WOODS 73). Heidorn uses an augmented phrase structure grammar with an interpreter for the rules. His system can be used for both generation and analysis. McDonald has examined the more specific problem of the use of pronouns versus naming through the use of nouns and proper nouns. He has developed a system for generation that incorporates the constraints he has observed.

## X. CONCLUSIONS

The paraphraser described here is a syntactic one. While this work has examined the reasons for different forms of expression, additions must be made in the area of semantics. The substitution of synonyms, phrases, or idioms for portions or all of the question requires an examination of the effect of context on word meaning and of the intentions of the speaker on word or phrase choice. The lack of a rich semantic base and contextual information dictated the syntactic approach used here, but the paraphraser can be extended once a wider range of information becomes available.

The CO-OP paraphraser has been designed to be domain-independent and thus a change of the database requires no changes in the paraphraser. Paraphrasers which use the template form however, do require such changes. This is because the templates, or patterns, which constitute the type of question that can be asked, are necessarily dependent on the domain. For different databases, a different set of templates must be used.

The CO-OP paraphraser also differs from other systems in that it generates the question using a transformational grammar of questions. It addresses two specific problems involved in generating paraphrases:

1. ambiguity in determining which noun phrases a relative clause modifies

2. the production of a question that differs from the user's

These goals have been achieved for questions using relative clauses through the application of a theory of given and new information to the generation process.

## XI. APPENDIX A

### MACROS and MQL Representation

MQL representation:

((.... sets .....)(... relations ....))

MQL is a list, the CAR of which is a list of Gensyms that identify the sets in a graph. The CADR is a list of lists. Each list in the list represents a relation. Its format is as follows:

(<gensym> (<gensym> <gensym> .....))

The CAR of the list is a Gensym which uniquely identifies a relation in the graph. The CADR of the list is list of Gensyms which identify the sets the relation is involved in. The first item in the list will be the high order set (or the subject) of the relation. The second item in the list will be the low order set (or the object) of the relation. If the arc is a disjunct arc, there will be more than one low order set.

All other information is located on property lists of the Gensyms. Each node in a graph has the following properties associated with it:

CAT The lexical category of the node. Will be either N (noun), PN (proper noun), ADJ (adjective), WH (wh-word).

QUANT The quantification on the node. This will be a list (e.g. (u u) for universal quantification).

NUM The number of the node. Will be either PLUR or SING.

LEX The lexical word associated with the Gensym (e.g. - USER).

TOPIC This will be true if the node is the topic of the question.

Each arc has the properties associated with it:

LEX All lexical information associated with an arc will be on this property list. Its form is a list of lists. There will be one list for each low order set associated with the arc. Each list will have the following form:

(<vrb> <prep> (<tnse> <num>) <reg> <main>)

<vrb> will be the lexical verb associated with the arc. (e.g. WORK). If there is none, it will be NIL. <PREP> will be the lexical preposition associated with the arc (e.g. - IN). If there is none, it will be NIL. Note that both of these slots can be non-NIL or one of them can be NIL, but they both can not be NIL in the same list. The next item is a list of tense and number of the

verb. If <vrb> is nil, these will be also. <tnse> can be either PRES, PAST, PRESP (present participle), or PASTP (past participle). <num> can be SING or PLUR. <reg> indicates whether the verb is conjugated regularly or not. If it is <reg> will be T. If not,<reg> will be a list of the proper conjugations taken from the lexicon. <Main> will be T if this arc is the main verb of the question.

## MACROS

ARC:HIORDSET
    Arguments:
        1. The arc for which the high order set is needed.
        2. The list of relations and their associated sets. (the CADR of the MQL graph).
    Returns:
        The Gensym identifying the high order set or subject of the arc.

ARC:LOWORDSET
    Arguments
        1. the arc for which the low order set is needed.
        2. the list of relations and their associated sets.
    Returns:
        the first low order set of the arc.

ARC:LEX
    Arguments:
        1. the Gensym identifying the arc to be translated.
    Returns:
        A list containing the lexical translation

of the arc. This will be either a preposition a verb, or a verb and preposition. The verb will be properly conjugated.

ARC:VRB
    Arguments:
        1. One list of the property list of an arc. i.e. - The lexical information corresponding to one arc, whether it be part of a disjunct arc or a simple arc.
    Returns:
        The unconjugated verb from that list.

ARC:PREP
    Arguments:
        1. One lexical list of an arc.
    Returns:
        A list of the preposition or NIL if there is no prepositon.

ARC:TNSE
    Arguments:
        1. one lexical list of an arc
    Returns:
        the tense of the verb from that list.

ARC:REG
    Arguments:
        1. one lexical list of an arc.
    Returns:
        T if the verb is regularly conjugated. Otherwise, it returns the list of conjugations for the irregular verb.

ARC:CAT
    Arguments:
        1. The Gensym identifying the arc.
    Returns:
        The lexical category of an arc. Note that this will be either PREP or VERB.

NODE:LEX
    Arguments:
        1. The Gensym identifying the node.
    Returns:
        The lexical translation of a node.

NODE:QUANT
    Arguments:
        1. The Gensym identifying the node.
    Returns:
        the quantification on a node. This will be in the form of a list (e.g. - (u u) for universal quantification).

NODE:CAT
    Arguments:
        1. The Gensym identifying the node.
    Returns:
        The category of a node. This will be either N, PN, ADJ, or WH.

NODE:NUM
    Arguments:
        1. The Gensym identifying the node.
    Returns:
        The number of the node. This will be either PLUR or SING.

NODE:TOPIC
    Arguments:
        1. The set of nodes in the graph.
    Returns:
        The topic of the set of nodes.

ARCS:
    Arguments:
        1. The MQL graph
    Returns:
        The set of Gensyms identifying the arcs in the graph.

NODES:
    Arguments:
        1. The MQL graph
    Returns:
        The set of Gensyms identifying the nodes in the graph. This is the car of the MQL.

## XII. APPENDIX B

Sample Questions and their Paraphrases

Q: Which programmers from the ASD group are in superdivision 5000?

P: Which programmers are in superdivision #5000? Look for programmers from ASD group.

Q: Who advises projects in area 36?

P: Assuming that there are projects in area #36, who advises those projects?

Q: Which advisors have accounts for projects in oceanography?

P: Assuming that there are accounts for projects (those projects are in oceanography.), which advisors have those accounts?

Q: Which users that work on projects that NASA sponsors are in division 3?

P: Which users are in division #3? look for users that NASA sponsors. The users must work on projects.

Q: Which users work on projects in area 3 that are in division 200?

P: Assuming that there are projects in area #3, which users work on those projects? Look for users that are in

division #200.

Q: What are the groups?

P: Display the groups.

Q: Which users work on every project in area 55?

P: Assuming that there is at least one project in area #55, which users work on every such project?

Q: Who sponsors more than 3 projects?

P: Who sponsors 4 or more projects?

Q: Which programmers work in division 3 or 4?

P: Which programmers work in division #3 or division #4?

Q: Which users work on all projects in oceanography?

P: Assuming that there are projects in oceanography, which users work on all of those projects?

## XIII. APPENDIX C

### A Note on Theme and Rheme

The concept of theme and rheme has been discussed in relation to its affect on the word order of a sentence in more detail than have other distinctions. Linguists of the Prague School postulate that the sentence is divided into elements providing common ground for the conversants (theme) and elements which function in conveying the information to be imparted (rheme). The definition of elements as thematic is governed by two constraints. In sentences containing elements which are contextually dependent (i.e. - conveying information known or determined from context), the contextually dependent elements always function as theme. In sentences lacking known or given information, theme is defined as those elements having the lowest degree of communicative dynamism* (CD), a vague concept. Rheme, on the other hand, is characterized by a high degree of CD. Since elements conveying new information carry a higher degree of CD than those that don't, rheme is close to the concept of new information.

The Prague School contends that in English there is a tendancy for theme to appear as subject of the sentence and for rheme to appear towards the end of the sentence. The

---

* (FIRBAS 74) defines the degree of communicative dynamism of an element as "the extent to which the element contributes towards the development of the communication."

reverse order is possible, though less likely. According to these observations, if context indicates that the transitive subject of the sentence conveys the new information, while the theme occurs as object, then the sentence would be passivized to re-establish the order of theme first, rheme last.

An analysis of theme and rheme and its function in the question has been made by both Halliday and Krizkova. Although they agree about the unknowns for the questioner (see Section IV), they disagree about which elements function as theme and which funcion as rheme. Halliday defines the theme of a question as a demand for information. The wh-item of a question is interpreted as theme because of its position in the question and because it indicates the speaker's want of knowledge and desire to fill it. He says:

> "In a non-polar interrogative for example, the wh-item is the theme by virtue of its being the point of departure for the message; it is precisely what is being talked about."
> (HALLIDAY 67; p. 212)

Krizkova criticizes this analysis, instead interpreting the interrogative words of the question as rhematic. For Krizkova, the rhematic elements are the unknowns in the question. The remaining elements function as theme. Krizkova's definition of theme and rheme is similar to the given/new distinction since she only considers what is known

or unknown in the question.

Halliday, on the other hand, is closer to the concept of topic/comment articulation in his definition of theme and rheme. Describing the difference between theme and given information, he defines theme as that which the speaker is talking about now, as opposed to given, that which the speaker was talking about. Furthermore, Halliday always ascribes the term theme to the element occurring first in the sentence. He labels the remainder of the sentence as rheme. For him, whether elements function as theme or rheme is determined by the order of the sentence. It is this difference in interpretation of the meaning of the terms theme and rheme that accounts for the conflict in his and Krizkova's analysis.

## XIV. REFERENCES


1. (AKMAJIAN and HENY 75). Akmajian, A. and Heny, F., An Introduction to the Principles of Transformational Syntax, MIT Press, 1975.

2. (CHAFE 76). Chafe, W. L., "Givenness, Contrastiveness, Definiteness, Subjects, Topics, and Points of View", Subject and Topic (ed. C. N. Li), Academic Press, 1976.

3. (CODD 78). Codd, E. F., et al., Rendezvous Version 1: An Experimental English-language Query Formulation System for Casual Users of Relational Data Bases, IBM Research Report RJ2144(29407), IBM Research Laboratory, San Jose, Ca., 1978.

4. (CULLICOVER 76). Cullicover, P. W., Syntax, Academic Press, N. Y., 1976.

5. (DANES 74). Danes, F. (ed.), Papers on Functional Sentence Perspective, Academia, Prague, 1974

6. (FIRBAS 66). Firbas, Jan, "On Defining the Theme in Functional Sentence Analysis", Travaux Linguistiques de Prague 1, Univ. of Alabama Press, 1966.

7. (FIRBAS 74). Firbas,Jan, "Some Aspects of the Czechoslovak Approach to Problems of Functional Sentence Perspective", Papers on Functional Sentence Perspective, Academia, Prague, 1974.

8. (GERRITSEN 75). Gerritsen, R., SEED Reference Manual, Version C00-B04 draft, International Data Base Systems, Inc., Philadelphia, Pa., 19104, 1978.

9. (GOLDMAN 75). Goldman, N., "Conceptual Generation", Conceptual Information Processing (R. C. Schank), North-Holland Publishing Co., Amsterdam, 1975.

10. (GRICE 75). Grice, H. P., "Logic and Conversation", in Syntax and Semantics: Speech Acts, Vol. 3, (P. Cole and J. L. Morgan, Ed.), Academic Press, N. Y., 1975.

11. (HALLIDAY 67). Halliday, M.A.K., "Notes on Transitivity and Theme in English", Journal of Linguistics 3, 1967.

12. (HIEDORN 75). Heidorn, G., "Augmented Phrase Structure Grammar", TINLAP-1 Proceedings, June 1975.

13. (JOSHI 79). Joshi, A. K., "Centered Logic: the Role of Entity Centered Sentence Representation in Natural Language Inferencing", to appear in IJCAI Proceedings 79.

14. (KAPLAN 79). Kaplan, S. J., "Cooperative Responses from a Portable Natural Language Data Base Query System", Ph.D. Dissertation, Univ. of Pennsylvania, Philadelphia, Pa., 1979.

15. (MAYS 79). Mays, E., forthcoming Masters Thesis, Dept. of Computer and Information Science, U. of Pennsylvania, Philadelphia, Pa., 1979.

16. (MCDONALD 78). McDonald, D. D., "Subsequent Reference: Syntactic and Rhetorical Constraints", TINLAP-2 Proceedings, 1978.

17. (MORGAN and GREEN 77). Morgan,J.L. and Green, G.M.: "Pragmatics and Reading Comprehension", University of Illinois, 1977.

18. (PRINCE 79). Prince, E., "On the Given/New Distinction", to appear in CLS 15, 1979.

19. (Sgall, Hajicova, and Benesova 73). Sgall,P., Hajicova, E., and Benesova, E., Topic, Focus and Generative Semantics, Scriptor Verlag Gmbh, Kronberg Taunus, 1973.

20. (SIMMONS and SLOCUM 72). Simmons, R. and Slocum, J., "Generating English Discourse from Semantic Networks", Univ. of Texas at Austin, CACM, Vol. 5, #10, October 1972.

21. (WALTZ 78). Waltz, D.L., "An English Language Question Answering System for a Large Relational Database", CACM, Vol. 21 #7, July 1978.

22. (WOODS 73). Woods, W. A., "An Experimental Parsing System for Transition Network Grammars", in Natural Language Processign, Courant Computer Science Symposium #8, R. Rustin (Ed.), Algorithmics Press, Inc., N.Y., 1973.