AD-A038 913 ILLINOIS UNIV AT URBANA-CHAMPAIGN COORDINATED SCIENCE LAB F/G 5/7
GETTING THE GIST: A COMPUTATIONAL THEORY OF SENTENCE UNDERSTAND--ETC(U)
DEC 76 S NAKAJIMA N00014-75-C-0612
UNCLASSIFIED T-37 NL

1 OF 1
AD
A038913

END
DATE
FILMED
5-77

REPORT T-37 DECEMBER, 1976

CSL COORDINATED SCIENCE LABORATORY

GETTING THE GIST: A COMPUTATIONAL THEORY OF SENTENCE UNDERSTANDING

SHIGEKI NAKAJIMA

UNIVERSITY OF ILLINOIS - URBANA, ILLINOIS

6

GETTING THE GIST:
A COMPUTATIONAL THEORY
OF SENTENCE UNDERSTANDING

by

10 Shigeki Nakajima

11 Dec 76

12 39p.

This work was supported in part by the Office of Naval Research under Contract N00014-75-C-0612 15 NEW

14 T-37

097 700

ABSTRACT

This paper shows the computation of English sentences in different task domains--the robot world, a children's story, and the front-end of information retrieval. The GIST (Grammar Instructed STructure) analyzes these sentences, using a grammar which provides a partial interpretation of sentences, and some guidelines towards a more complete understanding.

TABLE OF CONTENTS

I. INTRODUCTION

This paper presents a theory of understanding English sentences. By using well-known example sentences we shall discuss a new approach for computational understanding. First of all we shall see the process of parsing a sentence in isolation, which demonstrates a notion of grammar quite different from that of some grammars currently being used. We shall then apply GIST (Grammar Instructed STructure) to the parsed sentences in a paragraph comprehension task, which provides partial understanding and some guidelines towards a fuller understanding. This paper is conceptual in nature, and the functions and variables are tentative assignments.

I would like to make some preliminary remarks regarding the GIST, the essential properties of which have already been described (Nakajima 1975b). The term Grammar Instructed STructure emphasizes the contribution of grammar to the extraction of information from a sentence. Some characteristics of the GIST are the following:

1) It consists of sentence components such as subject, object, and complement(s).

2) By indicating the roles of objects in a description, it shows clearly the relationships among objects in a setting being described.

3) It separates the problem of determining what the present or subsequent state is from the problem of establishing a procedure or method of getting to that goal.

4) It provides a substantial tool for constructing a context which deals with the frame intention-action-goal.

II. A COMMAND IN THE BLOCK WORLD

In this chapter we consider the analysis of a sentence in isolation, a command, but before doing so I would like to say a few words about the preprocessing of the system. SYNAPS, described in the appendix of my M.S. thesis is a program which converses with a person at a teletype in a very simple subset of English, using a simple context-free grammar. It can only take input from the speaker in a rigid list format. I have rewritten some functions to read input in a looser format and to convert it into a list format that the rest of the system can digest; I will use this processing for the new system described in this paper. It can read input character by character and build a list, and read a dot as terminating the input sentence. It can analyze the morphemic structure of the words and check all words in the input sentence to make sure they are in the current vocabulary.

To demonstrate some of the basic elements of the analyzer in action, we will use as an example the following command:

Pick up a block.

In the preprocessing the system scans the input sentence for double words and idioms and converts them into single symbols. For example, PICK UP is changed to PICK-UP, and the features or the requests of PICK-UP1 are brought as follows:

```
PICK-UP1:

(REPLACE GIST
   (QUOTE ((ACTOR (#SUBJ)<=> ACT (#ACT))
         <= (GOAL (#OBJ) OBJECT ((HAND) PART (≡ACTOR)))
         <= (GOAL (≡ACTOR) SOURCE (#DONOR) OBJECT (#OBJ ))
          TIME (NIL)))
   NIL)
```

This assumes that the GIST for the sentence is the linear equivalent of the following graph:

```
                 (#SUBJ)<=> PICK-UP1
                        ⇑
((HAND) PART (#SUBJ))<-> (#OBJ)
                        ⇖
                 (#OBJ)<——>(#SUBJ)
                          └< (#DONOR)
```

That is, the picking-up is the transfer of some object by using the actor's hand. In this particular case, the SOURCE might be ON(TABLE1); this may be interpreted as

Pick up (with your hand) a block (on the table).

Notice that the information contained in the phrases "with your hand" and "on the table" must be added to the sentence; in Winograd's system, this information is automatically provided by theorems or "world knowledge," which actually present no alternative choices in most cases since the block world is so limited. We can say that Winograd's system understands not the sentence "Pick up a block," but its own interpretation of the sentence, after the context-dependent information about the block world has been added. In my system, the GIST of the sentence provides an interpretation which does not require the input of context-dependent information; it is therefore an incomplete and nonspecific interpretation, but it accounts for the fact that even without a tableful of blocks before him, a person can get some meaning from the sentence "Pick up a block."

In this particular example of a command given in isolation, since the program sees no information to the contrary, it might bring in the following:

```
(GADD PGIST
    (QUOTE ((ACTOR (SPEAKER)<=> ACT (#ACT))
           <= (GOAL (LISTENER) SOURCE (SPEAKER) OBJECT (MES
                      (#GIST (#SUBJ = LISTENER))))
            TIME (NIL)))
    NIL)
```

This is an interpretation of an imperative sentence. An equivalent statement might be

Terry says to Shrdlu to pick up a block.

provided that SPEAKER is Terry and LISTENER is Shrdlu.

In addition to this information, another feature of PICK-UP1 is provided, a caution that an expression of the purpose of picking-up might follow, since according to the ACT variable diagram in Appendix A (and in Nakajima 1975a) the word <u>pick-up</u> is in the group G_6. Of course, it is assumed that the (#ACT) in the above will be one of the higher level variables among the groups G_1 to G_5, perhaps a term like <u>say</u> or <u>tell</u>.

In a discussion of a complete system, we would have to consider at this point the possibility of an action taken by the system; since Shrdlu interprets the input sentence as a command given by Terry, it must be ready to do something, either a simulation in the case of Shrdlu, or a real robot action. I am not going into this matter here since it is a different task from mine, the computation of an English sentence.

When the word a is read, the program starts to build a noun phrase list and store the current list of requests in a certain variable. The phrase is completed when it encounters the final word BLOCK, which has the features

(BLOCK1 ROLE (OBJECT GOAL SOURCE))

(BLOCK1 WORDTYPE (NOUN))

This states that block cannot be an ACTOR and must have one of the ROLEs object, goal, or source. The requests attached to a are activated and the program builds the noun phrase "(BLOCK1 RED (A))". Subsequent actions choose this noun phrase to be the #OBJ and the final result is

```
((ACTOR (TERRY1) <=> ACT (#ACT))
 <= (GOAL (SHRDLU1) SOURCE (TERRY1)
    OBJECT (MES
          ((ACTOR (SHRDLU1)<=> ACT (PICK-UP1))
          <= (GOAL (BLOCK1) OBJECT ((HAND) PART (SHRDLU1)))
          <= (GOAL (SHRDLU1) SOURCE (#DONOR) OBJECT (BLOCK1))
          TIME (NIL)))
    TIME (TIMOO))
```

where (#ACT) and (#DONOR) are not yet assigned. Since no particular request has been made, (#ACT) remains as it is, but (#DONOR) should be filled out by searching the context since this information was not supplied in the sentence. Because the input does not say anything about how to pick up the block, the program must figure this out itself; a good example of this is in Winograd's system; I will not go into this aspect further. However, I would like to compare the GISTs of some input sentences.

According to Winograd's thesis, Shrdlu must interpret "Pick up a red block" as "Pick up with your HAND a block on the TABLE." A comparison of the GISTs of this sentence with those of some similar sentences points up some of the essential characteristics of this type of representation.

First, the distinction between statement and command is not given by labels, but by the arrangements of slots. Compare the following:

1) Pick up a block.

```
                TIMOO
                  ↓
        #SUBJ    <=>   #ACT

MES( * ) <-> #RECIP
              └< #SUBJ

 #RECIP  <=>  PICK-UP1
          ⇑
 BLOCK1  <->  #RECIP
          └<  #DONOR
```

2) Shrdlu picked up a block.

```
          TIMO1
            ↓
  SHRDLU1 <=> PICK-UP1
           ⇑
   BLOCK1 <-> SHRDLU1
           └< ON (TABLE1)
```

This distinction is, of course, maintained in reported speech as well:

3) "Pick up a block," said Terry.

```
           TIMO1
             ↓
  TERRY1    <=>   TELL2
             ⇑
  MES(*)    <->   #RECIP
             └<   TERRY1

#RECIP  <=>   PICK-UP1
         ⇑
BLOCK1  <->   #RECIP
         └<   #DONOR
```

3)' Terry told Shrdlu to pick up a block.

```
            TIMO1
              ↓
  TERRY1     <=>   TELL2
              ⇑
  MES(*)     <->   SHRDLU1
              └<   TERRY1

SHRDLU1  <=>   PICK-UP1
          ⇑
 BLOCK1  <->   SHRDLU1
          └<   #DONOR
```

4) Terry told Shrdlu that Shrdlu had picked up a block.

```
                          TIMO1
                            ↓
                  TERRY1  <=>  TELL2
                            ⇑
                  MES(*)  <->  SHRDLU1
                            └< TERRY1

      TIMO2
        ↓
SHRDLU1 <=> PICK-UP1
         ⇑
 BLOCK1 <-> SHRDLU1
         └< ON (TABLE1)
```

The difference between (1) and (3)-(3)' is in the assignment of values for LISTENER and SPEAKER; in (3)', both are specified; in (1), neither is specified; even when the information is unspecified, the GIST is able to provide an interpretation of the sentence.

In this respect, note that <u>tell</u> has two senses, TELL1 "let somebody know" sense, and TELL2 "instruct." In (4) we have the "let somebody know" sense, and in (3)' we have the "instruct" sense. This is predictable since we know <u>tell</u> is a variable in the group G_4 as TELL2, and can take another variable in G_6 or in some other group. As TELL1, it can take a statement with <u>that</u>. If <u>to</u> is read, then it is assumed that it is a command with the "instruct" sense; and if <u>that</u> is read with a following statement, it is assumed to be a statement with the "let somebody know" sense. The system can translate both GISTs accordingly.

The GIST also distinguishes between a desire and an act (or the report of an act). Compare the following with (1) and with (3):

5) Terry wanted Shrdlu to pick up a block.

```
SHRDLU1 <=> PICK-UP1
         ⇑
 BLOCK1 <-> SHRDLU1
         └< #DONOR
         ⇑
 TERRY1 <-> PLEASED
```

A desire, of course, can be reported, as in (6) below:

6) Terry said that he wanted Shrdlu to pick up a block.

```
                    TIMO1
                      ↓
             TERRY1 <=> SAY1
                      ⇑
             MES(*) <-> #RECIP
                      └< TERRY1
SHRDLU1 <=> PICK-UP1
         ⇑
 BLOCK1 <-> SHRDLU1
         └< #DONOR
         ⇑
 TERRY1 <-> PLEASED
```

Finally, consider the following example:

7) Terry said that he was glad that Shrdlu had picked up a block.

```
TIMOO:  T
TIMO1:  BEFORE TIMOO
TIMO2:  BEFORE TIMO1
TIMO3:  BEFORE TIMO2
```

```
                 TIM01
                   ↓
          TERRY1 <=>  SAY1
                   ⇑
          MES(*) <->  #RECIP
                   └< TERRY1
                                             TIM02
                                               ↓
                                     #SUBJ <=> #ACT
                                               ⇑
                                    MES(*) <-> TERRY1
                                               └< #SUBJ
                                               ⇑
                                    TERRY1 <-> PLEASED
     TIM03
       ↓
SHRDLU1 <=> PICK-UP1
        ⇑
 BLOCK1 <-> SHRDLU1
        └< #DONOR
```

This example points up a number of interesting features of the GIST. First, the GIST can distinguish the related concepts of <u>want X</u> and <u>be glad that X</u>. Although both representations indicate a (potential or actual) change of state of Terry to PLEASED, with <u>be glad that X</u> it is a message which causes Terry to be pleased; this reflects the fact that being glad about X presupposes knowledge of X; this is not true of <u>want</u>. Note also the sequences of tenses.

III. A PARAGRAPH

Next we will consider the following paragraph taken from Charniak.

1) Fred was going to the store.
2) Today was Jack's birthday.
3) And Fred was going to get a present.

In this section, instead of showing the process of the analysis of isolated sentences, I shall use the result of each sentence analysis to gain an understanding of the paragraph, and to determine the relationships among the sentences.

The first sentence is analyzed as (1)' below:

```
          TIMO1
            ↓
   FRED1  <=>  GO1
            ⇑
   FRED1  <->  STORE1
            └<  #DONOR
```

MANNER (PLANNED ACT)
(FRED1 WORDTYPE (NOUN NAME))
(FRED1 ROLE (ACTOR OBJECT GOAL SOURCE))

with another GIST for STORE1, which indicates that money is exchanged for objects at a store:

```
            STORE1
              ↓
     #SUBJ  <=>  #ACT
              ⇑
   MONEY1   <->  (OWNER #RECIP)
              └<  #SUBJ
              ⇑
      #OBJ  <->  #SUBJ
              └<  (OWNER #DONOR)
```

(STORE1 WORDTYPE (NOUN))
(STORE1 ROLE (LOC GOAL SOURCE OBJECT))

The following analysis of the second sentence:

```
             TIMO1
               ↓
(2)'  #SUBJ  <=>  #ACT
               ⇑
       #OBJ  <->  JACK1
               └<  #SUBJ
```

(JACK1 WORDTYPE (NOUN NAME))
(JACK1 ROLE (ACTOR OBJECT GOAL SOURCE))

offers one interpretation of birthday, the one relevant to the story, that Jack receives things on his birthday, Information about the "meaning" of birthday as an anniversary of one's birth is not particularly relevant here. This may not be inconsistent with the child's notion of birthday. #DATE has the components #MONTH, #DAY, and #YEAR; BIRTHDAY1 fills out the #MONTH and #DAY:

(DATE (TODAY (BIRTHDAY1 #YEAR)))

The final sentence is shown as (3)':

```
                      TIMO1
                        ↓
                FRED1  <=>  GET1
                        ⇑
     (PRESENT #OBJ)  <->  FRED1
                        └<  #DONOR
```

MANNER (PLANNED ACT)

with <u>present</u> represented as something which, as a result of a #SUBJ's action, is transferred to a #RECIP.

```
(PRESENT  ROLE (OBJECT GOAL SOURCE))
(PRESENT  WORDTYPE (PRED NOUN))
```

```
                    #SUBJ  <=>  #ACT
                            ⇑
       (PRESENT #OBJ)  <->  #RECIP
                            └<  #SUBJ
```

More specifically, we can represent this sentence as follows:

```
                      TIMO1
                        ↓
               FRED1  <=>  GET1
                        ⇑
     (PRESENT #OBJ)  <->  FRED1
                        └<  #DONOR
                        ⇑
               FRED1  <=>  #ACT
                        ⇑
     (PRESENT #OBJ)  <->  #RECIP
                        └<  FRED1
```

In the sentence, this is expressed not as a completed fact but as a planned act: Fred <u>was going to</u> get a present. In general, the structure <u>be going to</u> indicates an intended or expected event; this is represented by the MANNER (PLANNED ACT). The specific tense of <u>be</u> merely places the PLANNED ACT into some time reference, indicated in this case by the time TIMO1. Thus, "getting a present" is here a timeless concept, with the MANNER (PLANNED ACT).

Since the last part of (3)'' matches with the GIST (2)', we can write

```
                     TIMO1
                       ↓
              FRED1  <=>  #ACT
                       ⇑
     (PRESENT #OBJ)  <->  JACK1
                       └< FRED1
```

Combining (1)' with the representation of STORE1, we can rewrite (1)' as (1)'':

(1)''

```
               TIMO1
                 ↓
        FRED1  <=>  GO1
                 ⇑
        FRED1  <->  STORE1
                 └< #DONOR
                 ⇑
        FRED1  <=>  #ACT
                 ⇑
      MONEY1   <->  (OWNER #RECIP)
                 └< FRED1
                 ⇑
         #OBJ  <->  FRED1
                 └< (OWNER #DONOR)
```

Therefore, by matching we can combine (1)'' and the latter portion of (3)'' to give the following guess for a unified interpretation of the paragraph:

```
TIMOO:   T
TIME:    BEFORE TIMOO
(DATE (TODAY (BIRTHDAY #YEAR)))

                   FRED1  <=>  GO1
                            ⇑
                   FRED1  <->  STORE1
                            └< #DONOR
                            ⇑
                   FRED1  <=>  #ACT
                            ⇑
                  MONEY1  <->  (OWNER #RECIP)
                            └< FRED1
                            ⇑
         (PRESENT #OBJ)   <->  FRED1
                            └< (OWNER #DONOR)
                            ⇑
                   FRED1  <=>  #ACT
                            ⇑
         (PRESENT #OBJ)   <->  JACK1
                            └< FRED1
```

This is a "concept interpretation" of the paragraph, rather than a representation of the sentences themselves. It adds information not mentioned explicitly and organizes the events according to their relationships with each other. In this case, the order of the sentences is not particularly relevant to either the chronological order of events or the conceptual organization of the paragraph. For example, had the sentences been given in the order 1-2-3 or 3-2-1, the representation would be the same. The GIST, then, can provide an organization of concepts which is much more comprehensive than that provided by the tenses alone.

Let's take another example from Charniak:

1) Janet needed some money.
2) She got her piggy bank.
3) and started to shake it.
4) Finally some money came out.

From (1) we would have

```
          #SUBJ  <=>  #ACT
                  ⇑
         MONEY1  <->  JANET1
                  └<  #SUBJ
                 i⇑
(JANET1 WORDTYPE (NOUN NAME))
(JANET1 ROLE (ACTOR OBJECT GOAL SOURCE))
```

where #SUBJ may or may not be JANET1. The lowest arrow indicates an intentional consequence; that is, a provision is made for another event which is expected to be attached. In this case, the consequence implied is a transfer of money to Janet. This captures the notion that closely associated with a need is its gratification, whether or not it is actually accomplished.

The next sentence brings (2)':

```
                               TIMO1
                                 ↓
                         JANET1 <=> GET1
                                 ⇑
     (PIGGYBANK1 REF (HER)) <-> JANET1
                             └< #DONOR

(PIGGYBANK1 WORDTYPE (NOUN))
(PIGGYBANK1 ROLE (OBJECT GOAL SOURCE))
```

where we have two kinds of representations for PIGGYBANK1, reflecting our knowledge that a piggybank is something one puts money into, and something one gets money out of:

```
      #SUBJ  <=>  #ACT1                  #SUBJ  <=>  #ACT1
PIGGYBANK1   <->  #SUBJ            PIGGYBANK1   <->  #SUBJ
      #SUBJ  <=>  #ACT2                  #SUBJ  <=>  #ACT2
    MONEY1   <->  IN(PIGGYBANK1)       MONEY1   <->  #SUBJ
              └<  #SUBJ                          └<  IN(PIGGYBANK1)
```

The possessive form <u>her</u> in "her piggybank," however, causes a change in the GIST for PIGGYBANK1 so that JANET1 is chosen as the #SUBJ:

```
      JANET1  <=>  #ACT                  JANET1  <=>  #ACT
PIGGYBANK1    <->  JANET1          PIGGYBANK1    <->  JANET1
    MONEY1    <=>  IN(PIGGYBANK1)      MONEY1    <=>  JANET1
               └<  JANET1                         └<  IN(PIGGYBANK1)
```

For the third sentence we have the following representation:

```
               TIMO1
                 ↓
      JANET1    <=>  #ACT
PIGGYBANK1      <->  SHAKEN
```

Here the GIST does not specify or describe the action "shake" itself. This is outside the domain of the knowledge imbedded in the GIST; it gives the roles of participants and relationships among events, but it does not specify actual physical movement or its consequences.

To represent the fourth sentence, we have (4)':

```
        TIMO1
          ↓
  #SUBJ  <=>  #ACT
 MONEY1  <->  #RECIP
          └<  IN(#DONOR)
```

```
(MONEY1 WORDTYPE (NOUN))
(MONEY1 ROLE (OBJECT GOAL SOURCE))
```

From the term <u>come out</u> we get the information that the money was transferred from inside something, and can write IN(#DONOR). By partially matching (4)' with (1)' and with the second GIST for PIGGYBANK1, one might conclude the following:

TIM01
↓
JANET1 <=> #ACT
⇑
PIGGYBANK1 <-> JANET1
└< #DONOR
⇑
JANET1 <=> #ACT
⇑
MONEY1 <-> JANET1
└< IN(PIGGYBANK1)

Since the intentional consequence of a transfer of money to Janet is satisfied by the actual consequence, the mark on ⇑ has disappeared. This example illustrates both the limitations and the usefulness of the GIST. The GIST can specify the roles played by the objects described in a sentence, but not physical actions themselves. These might be better described as knowledge about sensory-motor planning, which performs actions in the real world.

Critics might argue that a cookie jar or a desk drawer can receive the same GIST as PIGGYBANK1 if it is used to store money until it is needed. This is true, and it illustrates precisely the most outstanding characteristic of GIST, the ability to provide analogical representations. For if a cookie jar or a desk drawer is used to keep money, and if this fact is understood by the interlocutors, then for the purpose of this story they fulfill the same function as PIGGYBANK1 and can be represented identically. There may be other representations which record their physical characteristics, or their other functions, but since these are not particularly relevant here they are not called up. Because the GIST is partial and somewhat vague, it is able to capture similarities and express analogies between things which are quite dissimilar in other ways. The GIST can, for example, express the analogy of function between Janet's piggybank and the First National Bank, by representing both as something which one puts money into and takes money out of; this may, in fact, encompass most of a child's conception of the function of both. The GIST can also represent other features of First National, for example, as a building which people enter and leave, and some of its other functions, as something which lends money, pays interest on money, invests money, and transfers money. These representations would, of course, be more complex than those in Janet's story, but so are the processes involved. The lexical entries, too, give the role possibilities of a lexical item without pseudosyntactic semantic features such as $\pm$Animate or $\pm$Human.

IV. A QUERY IN THE BLOCK WORLD

In this chapter we will use the following example, taken from Winograd's block world, to examine some of the problems involved in treating queries:

> How many blocks are supported by the cube which I told you to pick up?

A query is somewhat different from the cases discussed previously, and to analyze one we must observe how relationships among objects are determined and represented.

In his block world, Winograd treats the sentences below as equivalent by paraphrasing (SUPPORT 1 2) as (ON 2 1):

1) The block is on the cube.
2)a The cube supports the block.
  b The block is supported by the cube.

The GIST, however, provides different interpretations of (1) and (2), and an examination of the steps which lead to the divergent interpretations reveals the basic analogical approach of the GIST. Relevant to this approach is the fact that the two expressions are not necessarily synonymous, and the observation that the subject of <u>support</u> is analogous to an ACTOR.

If the following GIST for SUPPORT1

```
((SACTOR (#SUBJ) <=> ACT (#ACT)
  <= (GOAL (#RECIP) SOURCE (#SUBJ) OBJECT (#OBJ))
  <= (GOAL (SUPPORTED) OBJECT (#RECIP))
TIME (NIL))
```

is to accept CUBE1

```
(CUBE1 WORDTYPE (NOUN))
(CUBE1 ROLE (OBJECT GOAL SOURCE))
```

as its subject (i.e., SACTOR), CUBE1 must be interpreted as a pseudoactor or SACTOR. Using the following representation for BLOCK1:

```
(BLOCK1 WORDTYPE (NOUN))
(BLOCK1 ROLE (OBJECT GOAL SOURCE))
```

the GIST below is assigned:

```
((SACTOR (CUBE1) <=> (SUPPORT1))
  <=(GOAL (BLOCK1) SOURCE (CUBE1) OBJECT (#OBJ))
  <=(GOAL (SUPPORTED) OBJECT (BLOCK1))
TIME (NIL))
```

Of course, CUBE1 cannot play the role of ACTOR in the normal sense of the latter; however, in contrast to being treated solely as an OBJECT, it can be seen as analogous to an ACTOR, and to express this we use the designation SACTOR.

In this particular case, with CUBE1 as SACTOR, the GIST might be partially rewritten as

```
((SACTOR (CUBE1) <=> ACT (SUPPORT1))
  <= (GOAL (BLOCK1) SOURCE (CUBE1) OBJECT (#OBJ))
  <= (GOAL (ON (CUBE1)) OBJECT (BLOCK1))
TIME (NIL))
```

by applying the following request;

```
(REPLACE PGIST <=  (GOAL (SUPPORTED))
 <= (GOAL (ON (CUBE1))))
```

In this way, we can capture the relationship between <u>support</u> and <u>be on</u>; if the sentence "X supports Y" is read, then one can say "Y is on X." However, if the sentence "Y is on X" is read, it cannot be rewritten as "X supports y" internally--i.e., at the GIST level--since this is not always true. In the sentences with <u>support</u>, X is designated as SACTOR; in the sentence with <u>on</u> it is treated as an OBJECT. These assignments depend not on features such as ±Animate belonging to the lexical items themselves, but to the role the speaker attributes to an object.

For example, if you touch the top of your head with the palm of your hand, without letting its full weight rest on your head, you can say, "My hand is on my head," but not, "My head supports my hand." That is, in such cases we do not attribute the role of ACTOR to <u>head</u>, and cannot use <u>head</u> as the subject of <u>support</u>. The sentence above does not violate any "cooccurrence restrictions" proposed by grammarians; it is not ungrammatical, it is simply inaccurate in this case.

There are other meanings of <u>support</u>, one of which is illustrated in the following sentence:

John supports Mary by sending money.

The lexical representations for <u>John</u> and <u>Mary</u>:

```
(JOHN1 WORDTYPE (NOUN NAME))
(JOHN1 ROLE (ACTOR OBJECT GOAL SOURCE))

(MARY1 WORDTYPE (NOUN NAME))
(MARY1 ROLE (ACTOR OBJECT GOAL SOURCE))
```

match the GIST for SUPPORT2:

```
((ACTOR (#SUBJ <=> ACT  (#ACT)
  <= (GOAL (#RECIP) SOURCE (#SUBJ) OBJECT (#OBJ))
  <= (ACTOR (#RECIP <=> ACT (#ACT))
TIME (NIL))
```

to provide the following representation of the sentence:

John supports Mary by sending money.

```
          TIM00
            ↓
   JOHN1 <=> #ACT
            ⇑
    #OBJ <-> MARY1
            |
            L< JOHN1
            ⇑
   MARY1 <=> #ACT
```

Notice that since the GIST for SUPPORT2 requires that both the #SUBJ and the #RECIP be ACTORs, it will not match the representations of CUBE1 and BLOCK1, and will be rejected for the sentence "The cube supports the block." Thus we see that the GIST is capable of providing some general representation of the relationships among objects which does not depend upon the particular situation being described, and which is applicable even outside the block world.

In Appendix B are some remarks on the interpretation of a scene.

Let us return to the sentence we are analyzing:

How many blocks are supported by the cube which I wanted you to pick up?

Since in the description in Appendix B there is no mention of CUBE the program cannot find directly which object is specified. It can, however, get some information directly from the sentence by carrying out the verbal specification found in the relative clause. If it can match the representation of "the cube which I wanted you to pick up" (the lower portion of the GIST below) with a representation found previously, then it can actually identify some object as a candidate for CUBE1, using only linguistic understanding.

```
(BLOCK1 REF (HOW MANY)) <-> ON (CUBE1 REF (THE))

                                       TIMO1
                                         ↓
                                #SUBJ  <=>  #ACT
                                         ⇑
                                MES (*) <-> TERRY1
                                         ⇑
                               TERRY1  <->  PLEASED
          NIL
           ↓
SHRDLU1   <=>   PICK-UP1
           ⇑
  CUBE1   <->   #RECIP
```

Humans frequently supplement the knowledge provided directly from a verbal description with sensory-motor knowledge, as well as with their general knowledge about objects in the real world, spatial relationships among objects, and so on. Thus, a person might understand a query, and then bring into play his knowledge that a "cube" is a kind of block, and then examine the real scene and perhaps manipulate objects in it. The GIST is generally concerned only with the understanding which is provided by the linguistic description itself. Other systems, because they include information from these three different kinds of knowledge in their representations, appear to be more powerful, but actually they fail to represent all these components adequately, and fail to take advantage of all the information which can be gotten from linguistic descriptions. This is precisely the advantage of the GIST. Although it cannot perform all the operations or interpret all the components of human understanding, it can provide an interpretation of a statement, command, or query without direct knowledge of, or experience with, the situation itself.

Note that there is a difference between understanding what a cube is in general, and being able to identify which particular cube in the scene is being requested. Of course the description of the scene in Appendix B does not tell what a block or pyramid is in general. To provide this kind of information, we need a model of the setting being described, although we do not need direct sensory data about the objects at this level of analysis. The model must provide information about what a block or pyramid looks like in general.

For example, a block can be characterized by the following attributes: color, size, height, thickness, material, shape, and so on. These attributes give us a framework within which we can understand attribute values like red or big. In addition, there are representations which reflect direct experience with an object; the representation below, which might result from picking up an object, putting it somewhere, throwing it, or hitting it with one's hand, indicates that the object is manipulable.

```
((ACTOR (#SUBJ) <=> ACT (#ACT))
  <= (GOAL (BLOCK1) OBJECT ((HAND) PART (ACTOR)))
  <= (GOAL (#RECIP) SOURCE (#DONOR) OBJECT (BLOCK1))
TIME (NIL))
```

These representations are essentially nonlinguistic, and are in addition to the GIST.

Of course, we have already made use of linguistic information like the following:

```
(BLOCK1 ROLE (OBJECT GOAL SOURCE))
(BLOCK1 WORDTYPE (NOUN))
```

to interpret descriptions like the one in Appendix B.

In a real setting, where there is a need to model the setting more precisely in order to perform an action, the system must have a model of what the objects look like in order to recognize and identify them. It must use sensory data resulting from the analysis of TV data.

What I have been suggesting is that with the GIST one can make a general picture of a setting, though an incomplete one, which permits an interpretation and discussion of the setting based on a preliminary model of it. In this way, we can somehow separate verbal ability from sensory-motor ability. The GIST, hopefully, can be a good component for this sort of understanding system.

V. A QUERY IN THE NAVY DATA BASE

In the previous chapter, we used the GIST to provide an interpretation of a query in the block world. After showing how the GIST is able to provide an understanding of relationships like support, we saw that with this particular query, the system would be able to answer appropriately without using a sensory-motor program or TV sensory data, just by consulting memory. In this chapter, I would like to take a look at another query, dealing with a different setting. This sort of query often appears in the domain of information retrieval, which is quite different from the robot world. Although I do not dicuss the process of searching the data base which the system uses to find appropriate information items to answer a query, I do point out some ways in which the GIST is helpful in interpreting queries of this sort.

The Navy data base consists of data which includes a record of previous maintenance for a particular airplane, specifying the kind of maintenance which has been done, the date, and so on. Since the input sentence can almost always be expected to be some kind of question asking about these records, the type of input sentence may be a direct question, or a command with an embedded question, or perhaps a statement with an embedded question. For example, the following sentences are typical:

How many Phantoms required maintenance in April?
Tell me which Phantoms required maintenance in April.
I want to know how many Phantoms required maintenance in April.

In such cases, the system does not need to have a characterization of an airplane, or of what an airplane does, or of what one does with an airplane. Rather it must know what kinds of jobs are needed to maintain an airplane, what kinds of airplanes will be asked about, and when an event concerning the maintenance of an airplane occurs. Since the system does not have to manipulate an airplane as a physical object, or perform a job to maintain one, we do not need to provide sensory-motor information about airplanes.

Assuming that the system will have as a part of linguistic knowledge the following information about an airplane:

```
(PHANTOM1 WORDTYPE (NOUN))
(PHANTOM1 ROLE (OBJECT GOAL SOURCE))

(MAINTENANCE1 WORDTYPE (NOUN PRED))
(MAINTENANCE1 ROLE (OBJECT GOAL SOURCE))

(APRIL1 WORDTYPE (NOUN PRED))
(APRIL1 ROLE (DATE))
```

let's consider the sentence below:

How many Phantoms required maintenance in April?

REQUIRED brings the following assignment:

```
(CHOOSE TIME (BEFORE (NEWTIME)) NIL)
```

The second assignment would be

```
(REPLACE GIST
      (QUOTE ((ACTOR (#HSUBJ) <=> ACT (REQUIRE1))
              <= (GOAL (#SUBJ) SOURCE (#HSUBJ) OBJECT (#OBJ))
              TIME (NIL)))
NIL)
```

Since the first word of the sentence is <u>how</u>, the system tries to construct the noun phrase

```
(PHANTOMA REF (HOW))
```

When it hits the fourth word <u>require</u>, it gives the following representation:

```
((ACTOR (#HSUBJ <=> ACT (REQUEST1)
  <= (GOAL (PHANTOMA REF (HOW)) SOURCE (#HSUBJ) OBJECT (#OBJ)
  TIME (TIMO1)))
```

When the fifth word <u>maintenance</u> is read, it will be assigned as

```
((ACTOR (#SUBJ <=> ACT (REQUEST1)
 <= (GOAL (PHANTOMA REF (HOW))
    SOURCE (#HSUBJ) OBJECT (MAINTENANCE (#OBJ)))
  TIME (TIMO1))
```

The sixth word <u>in</u> brings the following request:

```
        :
IN1:
    (PROG
       (COND ((EQUAL NTWDBL (QUOTE DATE))
                 (GADD PGIST
                     (QUOTE (DATE (NTWD)))))
              (( . . .
```

The result of this analysis is

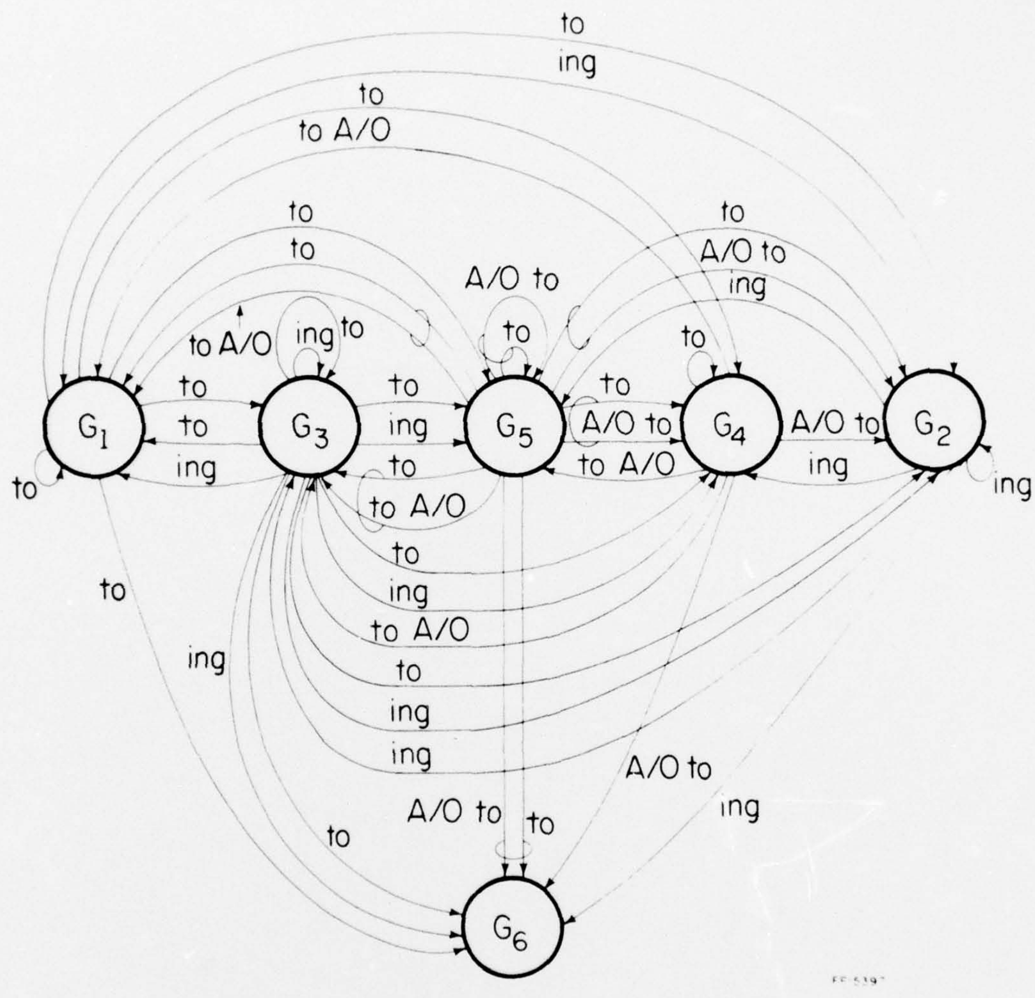
```
((ACTOR (#HSUBJ) <=> ACT (REQUEST1))
  <= (GOAL (PHANTOMA REF (HOW))
      SOURCE (#HSUBJ) OBJECT (MAINTENANCE (#OBJ)))
 TIME (TIM01) DATE (APRIL (#DATE)))
```

In this representation, #HSUBJ may be the particular shop which did the maintenance. The interpretation of the noun phrase (PHANTOMA REF (HOW)) will request a count of the number of the noun (PHANTOMA). Information pertaining to the kind of maintenance that has been performed will be entered as (#OBJ); furthermore, (#DATE) is waiting to be filled out as a specific date, for example, as April 1975.

As we have seen, the GIST can provide an adequate interpretation of a sentence. Still remaining is the task of constructing a searching program over the data base.

APPENDIX A

ACT Variable Network



APPENDIX A

ACT Variable Network

G_1

*agree	deserve	*pretend	
aim	fail	proceed	
be-able	hesitate	*promise	
*decide	*hope	refuse	
decline	*learn	*threaten	
*demand	offer		etc.

G_2

*acknowledge	**do-not-mind	*admit	
**accept	miss	**admit-to	
**address	**object-to	avoid	
**be-aware-of	postpone	complete	
**be-conscious-of	put-off	evade	
**check	*recall	give-up	
*consider	resist	practice	
contemplate	risk	quit	
control (G_6)	shun	finish	
decide-on	succeed-in		
*deny	*suggest	*advise (G_4)	
*detest	**think-about	*imagine (G_4)	
*dislike	**talk-about	*understand	
escape	**tolerate		
escape-from		arrange	
facilitate	keep	arrange-for	
*favor	keep-from	*choose	
*fear	stop	**hate	
fight	stop-from	hate-for	
fight-against	keep-A/O-from	**like	
flee-from	stop-A/O-from	**love	
cannot-help	prevent-A/O-from	*plan	
include	suspect-A/O-of	plan-on	
		*prefer	
		**cannot-stand	etc.

G_3

attempt	neglect	begin	
dread	*remember	continue	
*forget	try	start	etc.

---

*These ACT variables may be followed by that + clause.
**These ACT variables may be followed by the fact that + clause.

G_4

*advise (G_2)
allow
*forbid
order
permit
*teach
*tell
*remind

cause
get (=induce)
require

*feel
*hear
listen-to
look-at
*see
*watch

compel
direct
drive (=compel)
encourage
force
guide
incite
induce
influence
lead
move (=persuade)
*persuade
pull (=influence)
push (=force)
urge

*believe
*calculate
*determine
*estimate
*find
*judge
*imagine (G_2)
*know
*observe
*report
*show
*understand (G_2)

etc.

G_5

apply
*ask
*beg
desire
*expect
help (to opt.)
long (for)
*mean (for)
need
prepare
prepare (for)
say (for)
wait (for)
want

*arrange (for)
*choose
*hate
hate (for)
*like
like (for)
*love
love (for)
*plan (for)
*prefer
prefer (for)
*cannot-stand
cannot-stand (for)

etc.

G_6

accompany
add
amuse
aggravate
arrive
bake

befriend
bite
break
bring
build
build-up

burn
cannibalize
carry
catch
clean
combine

G_6 . . . cont.

comfort
compare
confuse
connect
contain
contribute
control (G_2)
cook
cough
cover
crash
crush
cry
cut
deliver
depart
describe
die
disband
discharge
disgust
display
disappear
dissolve
distribute
disturb
divide
draw
drink
drop
drop-out
eat
embed
employ
enter
exchange
exist
extend
fall
feel
find
fix
flatter
flee
flow
flower
fluctuate
fly
fold
freeze

give
go
grab
grow
hand
heat
hit
hunt
hurt
increase
insult
jump
kick
kill
kiss
knock
land
laugh
leave
lend
lie
list
live
malfunction
manufacture
mark
marry
mature
meet
merge
move (G_4)
pass
pick
pick-up
picture
place
please
*plot
point
polish
position
print
purchase
put
raise
reach
read
receive
repair
replace

ride
rise
roar
roll
rub
run
say (G_5)
sell
send
set
shake
shiver
ship
shoot
shoot-at
show
sit
sit-on
sleep
slip
smash
smell
smoke
sneeze
speak
step-on
stream
supply
support
surprise
surround
swim
swallow
take
take-off
taste
tear-down
throw
touch
trade
transfer
transport
travel
use (not <u>used</u>-<u>to</u>)
wake-up
walk
work-for

G_6 . . . cont.

alienate
appear
*assume
attract
change
come
communicate-with
*dream
*gather
get (G_4)
*guess
have
*hope
organize
*predict
*provide
punish
reward
*seem
*suppose
*wonder

etc.

APPENDIX B-1

In the following discussion we will consider a small-scale experiment performed on a scene taken from Winograd's thesis, in which a native speaker was asked to write an English description of a scene. Later, another person was asked to use this description to draw a picture of the scene. By studying the description and the picture drawn from it, we can make a number of interesting observations about the understanding of a human being.

First, the picture corresponding to sentence (2) in the description shows that, most typically, if X is on Y, then the surface of X which touches Y is smaller than or the same size as the surface of Y on which X rests.

Second, sentence (4) of the description states that a small red block is "near" the big red block, even though in the original picture this small block is <u>in front of</u> the large one; in the picture drawn from the description, the small red block is <u>beside</u> the large one. This shows that <u>near</u> specifies distance, but not direction.

Third, in sentence (6) "standing on end" is redundant since in the noun phrase <u>slab</u> has already been specified as "tall," which indicates that it must have a larger vertical dimension.

Fourth, judging from the figures corresponding to sentence (7)--"There is a blue pyramid in it"--the principle below was probably followed in interpreting the sentences:

> If X is in Y, then the size of the interior of X is less than the size of Y.

Fifth, the description of the scene in Appendix B-2 does not specify the size of the green block mentioned in sentence (2). In this case, using the principle described in my first comment, he could easily draw the picture by making the green block smaller than the red block it is resting on. Some other missing information is the specification of the size of the blue pyramid; a guideline for supplying this missing information is found in my fourth comment; although the subject could not get its absolute size from the description, he was able to establish its size relative to the box.

In Appendix B-4 is an interpretation of the description, using the same method we have been discussing. As this attempt shows, a full understanding of the scene is difficult to achieve without interacting with the real world. It is especially hard to identify the referents of pronouns, and to supply omissions of the type mentioned in my fifth comment, the size of BLOCK1 and PYRAMID2. In addition, the interpretation of <u>tall</u> and <u>flat</u> are pretty risky. There are also problems in interpreting the last sentence in the description. I tentatively assigned the following representation:

```
                     #SUBJ <=> #ACT
                            ⇑
((#OBJ) PART (#SUBJ) <-> (RIGHT-OF (SCENE))
                      |
                      L< (LEFT-OF (SCENE))
                      ⇑
              MES(*) <-> #SUBJ
```

where * stands for

```
(BLOCK2 REF (A))<->ON (TABLE1 REF (THE))
(BLOCK3 REF (A))<->ON (TABLE1 REF (THE))
(BLOCK4 REF (A))<->ON (TABLE1 REF (THE))
(SLAB1 REF (A))<->ON (TABLE1 REF (THE))
(BOX1 REF (A))<->ON (TABLE1 REF (THE))
```

This GIST matches partially with the GIST of Appendix B-4.

Since the table is neither described nor mentioned explicitly in conjunction with everything that is on it, it presents special problems. In short, it is difficult for the program to get complete information from an incomplete description, even though a human can do this easily. With humans, fully explicit linguistic descriptions are the exception, and may even be more difficult to understand if they are detailed to the point of being cumbersome. This is because human understanding comes not only from linguistic knowledge but makes extensive use of our understanding of the objects themselves and the relationships among them. In order to understand even a simple situation like the one in the example we use our knowledge of blocks and pyramids, and of how they look and behave in general. We may also employ sensory-motor knowledge gained by observing or manipulating the objects themselves. Thus, when information from these three components of knowledge are readily available, humans will employ all of them, and the information derived from the linguistic component will be incomplete. People are, however, capable of deriving a great deal of information from purely linguistic descriptions (and from playing "nonsense" language games that are really not nonsensical at all). The GIST, though unable to fill in this description to provide a complete representation of this scene because it lacks extensive general knowledge and sensory-motor knowledge, is able to provide an interpretation of the description itself, which can be supplemented with information from the other components.

There are advantages to having a separate linguistic representation, the chief one being flexibility, which is not only convenient, but one of the essential characteristics of human language. If we incorporate these other kinds of knowledge into linguistic knowledge and make our linguistic analysis too dependent on a specific type of situation, then we have robbed it of its flexibility and will be forced to devise new grammars for new situations. It is not necessary to interpret a sentence in terms of a specific situation before analyzing it grammatically.

APPENDIX B-2

A Native Speaker's Description Of A Scene Taken From Winograd's Thesis

(1) A hand is near a green block.

(2) The green block is on a red block.

(3) The big red block is on the table.

(4) A small red block with a small green pyramid on it is near the big red block.

(5) A big green block is on the table to the right and a little in front of the big red block. There is a tall red pyramid on this green block.

(6) Behind this big green block there is a tall, flat, blue slab standing on end.

(7) There is a very large box to the right of this blue slab. There is a blue pyramid in it.

(8) Going from left to right, a large red block a small red block, a large green block, a tall blue slab, and a big box are resting on the table.

APPENDIX B-3

A Native Speaker's
Drawing Of A Scene
From The Description In
Appendix B-2

APPENDIX B-4

An Interpretation Of The Description In Appendix B-2

(1)' (HAND1 PART (SHRDLU) REF (THE)) <-> NEAR (BLOCK1 REF (A))

(2)' (BLOCK1 REF (THE)) <-> ON (BLOCK2 REF (A))

(3)' (BLOCK2 REF (THE)) <-> ON (TABLE1 REF (THE))

(4)' (BLOCK3 REF (A)) <-> NEAR (BLOCK2 REF (THE))

(PYRAMID1 REF (A)) <-> ON (BLOCK3 REF (A))

(5)' (BLOCK4 REF (A)) <-> ON (TABLE1 REF (THE))

(BLOCK4 REF (A)) <-> TO-THE-RIGHT-OF (BLOCK2 REF (THE))

(BLOCK4 REF (A)) <-> IN-FRONT-OF (BLOCK2 REF (THE))

(PYRAMID2 REF (A)) <-> ON (BLOCK4 REF (THIS))

(6)' (SLAB1 REF (A)) <-> BEHIND (BLOCK4 REF (THIS))

(SLAB1 REF (A)) <-> ON (#GOAL)

(7)' (BOX1 REF (A)) <-> TO-THE-RIGHT-OF (SLAB1 REF (THIS))

(PYRAMID3 REF (A)) <-> IN (#GOAL REF (IT))

(1)' (BLOCK1 <=> (COLOR (GREEN)))

(2)' (BLOCK2 <=> (COLOR (RED)))

(3)' (BLOCK2 <=> (SIZE (BIG)))

(4)' (BLOCK3 <=> (COLOR (RED)))

(BLOCK3 <=> (SIZE (SMALL)))

(PYRAMID1 <=> (SIZE (SMALL)))

(PYRAMID1 <=> (COLOR (GREEN)))

(5)' (BLOCK4 <=> (SIZE (BIG)))

Appendix B-4 . . . cont.

```
       (BLOCK4<=>(COLOR (GREEN)))

       (PYRAMID2<=>(HEIGHT (TALL)))

       (PYRAMID2<=>(COLOR (RED)))

(6)'   (SLAB1<=>(HEIGHT (TALL)))

       (SLAB1<=>(THICKNESS (FLAT)))

       (SLAB1 <=>(COLOR (BLUE)))

(7)'   (BOX1<=>(SIZE (VERY LARGE)))

       (PYRAMID3<=>(COLOR (BLUE)))
```

REFERENCES

Charniak, E. 1972. "Toward a Model of Children's Story Comprehension," AI-TR-266, M.I.T.

Nakajima, S. 1974a. "Some Considerations for Conversational Information Systems," M.S. Thesis, University of Illinois.

_____. 1974b. "Syntax as Entertainment," CSL Memo.

_____. 1975a. "Linguistic Variable ACTs and their Frames," CSL Memo.

_____. 1975b. "Toward a Model of Story Understanding by DS," CSL Memo.

_____. 1975c. "Descriptive Structure and Computer Understanding of Sentences," CSL Memo.

_____. 1975d. "A Computational Imagining of the Block World by a Robot," CSL Memo.

Rutter, P. "Navy Data Base Read Package," CSL AI Working Paper 3, University of Illinois, 1974.

Waltz, D. 1975. "Natural Language Access to a Large Data Base: An Engineering Approach".

Winograd, T. 1972. Understanding Natural Language, Academic Press, New York.