TALKING WITH A ROBOT IN ENGLISH *

L. Stephen Coles
Stanford Research Institute
Menlo Park, California

Abstract

This paper gives an overview of the present status and future plans of a research project aimed at communicating in natural language with an intelligent automaton. The automaton in question is a computer-controlled mobile robot capable of autonomously acquiring information about its environment and performing tasks normally requiring human supervision. By natural language communication is meant the ability of a human to successfully engage the robot in a dialog using simple English declarative, interrogative, and imperative sentences. Communication is accomplished by means of a natural language interpretive question-answering system (ENGROB) consisting of six distinct components: a syntax analyser, a semantic interpreter, a model of the robot's environment, a deductive, automatic theorem-proving system, an English output generator, and a repertoire of basic robot capabilities for sensing and manipulating the environment. An example is given that illustrates the type of processing done by each component, and the nature of component interactions.

Descriptive Terms: Natural language, English, Systems, robots, intelligent automata.

X, Introduction

The advent of computer-controlled robots capable of autonomously sensing a real-world laboratory environment, constructing a dynamic model of such an environment, and manipulating various objects in that environment has provided a unique opportunity for research in computational linguistics. The question of how one might apply current linguistic theory in the design of a conversational, natural language robot communication system is certainly an interesting problem in its own right. It is the author's contention, however, that some aspects of linguistic theory itself could be significantly influenced by research in this area. We will examine the argument for this position in the conclusion.

There are at least three projects throughout the country attempting to design integrated artificial intelligence systems that include a computer-controlled automaton of the type described above. Intelligent hand-eye machines are being investigated in two separate programs, one under Professors M. L. Minsky and S. Papert at MIT, and the other under Prof. J. McCarthy at Stanford University. At Stanford Research Institute we are endeavoring to build a <u>mobile</u> automaton capable of exploring a real-world laboratory environment. A more general discussion of the goals of the SRI robot project may be found in a paper by Nilsson;[+][1] while details of robot problem-solving capabilities may be found in Green.[2]

The present paper is based on work in progress on a system called ENGROB, a natural-language, interpretive, question-answering system used to communicate with the SRI robot in simple English sentences. Because ENGROB is not yet fully implemented, some of what follows should be considered to be speculation. However, simple examples based on running programs will be used to illustrate the nature of the problems encountered in natural-language communication with the SRI robot. The appendix gives a representative list of English sentences that can be processed by ENGROB, together with their translations. The list is perhaps the simplest way for the reader to obtain an intuitive feel for ENGROB*s current level of performance.

The basic paradigm that has guided the development of ENGROB is (1) translate English statements, questions, and commands into a formal language based on the first-order predicate calculus; (2) perform any necessary deductive Inferences based on the current set of operational axioms and the current state of the robot's model of the environment; and (3) generate as appropriate an English output sentence and/or a sequence of primative functions within the set of basic robot capabilities for sensing and manipulating the environment. The Initial translation to the predicate calculus is accomplished by means of syntactic and semantic analyses based on a large collection of productions or pattern-operation rules, while deductions are carried out by means of a resolution-based automatic theorem prover. English output sentences are produced by translating answer expressions in the predicate calculus into their English equivalents, again by means of a set of productions. For

The research reported here was supported by the Rome Air Development Center under Contract No. AF30(602)-4124, by the Rome Air Development Center and the Advanced Research Projects Agency under Contract No, P30602-69-C-0056, and by the Air Force Office of Scientific Research under Contract No. AF19(628)-5919.

---

* References are listed at the end of this paper.

** ENGROB depends upon the work of many individuals in our group. B, Raphael, C.Green, R.Yates,J,Munson, and N. Nilsson have contributed to the theorem-proving component; L.Chaitin and C Fennema have developed the FORTRAN component; R.Duda and P.Hart contributed to the vision component; and A.Robinson aided in the implementation.

comparison with standard terminology used by linguists, the predicate calculus plays essentially the same role as a natural-language "deep structure" (cf. Chomsky*), and it is proper to regard the predicate calculus in ENGROB as a sort of deep structure (cf.Bohnert and Backer[4]).

If, during the course of translating the source statement the semantic analyzer uncovers an unclear portion of the text or an unresolvable ambiguity, the system assigns to the user a series of questions on the unclear portions. The character of these questions depends in part on the context of the conversation. The user's replies to these questions may be regarded as paraphrases of the unclear portions. The system then reanalyzes the text. If necessary, the system again assigns questions to the user, and in this manner establishes a dialog between the user and the robot. By means of this dialog the user continually simplifies the formulation of his task specification until it is completely understood by the system. An example will illustrate how this paradigm works in practice.

II.Organization of Robot Software

Figure 1 shows the organization of the robot software. The left-hand side of the figure (LISP) essentially corresponds to the ENGROB system. The right-hand side (FORTRAN) essentially corresponds to the primitive functions and reflexive actions necessary to support the robot's basic sensory-motor Interaction with the real world. These functions are programmed for the most part in FORTRAN and machine language, while the higher-level routines in ENGROB are programmed for the most part in LISP. Interaction between the FORTRAN and LISP components is facilitated by a specially designed monitor called the VALET. The subcomponents of ENGROB Indicated In Fig. 1 and the flow of control between them bears strong resemblance to the organization originally suggested by Bobrow for his SENSE natural-language question-answering system

More precisely, ENGROB is composed of six major components which we shall consider in turn: a syntax analyzer, a semantic Interpreter, an axiom model, an inferential component, an output-sentence generator, and an output-action generator.

III. Syntax Analyzer

The syntax analyzer is based on a transformational grammar for a subset of English imperative, declarative, and Interrogative sentences. The vocabulary Is unrestricted Insofar as adjectives and nouns are concerned and in this sense ENGROB's analyzer Is similar to a transformational parser proposed by Thorn.[6] The grammar consists of two subcomponents: a transformational component serves the purpose of decomposing complex sentences Into their simpler kernel sentences so that parsing can be accomplished by the base component in a more efficient manner. The base component is derived from a simple phrase structure grammar written in Backus-Naur Form.

The use of transformations in the syntax analyzer is currently restricted to string transformations that map terminal symbols into other terminals. The most conspicuous use of transformations in the current grammar is to recognize Interrogative sentence forms either through subject-predicate inversions or interrogative pronouns, and to map them into their corresponding declarative-sentence forms. These transformed declarative sentences are then passed to the base component for complete analysis. In this manner, by adding a dozen transformations to the transformational component, we eliminate the need for practically doubling the size of the declarative base analyzer merely to handle interrogative sentences. Another simple but important use of transformations is in mapping plural noun and verb forms into their corresponding singular form to facilitate unique identification in the deep structure.

The base component of the grammar was taken essentially without change from the GRANIS system,[8] a predecessor of ENGROB developed by the author for application to graphical question-answering systems. Historically, this base component was Implemented as a set of productions in Formula Algol. With small effort these productions were then transliterated into LISP (with their control programs) in order to maintain compatibility with the remainder of the system. In previous work this base component was expanded by first adding new rules to the BNF grammar, applying the Earley Algorithm[9] to the BNF grammar, and then post-editing the resulting productions to obtain an efficient one-pass, syntax-directed recognizer for the BNF grammar. In more recent work with ENGROB, however, it has been found to be more convenient to work directly with the productions themselves, abandoning the original BNF grammar. Thus, under the current strategy the productions are treated as a separate programming language for grammar construction, and new productions are added directly to the recognizer as needed.

The form of the productions is as follows:

LI: or / > 0 / Y * L2;

where LI and L2 are labels, cr and 0 are strings, > indicates a replacement operation, V Is a sequence of semantic productions, the asterisk indicates a "read" operation taking the next word in the input string and placing it at the top of the syntactic stack, and the semicolon Is a punctuation mark delimiting the scope of the production. LI, >, 0, v, and the asterisk are optional characters, while both diagonal bars, or, L2 and the semicolon are mandatory for each production. Flow of control for the productions is defined as follows: If, in the course of analysis, control reaches the cluster of productions labeled LI and the right-hand portion of the contents of the syntactic stack is an Instance of the pattern string or, then:

(1) Replace that portion of the stack that was matched by a, with 0 (which will In general depend on the portion of the stack matched, since free-class variables become bound if the match is successful).

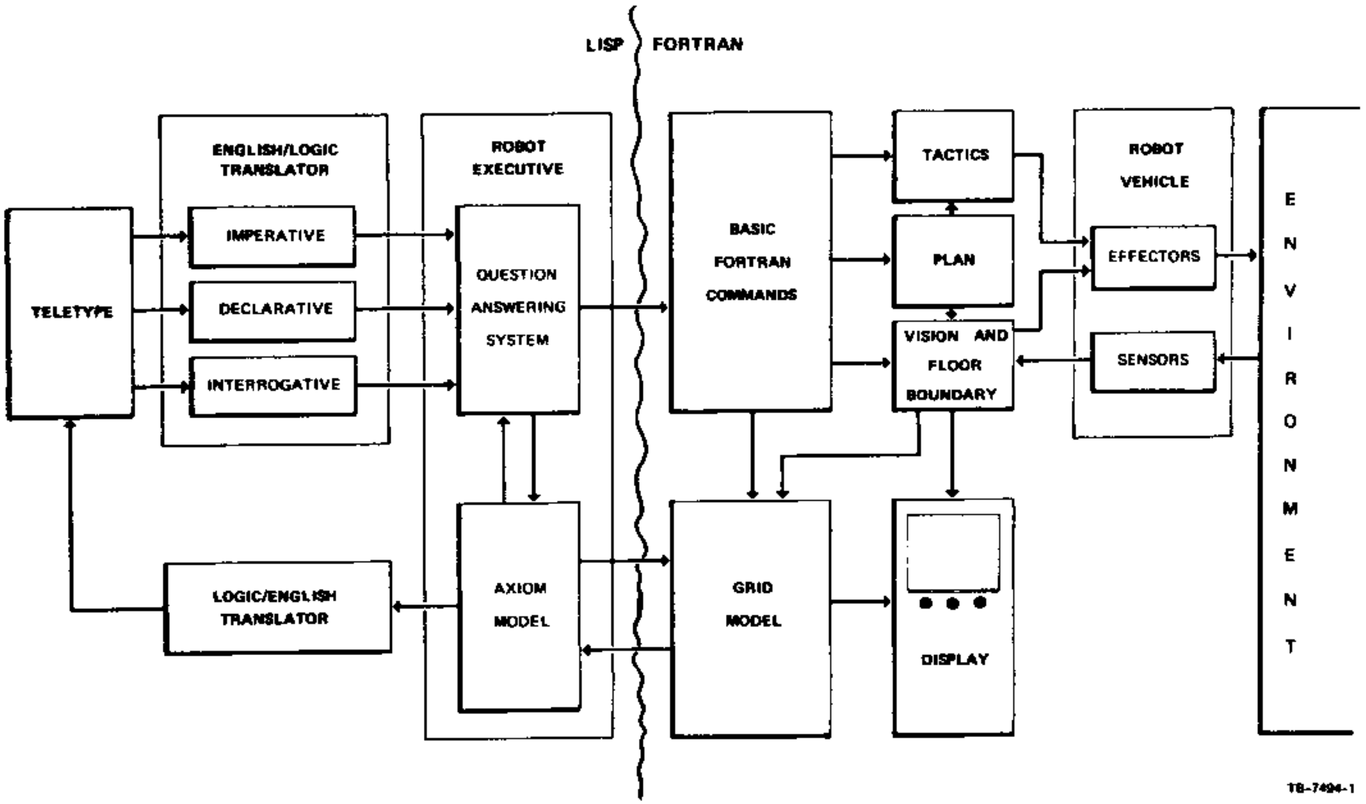

Figure 1 ORGANIZATION OF ROBOT SOFTWARE

(2) Execute the sequence y if present.

(3) If an asterisk is indicated, read a new word into the syntactic stack from the input string.

(4) Go to the cluster of productions labeled L2.

Otherwise, if the stack fails to match the pattern string or, control is passed to the next production in the sequence. Possible pattern elements for the pattern string a include terminal constants, class variables defined in terms of terminal constants or boolean combinations of other classes, the pattern $1, which can match a single arbitrary constituent, or the pattern $, which may match an arbitrary number of arbitrary constituents much as in the COMIT language. So that particular values of the stack may be referenced in the replacement portion, 0, the result of any successful match may cause optional extraction variables to be bound to the value of a match with a class variable. More explanation together with examples of this process may be found in Ref. 7.

Transformational productions have the same form as base component productions except for the fact that the scanning for a match is from left to right across the entire sentence rather than from right to left across the syntactic stack. Any pattern-element sequence can be quoted, indicating that pattern matching is to be accomplished at the character level in a particular word rather than at the lexical level, and in this manner testing for plurals and standard suffixes or prefixes can be achieved.

One of the difficulties uncovered by our research on the syntactic component was a purely pragmatic one. We were disconcerted to find that as we added more and more transformations to the grammar, the time for processing kernel sentences (to which transformations are inapplicable) increased in proportion to the complexity of the grammar. This was a clearly unacceptible state of affairs, since in an ideal implementation the processing time for kernel sentences should remain essentially constant, regardless of the number of transformations. This led us to the notion of distributing the transformations throughout the base component, thereby blurring the distinction between the two subcomponents in our implementation, leaving us with a grammar that although technically not a transformational grammar, still has transformational power. Preliminary evidence shows that this approach yields a marked improvement in parsing efficiency, but we do not yet perceive any theoretical implications in this strategy.

IV, Semantic Interpreter

Translation of a well-formed English source statement into an equivalent well-formed formula in the first-order predicate calculus is accomplished by means of a set of semantic productions interleaved with the syntactic productions. The semantic productions have an identical form and flow control with the exception that the * operation is never used and the productions operate on a separate semantic stack. The method of integrating the syntactic and semantic analysis within a common production framework has been called Syntax-Directed Interpretation, and examples of this process can also be found in Ref. 7.

The appendix shows thirty sample sentences together with their translation into the predicate calculus. Declarative sentences, such as S1 - S10, are entered directly into the question-answering system as axioms; interrogative sentences, such as Q1 - Q10 are submitted as assertions to be proved by the inferential component. Simple requests, such as C1 - C4, are translated directly into FORTRAN commands (cf. Table 1) and passed to the command interpreter. Complex imperatives, such as C5 - C10, are treated as assertions about the possibility of discovering a sequence of primitive actions that accomplish a task subject to specified constraints. Therefore, they are treated in much the same fashion as questions.

Most of the predicates are self-explanatory, but a detailed look at one of the more difficult sentences, S10, will serve as a guide for understanding the remaining formulas. The initial determination by the syntactic component is the applicability of the active/passive voice transformation which then maps the given sentence into its active form: "John pushed the tall box." Next, the base component recognizes the past tense of the verb "push" and sets the Time predicate accordingly. The adjective "tall" and the noun "box" each map into the In predicate. The final well-formed formula can be interpreted roughly as follows: There exists a state s, an object x, and places y and z such that s is equal to a state obtained from some initial state S_i by having John push x from y to z, where x is characterized by being both in the class of tall objects and in the class of boxes, and, furthermore, s happened in the past. Note that the letter R (for Robot) in some of the other sentences corresponds to the antecedent of the pronoun "you".

The predicate calculus has thus far proved to be a sufficiently powerful internal representation for capturing the meaning of our simple English sentences. As our grammar expands to handle increasingly complex sentences, it will probably continue to serve as our "deep structure" representation with a few minor modifications. Of course an additional advantage of using predicate calculus as an internal representation for the meaning of English sentences is that we then have a common language for representing both the linguistic and nonlinguistic information about the world vital to intelligent communication with the robot. Moreover, we then capitalize on effort expended by logicians in establishing the logical properties of the predicate calculus, and can precisely describe the class of deductions possible within our framework. The logical limitations on competing representations, such as directed graph networks or description lists, are not always obvious.

One of the theoretical issues uncovered by our work on the semantic component was how to find a canonical set of predicates for describing actions. What is being sought is a reasonably small but complete list of predicates that could exhaustively describe all the essential features of an action. A tentative list based on work by N. Rescher[10] is presented below:

	Predicate	Question
(1)	Agent (x,y)	Who did it?
(2)	Act (x,y)	What did he do?
(3)	Object (x,y)	To what or whom did he do it?
(4)	Setting:	In what context did he do it?
	Itime (x,y) Ftime (x,y)	When did he do it?
	Iloc (x,y) Floc (x,y)	Where did he do it?
	Circ (x,y)	Under what circumstances did he do it?
(5)	Modality:	How did he do it?
	Means (x,y)	By what instrument or method did he do it?
	Manner (x,y)	In what manner did he do it?
(6)	Rational:	Why did he do it?
	Cause (x,y)	What caused him to do it?
	Aim (x,y)	With what intent did he do it?
	Mentality (x,y)	In what state of mind did he do it?

Our basic premise is that the adequacy of any deep-structure representation should be measured by the class of questions that can be easily answered by the data when represented in that form; hence the requirement for closely tying the predicates to questions that can be reasonably asked about an action. We are not yet certain that the above list is generally adequate, but for purposes or our robot work it seems to be sufficient for the time being.

V. The Axiom Model

Three kinds of information are contained in the axiom model: geometric relationships represented in the grid model, rules describing constraints on the robot's capabilities for sensing and manipulating the world, and descriptive information extracted from declarative sentences obtained during conversation with humans. The grid model describes the position, size, and orientation of various objects and obstacles by partitioning a plan view of the robot's environment and imposing a cartesian coordinate system. Axioms about the position and orientation of various objects including the robot are entered into the axiom model automatically as they are updated in the grid model. Axioms that describe the initial and boundary conditions for various sequences of primitive FORTRAN commands are permanently entered here, and are used during problem-solving and question-answering operations. The axiom model grows dynamically during the course of conversation as humans type declarative sentences, since these statements are translated into the predicate calculus and entered directly into the store of axioms that can be used for future inferences.

VI.The Inferential Component

Deductions are implemented by means of a highly efficient, automatic, deductive theorem-proving system, QA3, developed by Green and Raphael[11] and based on Robinson's resolution procedure. QA3 discovers proofs by refutation. To prove a theorem by refutation, one first hypothesizes the negation of the theorem and then attempts to obtain a contradiction, if one exists, by attempting and then failing to construct a model that satisfies both the axioms and the negation of the theorem. If such a model cannot be found, then it has achieved a constructive proof of the affirmative statement of the theorem, and can answer not merely YES or NO as to whether the original hypothesis was a theorem, but also for what values of the existentially quantified variables the theorem will be valid. It is this important feature of QA3, as a theorem prover, that permits its application to question answering and problem solving.

QA3's efficiency is greatly enhanced by the addition of a number of completeness-preserving heuristics—i.e., heuristics that limit the scope of search for a proof without violating the logical completeness of the basic resolution procedure. The discovery of new heuristics of this type appears to be a fruitful area for future research.

VII. The Output-Sentence Generator

Output sentences are produced by means of a small generative grammar based on the same productions described earlier. Thus, we see a very wide application of this sort of rewrite rule appearing in all of the linguistic components of this system. The form of the reply sentence is frequently determined by applying a simple transformation to the input question or command. For example, "Will you do x?" may give rise to "Yes, I will." or "No, I will not do x." Occasionally, however, the output sentence will have a nontrivial syntax—i.e., one that is not immediately obtainable from a simple transformation of the input. For example, "Move ten feet forward," may give rise to the reply "I can move only five feet because there is a wall in front of me." Here we see that semantic information contained in the axiom model determines, in part, the form of the reply.

VIII. The Output-Action Generator

The result of most imperative sentences (as well as certain interrogative sentences that require for their answer not only information contained In the model, but also Information that must be obtained by inspection of the real world) will be a sequence of two-letter FORTRAN commands with appropriate arguments which are then passed to the FORTRAN subsystem for execution. Table 1 shows a partial list of these commands to give the reader a better understanding of the robot's repertoire of basic actions. Upon execution, each command returns information to ENGROB about its success and other sensory data acquired, if any, for Incorporation into the model. In this manner ENGROB can monitor the progress of the robot In executing a sequence of primitive commands, and reassign a new sequence or subsequence as necessary due to unanticipated obstacles.

ix AN EXAMPLE

Now that we have examined the various components of ENGROB individually, let us see how they actually interact by means of a concrete example. Consider the following scenario which we expect to accomplish during the next few months:

Scene: Two people are seated at teletypes in the robot room which is filled with various cubes and wedges.

Time: 2:45 p.m.

Person.: Bring me a small cube at 3:00 p.m.
Robot: There are two small cubes.
Person.: Bring me the smaller cube.
Robot: OK
Person.: Will you push a small cube?
Robot: Yes, I will push a small cube.
Person : When will you push the cube?
Robot: I will push the cube at 3:00 p.m.
Time: 3:01 p.m.
Robot: I have brought you a small cube.
Person : Thank you.

The first step in processing the sentence "Bring me a small cube at 3:00 p.m." is to translate it into the predicate calculus. The syntactic component establishes that it is a well-formed imperative sentence, and the semantic component actually carries out the translation, giving

$$\text{C:}\quad (\exists s,x)\{At(x,Person_1,s) \wedge In(x,Small) \wedge In(x,Cube) \wedge Time(s,1500)\}\ .$$

The "C" asserts that the logical type of the following wff is "command/' The wff Itself asserts that there exists a state s and an object x such that the object is at Person, in State s, the object is small, the object Is a cube, and the state occurs at time 1500.

The next step is to pass the wff to QA3 as an assertion to be proved. Let us assume that among our data base of facts about the environment we have:

$$At(OB_4, P_4, S_1)$$

$$At(OB_6, P_6, S_1)$$

$$In(OB_4, Small)$$

$$In(OB_6, Small)$$

$$In(OB_4, Cube)$$

$$In(OB_6, Cube)\ .$$

In addition, we have an axiom of the form

$$(\forall s,x,y,z)\{At(x,y,s) \Rightarrow At(x,z,Push(R,x,y,z,s))\}$$

meaning that if an object x is at location y in state s, then it will be located at z in the state that results from the robot, R, pushing x from y to z. Furthermore, we know that Time (x,y) is an evaluable predicate. Then, under the condition that Time(s,1500) evaluates to true, QA3 will reply: yes, if $x = OB_4$ and $s = Push(R,OB_4,P_4,Person_1,S_1)$
or $x = OB_6$ and $s = Push(R,OB_6,P_6,Person_1,S_1)$.

This reply in turn guides the generation of the output sentence, "There are two small cubes" by the output sentence generator. The conversation continues, and assuming $Size(OB_6) > Size(OB_4)$ obtained by perceptual rather than linguistic information, the output action generator will submit the command "PU x_1,y_1,r,x_2,y_2," when clock time is 1500, where (x_1,y_1) are the coordinates of P_4, r is the radius of OB_4, and (x_2,y_2) are the coordinates of $Person_1$.

The conversation with $Person_2$ also yields translations into the predicate calculus of the form:

$$\text{Q:}\quad (\exists s,t,x,y,z)\{Eq(s,Push(R,x,y,z,S_1) \wedge In(x,small) \wedge In(x,cube) \wedge Time(s,t) \wedge Future(t)\}$$

with replies from QA3 of the form: yes, if

$$s = Push(R,OB_4,P_4,Person_1,S_1)$$

$$t = 1500$$

$$x = OB_4$$

$$y = P_4$$

$$z = Person_1\ .$$

These replies are then used to generate the answer sentences.

The reply sentence "I have brought you a small cube" is generated automatically by the successful execution of the PU FORTRAN command.

X. Implications for Linguistics

The complexity of the processing necessary to accomplish the superficially simple-minded task described in the previous section is enormous. In fact, the syntactic and semantic analysis necessary to translate these expressions into the predicate calculus and the transformations necessary to generate the output replies comprise less than half of the total processing required. The greatest portion of the processing time is consumed by the theorem prover, QA3, in establishing the feasibility of composing a sequence of primitive functions which, when executed, will accomplish the desired goal. Preliminary evidence indicates that as humans desire to have more substantive conversations with the robot, inferential requirements will grow, with the result that QA3 will consume a still greater proportion of the total processing time.

During a demonstration, many people, their enthusiasm whetted by some preliminary success with a comparatively trivial command like "Turn right", will go on to pose a problem for the robot that is dramatically beyond its present capability. There seems to be a machine-like regularity in most humans' lack of appreciation for the tacit assumptions regarding geometric space-time relationships and the volume of unarticulated knowledge about the possibility of, and constraints on, certain kinds of behavior that are implicit in the simple English sentences they can type to the robot. This phonomenon is vaguely reminiscent of the master chess player who is incapable of articulating by what principles he is able to play master-level chess to the designer of a chess-playing program.

As I see it, the implication this observation has for linguistics is as follows: Insofar as linguists seek to explain how people "understand" language, they will have to shift some of their attention away from the grammatical aspects of language--generation and parsing sentences--and focus more attention on how people bring their immense data base of knowledge about the world to bear in a relevant manner on the comprehension of some string of lexical items in the context of some particular universe of discourse. And, furthermore, they will have to focus on the methods by which people bring their knowledge to bear even when inferences are required to several levels of indirectness.

Because of the enormous complexity of this total process, and because humans appear to do it in what seems to be negligible time and effort, there is a strong temptation to describe the process as some kind of Gestalt phenomenon, not describable in terms of a collection of analytic procedures. Based on the preliminary results of experiments with our robot, however, I suggest that such an interpretation is erroneous. The fact that humans are largely unaware of all the linguistic analysis and data analysis they perform, and that they can perform it quickly, does not constitute evidence that they don't do analysis. Our robot can perform simple tasks today that occasionally provide surprising evidence for its understanding of language, even though that understanding is limited, the processing time is lengthy, and the motions of the vehicle are awkward.

The technology of robot hardware is bound to improve, as is the technology of computer hardware. In my judgement a conclusive demonstration of robot understanding of natural language is still a long way in the future, but ENGROB does serve as a demonstration that computer understanding of language that refers to the real world is possible. Furthermore, it shows that one could build a system around the principles discussed above that would permit robots and men to communicate in restricted English in real-world environments.

XI Conclusion

We have been discussing problems in the design and organization of a computer program that can permit robots and people to communicate in natural language. Progress on these problems thus far has been limited to a few simple scenarios that systematically exercise all of the basic capabilities of the hardware. Work is underway, however, on each of the six components of ENGROB. The transformational grammar is being extended to include nonterminal transformations; the predicate calculus is being expanded to handle a larger family of quantifiers; the class of updatable predicates will be augmented in the axiom model; QA3's heuristics are being refined; the output sentence generator will subsequently draw on QA3 for producing semantically relevant replies; and the action generator will have closer feedback with reality.

Before creating a false sense of optimism that dramatic improvements are just over the horizon, I might add that even if appropriate progress were made in each of the components, there still remain, among other problems, enormous systemic difficulties in integrating the components into a functioning whole properly embedded within a complex time-sharing system operating in a hardware environment of partial uncertainty. Frequently, one spends as much of one's time on these systemic problems, getting the robot to operate on a day-to-day basis, as on the major theoretical issues. Other difficulties appear on the horizon. Can we really ever get to investigate the nontrivial problems we would like to, within the memory-response-time limitations of our hardware? Will the predicate calculus prove inadequate as a natural-language deep structure, even when augmented by probabilistic, multi-values, or modal logic? Can the semantic component participate in the parsing process so as to resolve lexical and syntactic ambiguity with respect to the universe of discourse? Will our aspirations falter because the vision routines for years to come will never be able to recognize anything more complex than the difference between a cube and a triangular prism? Speculation of this kind is sobering, but our immediate goals are well defined, and only future research will tell whether our underlying optimism Is justified.

Finally, let us return to the earlier conjecture made in the introduction that certain portions of linguistic theory itself will be influenced by natural language communication with robots. The argument goes as follows: First, because robots provide the computer with a "window on the real world," they offer a host of new opportunities for empirically studying the relationship between language and reality* But it is this relationship that falls by definition under semantics, precisely the portion of linguistics that has received comparatively little theoretical attention thus far[1]. 13,14,16 in particular, the work reported in this paper having to do with space-time relationships, such as illustrated in the sample scenario, has forced us to think more carefully about how to encode the meaning of statements about real-world activities. Clearly, statements of this sort, which reference space-time relationships, abound in all human conversation as well as in the most elementary children's books, and an adequate model of these relations must be an essential ingredient in any theory of semantics. Robots will serve in a sense as a laboratory for testing the adequacy of our semantic representations and our logics, and ultimately may reveal new approaches to these basic questions.

In addition, robots have a number of purely philosophical implications. For the first time we have an opportunity to empirically investigate such important philosophical questions as "free will" or "self-awareness". We will be required to define in a precise and operational manner such concepts as possibility and necessity as well as other concepts such as can, cause, knows, believes, understands, and so on. These in turn must be based on epistemologically and metaphysically adequate representations of reality together with logical formalisms suitable for inference making and problem solving.[16] Here again robots will serve as a basis for empirical investigations that heretofore could be conducted only from the armchair.

APPENDIX

Sample English Sentences With Translations Into Predicate Calculus

Declarative Sentences

S1.	All men are mortal.	$(\forall x)\{In(x,man) \Rightarrow In(x,mortal)\}$
S2.	If John is a man then he is mortal.	$In(John,man) \Rightarrow In(John,mortal)$
S3.	John and Fred are tall thin men.	$In(John,tall) \wedge In(John,thin) \wedge In(John,man) \wedge In(Fred,tall) \wedge In(Fred,thin) \wedge In(Fred,man)$
S4.	Some tall men are not boys.	$(\exists x)\{In(x,tall) \wedge In(x,man) \wedge \sim In(x,boy)\}$
S5.	No man is a woman.	$\sim(\forall x)\{In\ (x,man) \Rightarrow In(x,woman)\}$
S6.	John has two hands.	$Hasp(John,hand,2)$
S7.	Every hand has five fingers.	$(\forall x)\{In,(x,hand) \Rightarrow Hasp(x,finger,5)\}$
S8.	Anything that is green is a tall thin box.	$(\forall x)\{In(x,green) \Rightarrow In(x,tall) \wedge In(x,thin) \wedge In(x,box).$
S9.	Any box smaller than a green cube that is on the right is a red and white prism.	$(\forall x)\{In(x,box) \wedge (\exists y)\{Smaller(x,y) \wedge In(y,green) \wedge In(y,cube) \wedge (\forall z)\{Right(y,z)\}\} \Rightarrow In(x,red) \wedge In(x,white) \wedge In(x,prism)\}$

---

*It is not implied here that robots are necessarily the only way that this might be done, but rather one of the more convenient methods of achieving this goal.

S10. The tall box was pushed by John.	(∃s,x,y,z){Eq(s,Push(John,x,y,z,Si)) ∧ In(x,tall) ∧ In(x,box) ∧ Time(s,PAST)}

Interrogative Sentences

Q1. Is there a man?	(∃x)[In(x,man)}
Q2. Is Jane a man?	In(Jane,Man)
Q3. Who is Jane?	(∃x){In(Jane,x)}
Q4. How many fingers does John have?	(∃x){Hasp(John,finger,x)}
Q5. Which box is the cube near the door?	(∃x){In(x,box) ∧ In(x,cube) ∧ (∃y){Near(x,y) ∧ In(y,door)}}
Q6. Are there any boxes on your left?	(∃x){In(x,box) ∧ Left(x,R)}
Q7. Where are you?	WH *
Q8. Will you push the box?	(∃s,x,y,z){Eq(s,Push(R,x,y,z,Si) ∧ In(x,box) ∧ Time(s,FUTURE)}
Q9. When will you push the box?	(∃s,t,x,y,z){Eq(s,Push(R,x,y,z,Si) ∧ In(x,box) ∧ Time (s,t) ∧ Future(t)}
Q10. Did you push a box yesterday?	(∃s,x,y,z){Eq(s,Push(R,x,y,z,Si) ∧ In(x,box) ∧ Time(s,YESTERDAY)}

Imperative Sentences

C1. Stop.	ST *
C2. Turn around.	TU 180., *
C3. Move ten feet.	MO 10., *
C4. Turn right 45 degrees.	TU-45., *
C5. Go to the big red prism.	(∃s,x){At(R,x,s) ∧ In(x,big) ∧ In(x,red) ∧ In[x,prism)}
C6. Push the black box on top of the platform.	(∃s,x,y,z){Pushed(x,s) ∧ In(x,black) ∧ In(x,box) ∧ On(x,y) ∧ In(y,top) ∧ Of(y,z) ∧ In(z,Platform)}
C7. Move the wedge that is on the left to the platform.	(∃s,x,y){At(x,y,s) ∧ In(x,wedge) ∧ (∀z){Left(x,z)} ∧ In(y,Platform)}
C8. Roll up the ramp next to the platform.	(∃s,x,y){On(R,x,s) ∧ In(x,Ramp) ∧ Next(x,y,) ∧ In(y,platform)}
C9. Collect all the cubes into the center of the room.	(∀x)(∃s,y,z){At(x,y,s) ∧ In(x,cube) ∧ In(y,center) ∧ Of(y,z) ∧ In(z,room)}
C10. Explore John's office.	(∃s,x){Explored(x,s) ∧ In(x,office) ∧ Of(x,John)}

NOTE: As of this writing, sentences C6, C7, and C10 have not yet been executed as robot tasks, although they can be correctly translated within ENGROB into the predicate calculus. We expect to have the robot actually carry out these commands during the next few months.

BASIC FORTRAN COMMANDS

Command	Explanation
ST	Stop.
CL	Clear the model.
RM	Read the model.
MO N	Move forward N feet.
MA	Display a map of the room
TU N	Turn counterclockwise N degrees.
XR N	Set the current x-coordinate of the robot to N.
YR N	Set the current y-coordinate of the robot to N.

Table 1 Continued

BASIC FORTRAN COMMANDS

Command	Explanation
XG N	Set the x goal coordinate to N.
YG N	Set the y goal coordinate to N.
AN N	Set the current angle of the robot to N degrees.
GO	Go to the goal (by touch sensors only).
TE	Plan a journey to the goal using vision, and execute it.
PI X,Y	Take a picture at location (X,Y).
IR	Iris.
FO	Focus.
TI N	Tilt the camera N degrees.
PA N	Pan the camera N degrees.
SC N	Scan the room in N steps with the range finder.
OV N	Turn overrides on or off as a function of N.
PU X_1,Y_1,R,X_2,Y_2	Push the object located at (X_1, Y_1) of Radius R to the goal location (X_2,Y_2).
WH	Print the current values of XR,YR, and AN.

REFERENCES

1. N.Nilsson, "A Mobile Automaton: An Application of Artificial Intelligence Techniques," paper to be presented at the International Joint Conference on Artificial Intelligence, Washington, D.C., May 1969.

2. C.Green, "Application of Theorem Proving to Problem Solving," paper to be presented at the Internation Joint Conference on Artificial Intelligence, Washington, D.C., May 1969.

3. N. Chomsky, *Aspects of the Theory of Syntax* (MIT Press, 1965).

4. H.G. Bohnert and P.O. Backer, "Automatic English-to-Logic of Grammar" (IBM, Yorktown Heights, N.Y., March 1966).

5. D.G. Bobrow, "Problems in Natural Language Communication with Computers," *IEEE Trans. on Human Factors in Electronics*, Vol. HFE-8, pp. 52-55 (March 1967).

6. J.P. Thorn, P. Bratley, and H. Dewar, "The Syntactic Analysis of English by Machine," *Machine Intelligence 3*, Michie, ed., pp. 281-309 (Edinburgh University Press, Edinburgh, Scotland, 1968).

7. L.S. Coles, "An On-Line Question-Answering System with Natural Language and Pictorial Input," *Proc. 23rd National ACM Conference*, pp. 157-167 (Brandon Systems Press, 1968).

8. L.S. Coles, "Syntax Directed Interpretation of Natural Language," Ph.D. Thesis, Carnegie-Mellon University, Pittsburgh, Pa. (1967).

9. J.C. Earley, "Generating a Recognizer for a BNF Grammar," Comp. Center Paper, Carnegie-Mellon University, Pittsburgh, Pa.(June 1966).

10. N. Rescher, "Aspects of Action", in *The Logic of Decision and Action*, Rescher,ed. (University of Pittsburgh, Pittsburgh, Pa. 1967).

11. C.Green and B. Raphael, "The Use of Theorem-Proving Techniques in Question-Answering Systems," *Proc. 23rd National ACM Conference*, pp. 169-181 (Brandon Systems Press, 1968).

12. C. Green, "Theorem Proving by Resolution as a Basis for Question-Answering Systems," in *Machine Intelligence 4*, Michie and Meltzer, eds. (Edinburgh University Press, Edinburgh, Scotland, 1969).

13. J.J. Katz and J.A. Fodor, "The Structure of a Semantic Theory," *Language*, Vol. 39, pp. 170-210 (1963).

14. U. Weinreich, "Explorations in Semantic Theory," in *Current Trends in Linguistics*, T.A. Sebeok, ed. Vol. 3, pp. 395-477 (Mouton and Co., The Hague, 1966)

15. M.R. Quillian, "Word Concepts: A Theory and Simulations of Some Basic Semantic Capabilities," in *Behavioral Science* (September 1967).

16. J.McCarthy, "Some Philosophical Problems from the Standpoint of Artificial Intelligence," in *Machine Intelligence 4*, Michie and Meltzer, eds. (Edinburgh University Press, Edinburgh, Scotland, 1969).