MASSACHUSETTS INSTITUTE OF TECHNOLOGY
A. I. LABORATORY

Artificial Intelligence
Memo No. 284

June 1973

PROPOSAL TO ARPA FOR CONTINUED RESEARCH ON A.I. FOR 1973

Marvin Minsky
Seymour Papert

Work reported herein was conducted at the Artificial Intelligence Laboratory, a Massachusetts Institute of Technology research program supported in part by the Advanced Research Projects Agency of the Department of Defense and monitored by the Office of Naval Research under Contract Number N00014-70-A-0362-0003.

Reproduction of this document, in whole or in part, is permitted for any purpose of the United States Government.

TABLE OF CONTENTS

	Preface	2
Part I	Sub-Projects and Milestones	3
1.0	Anticipated Milestones	3
1.1	The Language-Vision-Action Demonstration	3
1.2	Other Milestones: Vision and the Blocks World	5
1.3	Vision outside the Blocks World	6
1.4	Understanding English	7
1.5	Computers, Knowledge, and Intelligence	7
Part II	Logic, General Knowledge, and Common Sense	9
2.1	Polishing the Language Understanding Problem	10
2.2	More Powerful Problem-Solving with SHRDLU	11
2.3	New Models for Meaning	12
2.4	New Grammar Organizations	13
2.5	Theories of Typical Situations and Generic Expressions	15
2.6	Learning to Understand	17
2.7	Understanding Stories	19
2.8	Sussman's Program: HACKER	22
2.9	Knowledge About Proceedures and Their Likely Bugs	24
2.10	Goldstein's "Solution" of the Halting Problem	26
2.11	Goldstein's Program-Understanding Program	28
Part III	Computer-Controlled Vision and Manipulation	30
3.1	The Heterarchical Vision System	32
3.2	Generalization of Labelling Theory	32
3.3	Grouping and Tactile Scene-Analysis	33
3.4	Analysis of Curved-Line Drawings	35
3.5	Color Vision	36
3.6	Touch and Tactile Programming	37
3.7	"Low-Level Vision" Programs	38
3.8	Physical Knowledge and Stability	39
3.9	Liquids	39
3.10	Electronic Assembly: Another Robot "World"	41
3.11	Analyzing Complicated Objects	42
3.12	Groups, Descriptions, and Conflicts	43
Part IV	Development of Research Methods and Tools	44
4.0	Programming Languages for A.I.	44
4.1	PLANNER Progress	46
4.2	Parsing Strategy Research	48
4.3	Debugging Aids	49
4.4	Heterarchy	49
4.5	Automatic Theorem Proving	50
4.6	Research on Education and Cognitive Development	51

PROPOSAL TO ARPA FOR 1973

PREFACE

The Artificial Intelligence Laboratory proposes to continue its work on a group of closely interconnected projects, all bearing on questions about how to make computers able to use more sophisticated kinds of knowledge to solve difficult problems. This proposal explains what we expect to come of this work, and why it seems to us the most profitable direction for research at this time. The core of this proposal is about well-defined specific tasks such as extending the computer's ability to understand information presented as visual scenes, or in natural, human language. Although these specific goals are important enough in themselves, we see their pursuit also as tightly bound to the development of a general theory of the computations needed to produce intelligent processes. Obviously, a certain amount of theory is needed to achieve progress in this and we maintain that the steps toward a comprehensive theory in this domain must include thorough analysis of very specific phenomena. Our confidence in this strategy is based both on past successes and on our current theory of knowledge structure. Our proposed solutions are still evolving, but they all seem to revolve around new methods of programming and new ways to represent knowledge about programming. The field of Artificial Intelligence has made enormous progress in the past few years toward becoming a scientific subject. This proposal deals with our main goals, both for long-range research and for particular application areas. The general technical position of our 1970 and 1971 proposals still represents the direction of our approach. However, although those proposals are relatively explicit about the high-level problems motivating our research, they do not give a very clear picture of the actual projects, or of their practical consequences. In this proposal we shall concentrate on giving a more concrete view of our immediate goals. The next four sections are each directed to an "area" of application. It will be obvious that the concepts and methods used in these areas are closely related. Our intentions about the applications vary, naturally. In some cases we have major efforts toward completing effective operational prototypes. In the rest, we expect to produce demonstration prototypes, or only to develop some theory, clarify problems, and attempt to direct the attention of others to problems which we think need immediate attention.

Part I. Sub-Projects and Milestones

1.0 Anticipated Milestones

We define current and proposed sub-projects largely by relating them to milestone tests of achievement which we have set up to make the immediate goals and criteria of our work as unambiguous as possible. We do not try to predict the date of success for each milestone beyond saying that we would be surprised if any of them took as much as three years. It would be best to begin by discussing a milestone to whose achievement we are firmly committed because we are sure that it is in the state of the art and because it will have important theoretical and industrial consequences. It combines in a very interactive—more than merely additive!—way a large number of theoretically related sub-systems that have, for practical reasons, been built up separately.

1.1 The Language-Vision-Action Demonstration

This milestone will be passed when we can give the computer (equipped with eyes and hands) orders in brief, business-like language such as

> "I think there is a defective diode on this circuit board. Probably the third in the row on the left. Check and replace if necessary."

For this test to be meaningful, we must assume that

- the computer has no prior knowledge about the particular circuit board (e.g. in the form of a diagram or symbolic description).
- the computer is able to see the circuit board by natural vision (so, for example, there are no special marks on it like the magnetic characters on checks).
- the English expressions used really are free (e.g. there is no hidden format beyond the restriction to what a real, not particularly smart, human technician would accept as a straight, natural and clear instruction).

The achievement of this demonstration depends on realising a number of sub-goals of which some depend on parts of what is seen in the demonstration itself while others are more like the oil in the machine—no less vital and difficult for being invisible to the naive observer.

These sub-goals include:

— extending the domain of the vision system from blocks with plane surfaces to the more natural objects found in electronic componentry with their curves, highlights, colors, stripes, shadows and so on.

— corresponding extension of the manipulative ability.

— extending the natural language programs in several directions:
  — new domain (electronics instead of blocks);
  — modalities such as probable-possible-necessary;
  — modelling the person giving the instruction, so as to deal properly with such information as "I think..."

— further evolution of programming languages, computer control structures and debugging aids, in forms particularly suitable to work in Artificial Intelligence.

— theoretical problems in the area of representing knowledge, including "meanings" or "world models" constructed from visual and linguistic inputs.

This rough breakdown into sub-goals illustrates one of the major theoretical theses of our Laboratory. An "intelligence" is a complex system with a large number of interacting but separate parts. It can be understood and reproduced only by dealing specifically with these parts. We do not believe in the chimera of finding a single powerful "method" or "principle" which would give rise to intelligence the way the laws of Newton give rise to elliptic orbits. In surveying the variety of specific sub-projects mentioned below, the reader is entitled to be concerned about how many sub-problems we anticipate having to solve in order to achieve new levels of intelligence in automata! Is it tens? hundreds? millions? Are we merely scratching the surface? One of the purposes of unified milestone demonstrations (such as the one under discussion) is to show that the numbers can be contained. Speaking informally, we believe that the number of subsystems of intelligence is large, but not as large as the number of subsystems engaged in, say, an Apollo mission. The topics that have already been considered in the Laboratory constitute a significant fraction of what is necessary.

1.2 Other Milestones: VISION and the BLOCKS WORLD

We will describe the other milestones more briefly, though they are no less serious. Some of them are easier and will be achieved earlier than the Language-Vision-Action demonstration in electronics; some are much more difficult.

The BLOCKS WORLD has played a central role as a culture medium for ideas about vision and will continue to do so despite the new concentration on other more useful and complex areas of work. To place new milestones in perspective we recall briefly some past history.

> The first important steps toward the present approach were the work, more than a decade ago, by H. Ernst on touch-sense controlled manipulation and by L. Roberts on machine scene-analysis of photographs of three-dimensional plane-surfaced objects.
>
> The next significant step, this one under ARPA support in the A.I. Lab (then part of Project MAC) was the development of visually-controlled tower-building programs in 1965-1966. The major sub-system was direct "natural" vision of real, but unoccluded and clearly illuminated blocks.
>
> In more recent milestone demonstrations, we have seen machine manipulation of several blocks in visually occluding relationships. Behind this practical step was a more important theoretical one, first taken by Guzman, of basing scene analysis on general knowledge about "bodies" or "objects" without using (as was done before) specific knowledge about particular kinds of bodies.

In this older work on vision, shadows and other side effects of lighting were treated as embarrassing complications to be minimized. However, these effects are sources of information, and we now understand them well enough to exploit them as such. A milestone for the very near future will be a Vision System to see highly shadowed and badly illuminated scenes, with the help of (rather than in spite of) the shadows. In the Language-Vision-Action demonstration, the relation between the hand and its shadow should be exploited to anticipate the contact of the hand and its target.

The MIT VISION SYSTEM is, at present, still an experimental prototype developed to explore new ideas both about machine vision and new styles of programming. It is the most powerful system available within its narrow domain of analysing monocular, polyhedral scenes. We are extending its capabilities to deal with more natural objects. Up to now, this system could deal only with polyhedral objects having uniform plane surfaces, such as blocks, wedges, pyramids, and the like.

Vision programs generally use information of just one kind: light intensities on a two dimensional projection. An important imminent advance is the use of multiple sources of information. Some of these will be other dimensions of vision—color, range, etc. Other sources will be symbolic, in particular those derived from interaction with SHRDLU. We do NOT see as a significant milestone merely using top level commands in English, passed on through SHRDLU, to operate a Vision System. This may be a good demonstration to the outside world of the flexibility and utility of SHRDLU, but it is really quite easy. The much more significant step will be to marry SHRDLU and VISION so as to permit statements made in English to give useful advice to the vision program: "Look more carefully in the shadow of the cube to the right of the tall block". A much more difficult step—sufficiently significant to count as a separate milestone if done in an insightful way—is telling the program in English how to extend its mini-world of competence. For example, one might tell a program designed to see and manipulate blocks about a good strategy for putting things in boxes.

1.3 Vision Outside the BLOCKS WORLD

Liquids: An excellent test-bed for ideas about changing shapes. The milestone is an eye-hand system that will make a cup of regular instant coffee. This is described below in more detail.

People-Watching: Our own research on the control and acquisition of motor (and postural) skills in humans will benefit from a program capable of observing a person engaged in a task such as learning to walk a tight-rope. A simple application of this is detecting when a person approaches dangerously close to the edge of a platform, and warning him.

Interpretation of Drawings: The interpretation, by machine, of not-quite realistic or conventional drawings poses problems intermediate between vision and language. It is a task that has not yet been performed significantly by machine, yet seems ripe as a future milestone. We hope, through work like that of Goldstein (see below), to develop a program that can describe in words—that is, by understanding—what is shown in a drawing, cartoon, or action-sketch with stick-figures.

1.4 Understanding English

Our Laboratory has pursued over the years a working hypothesis that can be stated briefly as: before one can get a machine to understand English, one must find how to make it understand at all. Translated into concrete terms, this means making programs to understand complex English statements about a simple mini-world which the computer is able to understand very thoroughly by drawing on a stack of special knowledge. An early milestone whose achievement encouraged this point of view was the early program, STUDENT, by Dan Bobrow. The latest clear break-through is SHRDLU by Terry Winograd.

A secondary milestone is the operational use of SHRDLU as a "front-end" to communicate with programs written for quite different, practical purposes. An early example of this kind of use is the coupling of SHRDLU to a question answering program made by C.C.A. We know of several other such projects. Though it is encouraging to see Artificial Intelligence programs being used more and more extensively, these uses of SHRDLU do not go beyond its original formal capacity. More substantial milestones which we see as significant and yet not far off include:

Using More Linguistic Information: A program that makes significant use of tenses, modalities (e.g. possible-probable) and indirect reference (e.g. He thinks that . . .) would be a considerable advance.

Getting Away With Less Linguistic Information: We have in mind using "common sense" interpretation to fill in only partially specified information. For more precision, see "Narrative", below.

Extending SHRDLU in English: A most important milestone will be passed when we are able to describe, in English, extensions to SHRDLU itself. SHRDLU's interpretational power would then act in a bootstrapping fashion. This goal is more inclusive and more difficult than the others we have listed.

1.5 Computers, Knowledge, and Intelligence

The following sections describe what we propose to do in much greater detail. Before entering this forest of detail, we want to explain our image of how this approach relates to others.

The world of Science is well-organized for the management of some kinds of knowledge. Its theories provide a firm (though not, of course, infallible) understanding of what kinds of knowledge are relevant to such tasks as predicting astronomical events or designing bridges. There are institutionalized repositories of such knowledge (handbooks, tables, encyclopedias) as well as the means of transmitting it to the next generation (schools).

Not all knowledge has been treated in this formal manner. In particular, a very large body of knowledge has traditionally been neglected for the simple reason that "everyone knows it anyway". This is the kind of common sense knowledge that leads one, for example, to rearrange the contents of a box (or throw something out) when more space is needed.

Research in Artificial Intelligence is forced to deal with this kind of knowledge quite explicitly. One might argue that the "intelligent computer" ought to acquire such knowledge by the tacit, informal process that leads humans to have it without explicit formalization. This is certainly true in some sense; but even to understand what is being asserted we need to formalize more explicitly, and characterize more insightfully, the kinds of knowledge in question and the kinds of processes that might lead to its acquisition.

Among our (human) ways to acquire knowledge, two stand out beyond others and our work has centered on them: Language and Vision. The rest of this proposal is divided into three parts, accordingly.

> Part II is concerned with issues related to logic, general knowledge, and common sense, in a context centered mainly around problems of understanding natural language. This section also includes closely related work on understanding procedures, programming languages, and descriptions of programs.
>
> Part III discusses issues connected with robotics, machine vision, manipulation, and other activities that involve physical-world interactions.
>
> Part IV discusses work needed to develop tools: hardware, programming languages, software, and other things needed to support the goal-directed projects.

These areas are, of course, not really separable; all of them, for example, have sub-areas that work with knowledge about the BLOCKS WORLD, a simple model physical world of simple, easily described objects and interrelations, that serves as well as the subject for logical, linguistic, robotic, and program-theoretical studies.

er of projects in the Laboratory are centered around the problems of understanding natural language. There studies include both theoretical and practical problems:

2.1 Export Version of Winograd's Language-Understander
——————————Terry Winograd and group

2.2 More Powerful Problem-Solving in SHRDLU--Using the CONNIVER Language Instead of the Primitive Micro Planner System
——————————Winograd and group
W. Martin and G. Brown of MAC

2.3 New Models for Meanings
——————————Minsky, Miller, Charniak, and others

2.4 New Grammar Organizations
——————————D. McDonald, A. Rubin, V. Pratt

2.5 Theories of Generic Expressions and Typical Situations; Scenarios, English, Quantifiers, and Logic
——————————R. Moore, M. Marcus

2.6 Learning to Understand
——————————D. McDermott

2.7 Understanding Stories
——————————E. Charniak

2.8 Sussman's Program: HACKER
——————————G. Sussman

2.9 Knowledge about Procedures and Their Likely Bugs
——————————G. Sussman

2.10 Goldstein's 'Solution' of the Halting Problem
——————————I. Goldstein

2.11 Goldstein's Program-Understanding Program
——————————I. Goldstein

2.1 Polishing the Language-Understanding Program

A project that will be completed this year is the conversion of Winograd's language-understanding program (called SHRDLU) from a first demonstration prototype into a generally accessible, well-documented experimental facility for further development of linguistic models and applications that are related to the basic theory behind this model. The new system contains extensive tracing and debugging aids, and a detailed manual is being prepared. This will make the system accessible to users who begin with very little prior knowledge about its details. The system is available over the ARPA network. An early version of the manual was prepared last summer with the help of Stuart Card at Carnegie-Mellon, and we are now adapting it to provide a full guide to the system. Current sections include:

Brief Description of SHRDLU
Implementation and Version Information
The Distribution Package
Instructions for Running SHRDLU
Some Facts about the Program
References for More Information
Examples of Using the Features of the System

The system has many features for making it easy to develop and debug extensions, and specializations for particular applications. These include a self-explanatory command tree system that allows users to view various parts of the system in interactive modes.

2.2 More Powerful Problem-Solving in SHRDLU

Several students are rewriting parts of the language understanding system to use CONNIVER instead of Micro-Planner. The new language, which permits cross-reference between different parts of a problem solving process, will provide a better base for extending the language program to a larger world that can use hypothetical contexts, plausible reasoning, and more complete handling of tenses.

We are also working with other groups in their uses of this language understanding system. The project of W. Martin, at Project MAC, will probably use it as a means of communication with their automatic programming systems, and G. Brown, also at MAC, is exploring the possibilities of basing a language translator on SHRDLU; at present she is working on a very small specific set of problems involving German noun cases and prepositions, but the basic ideas may be generalizable. In any case, by writing a comparable grammar for German, she is increasing our understanding of which basic mechanisms are best for representing natural language grammars in general.

2.3 New Models for Meaning

At a theoretical level, several of the staff and students are exploring some new formalisms for representing meanings. One problem is to tie together a variety of phenomena that have been addressed more or less separately by such models as Fillmore's Case Grammars, Abelson's "molecules, texts, scripts, etc.", Schank's Conceptual Grammars, Halliday's Systemic Grammars, etc. We believe that these can be tied together in a more consistent way by viewing language understanding as an active procedure, constantly engaged in an operation of "fitting together" inputs and the implications of inputs. These theories are just beginning, but we hope they will lead to a second generation system much better equipped to handle plausible reasoning, incomplete inputs and a wider range of ways in which language conveys meaning. It is too early to describe the theories in detail, because they do not yet have adequate formalisms, but among their new elements are what might be called "scenarios"— structural elements that describe "the ways things usually happen"— that are substantially larger than the kinds of elements found in earlier syntactic or semantic theories.

2.4 New Grammar Organizations

David McDonald is working on problems of generating sentences that represent meanings in ways that are responsive to the needs of the system's user. This means that the system must use its knowledge about what (it thinks) the user believes. Winograd's original SHRDLU system does this to a certain extent; it takes into account some of what the user can be deduced to know when it answers questions. McDonald's approach to generating coherent discourse views it as concerning multiple procedures operating on a common data structure. First, there is a logical process concerned with choosing words and producing structures which convey the underlying meaning. Second, there is a discourse procedure whose job is to structure the output for coherence, making the necessary inter-sentence connections, introducing pronouns when appropriate (and when their referents can be determined, presumably). Third, there is a syntactic specialist to make sure the output is in a proper grammatical form. McDonald wants to avoid the kind of inflexible organization in which each of these is forced to work on a completed output of another. Thus, he wants a heterarchical system (see below, in connection with Vision and Robotics) in which they time-share, each being allowed to make suggestions as to how the output should be formulated, and reacting to suggestions of the others. The goal is to produce an output system for use with the SHRDLU program with capability complementary to that system's input understanding ability.

A. Rubin is working on a new version of Wincgrad's grammar, exploring different types of organization which get away from the highly linear organization of the current grammar. More efficient and flexible grammars may take more syntactic cues from the words themselves, resulting in a still primarily top-down parser whose operations are primarily initiated by bottom-up methods. Words such as "a" and "the", or propositions and question words have immediate syntactic implications and should be able to direct the parsing process. Reliance on the words themselves instead of their position in the utterance should be a great help in parsing incomplete and ungrammatical utterances, especially when coupled with a coherent discourse content. In fact, such exploitation may be simply necessary for dealing with real-life interactions, many of which are fragmentary yet perfectly intelligible. Rubin is preparing flowcharts so that the resulting grammar can be used both for constructing programs and for describing and studying English. People with limited knowledge about PROGRAMMAR and LISP (the computer languages used by the system) but who have substantial knowledge about linguistics will be able to use these charts to get access to a computer-based grammar for a substantial part of English.

We hope this will accelerate interdisciplinary communication and progress, and this seems likely because there is already world-wide interest in this system. The flowcharts should make it possible for people unfamiliar with details of the program to experiment, modify, and extend the grammar for their own applications.

Rubin is also studying problems of extending the system to use complex tense and time semantics; perfect and progressive tenses, futures, and modals. Tense and other time components combine in not-well-understood ways; compare: "I see my advisor Thursday" with "I see my advisor Thursdays"; some deduction is required to decide which sense of the tense is indicated. Compare "The car runs on gas", which is essentially tenseless, with "the car needs to be washed". To deal with these will require changes in both semantic and deductive parts of the system, using new ways to represent past, future, and hypothetical assertions.

2.5 Theories of Typical Situations and Generic Expressions

R. Moore has been exploring some of the problems in connecting English expressions to their underlying logical forms. He is concentrating particularly on English quantifiers and their connections with mathematical logic. The connections are not straightforward, and one does not often "mean" a quantifier in the same sense of the conventionally equivalent logical symbol. A word like "any" can have several different meanings in logical terms. In the sentence "Did anyone come to see me?" it corresponds to the logical "there exists" and we can represent it as (Exists X)(X came to see me). But in the sentence "Intercept anyone who comes to see me." it represents a

universal: (Forall X)(If X comes to see me, intercept X). Moore is working on a formalism that combines predicate logic with a sort of lambda calculus which can deal with this sort of phenomeron, as well as more complex problems involving embedding and pronoun reference.

M. Marcus is also studying questions that seem closely related. What are "generic" nouns and what is their role in commonsense reasoning? When one says "A bird can fly", one logical interpretation is that this means "If X is a bird, then X can fly". We don't believe that this traditional interpretation is correct enough to be usable. Some birds, like the ostrich, can't fly in any case; other birds can't fly now because they have broken wings, are tethered, etc., etc. So one usually means something like "typical birds fly" or "If X is a typical bird, X can fly". What does this mean, in turn?

One direction many workers have explored is that of formalization within other logical schemes. We believe that this is not entirely a matter of formalization, however! Perhaps there is a much deeper issue here, one of content. The understanding of what is "typical" depends on one's knowledge and familiarity about the particular subject matter. We think that useful solutions to understanding such statements are probably to be found in a system that uses "scenarios" of the kinds of things that "usually" happen, and that, hence, the interpretation of statements

about "the average man" and the like are to be based on detailed "general" knowledge about the subject rather than on the discovery of a new logical quantifier or rule of inference.

2.6 Learning to Understand

D. McDermott is developing a broad, but not deep, program called TOPLE, which tries to understand simple declarative statements about a world like that of Winograd (1971). TOPLE can "visualize" certain spatial relations, guess causal explanations of simple behavior, and make predictions about the future course of a sequence of events. It has a limited ability to understand the actions of other creatures, and make very simple models of their states of mind. This research is addressed to the problem of developing programs that can be told new things instead of having to have them painstakingly programmed in. It has often been suggested that this could be done using a set of programs for translating declarative statements into deductive assertions and theorems. There are several syntactic and semantic reasons why this is not sufficient:

1. The program will inevitably be seeing new words and phrases as we talk to it. It must have sophisticated ways of translating these into formal language. It is probably possible to guess the syntactic category of a new word, or even phrase, from its context (Thorne, Bratley, and Dewar, 1968), but that is only the beginning of the problem. If "pen" is a relatively new word (whose meaning, however, has been told to the program), it must be clever enough to avoid translating
"The man shot the pig in the pen."
the same way it does
"The man shot the pig in the heart."

2. Even when completely stupid interpretations are discarded, many sentences contain syntactic and semantic ambiguities that cannot be resolved without guessing which interpretation is most plausible. For example, without taking context into account, it is impossible to say to whom "him" refers in:

"Her father hated Tom, but she loved him aryway."

3. Human language is abbreviated. People very often leave out steps in arguments and stories, explanations for events they recount, and qualifications of sweeping statements, when they think these can be reconstructed by their listeners. In human communication the speaker usually sketches the situation he is describing with as few strokes as possible. It is up to the listener to fill in the most plausible details consistent with what he says. If he says,

"The club grew. Soon they needed a bigger clubhouse."

no human would fail to see the causal progression the speaker intends. This reconstruction must be applied to almost everything the machine hears, since an English speaker often leaves unsaid information regarding time, causality, and situation, all these being implicit in what has gone before and what he is most likely to mean.

4. Human beings are not infallible. They can be inconsistent, misinformed, or unclear. Two people may disagree, and some people even tell lies. In any such case—when imagining the situation being discussed is impossible or very costly—the machine must refuse to believe in it.

McDermott's work on problems 2 and 3 have influenced the development of the CONNIVER programming language, and the development of TOPLE, which is written in CONNIVER. Parts of the system are already programmed.

So far, the system cannot absorb new CONNIVER programs, but consists mostly of a set of carefully tuned programs which take responsibility for adding new statements to the data base as well as retrieving conclusions from what it already knows. These programs constitute a belief system (or, in Abelson's (1973) terminology, a knowledge system), which is committed to making as much sense as possible from things it hears. The programs (called "if-needed methods") which embody its knowledge of the world, possess some knowledge about what things are easiest to believe, what contradictions are best to work on, etc. TOPLE does bookkeeping, but lets these experts do most of the analysis of statements it hears. (It does not handle natural language directly, but accepts statements in a predicate calculus-like format that is directly assimilable by the pattern matcher that calls if-needed methods.) These methods ponder questions about the possibility and plausibility of the things they hear; communicate their difficulties back to their callers, which can give advice or try something else; and are free to construct

hypothetical worlds to test theories about what the world really looks like, or what a speaker really meant, given what they are told.

Thus, for example, the routine which accepts statements about physical locations attempts to find the most likely arrangement of objects which is consistent with the locations given by a speaker. The routine which handles statements about going places recognizes the vagueness with which people specify destinations, and attempts to figure out the most likely destination that the creature it is told about could be aiming at. (For example, "He went over to the chandelier." usually means, "He went over to the part of the floor directly under the chandelier.")

Basically, TOPLE will be a very skeptical program, which tries to resist changes to the data base, choosing the least jarring changes when they are enforced by its being spoken to. Since its programs are doing plausible reasoning instead of airtight deduction, it is essential that they record their reasons for the beliefs they add to the world model, and that these reasons be accessible to any program that later comes into conflict with them. These reasons are stored in a very general fashion as programs which haggle with the newcomer about what the world is really like. The current version of TOPLE, when finished, will be a substantial extension in our understanding of problem-solving with incomplete data by "commonsense" reasoning. If all goes well, a later version of this program should have linguistic knowledge and could be attached to a system like Winograd's SHRDLU parser and semantic routines. There is a broader problem here that should be attacked: if a routine is to understand people's speech fully, it should understand how language is used as a tool, what people are likely to want to accomplish when they say something, as well as what a sentence would mean taken completely out of context. Until this sort of knowledge is formalized, machines will be deficient in language understanding and in fully understanding a person's motives and intentions from what he says, and hence will miss much of happens around them, since most of what goes on in stories and in real life is dialogue.

2.7 Understanding Stories

E. Charniak, whose thesis on representation of knowledge in children's stories was completed last year, continues to work on problems of text-comprehension. Charniak believes that the best way to construct a

theory of knowledge is by exhaustive analysis of particular story fragments in an attempt to pin down exactly what facts are needed in order to understand the story, and to see how these facts ought to be accessed. Charniak begins with the model presented in his thesis, in which a variety of "demons" or semi-autonomous programs control inference and information retrieval when "set" or "activated" by characteristic words or events in the story.

When one studies knowledge at this degree of fine detail, one encounters a great many problems. Because there is not much systematic theory yet, we will illustrate the situation by example.

In Charniak's thesis occurs the following example to illustrate some problems in determining noun phrase reference:

(1) Today was Jack's birthday. Penny and Janet went to the store. They were going to get presents. Janet decided to get a top. "Don't do that" said Penny. "Jack has a top. He will make you take it back."

The problem is to recognize that the "it" in the last sentence does not refer to the top Jack currently owns, but rather to the one Janet is thinking of getting. In the thesis, it was suggested that a rule of the following form might be at work.

(2) If we see that a person P might not like a present X, then look for X being returned to the store where it was bought. If we see this happening, or even being suggested, assert that the reason why is that P does not like X.

Charniak continues:

> "While the exact form of the rule was not crucial to the argument in the thesis, I have given some further thought to the problem of exactly what information is at work in (1). My first conclusion is that (2) is not the piece of information which is used in (1). Consider:
>
> (3) Today was Jack's birthday. Penny and Janet went to the store. They were going to get presents. Janet decided to get a top. When she told Penny this, Penny said, "Jack will make you take it back."

In this story, prior to Penny's statement we had no reason to believe that Jack did not want a top. Of course, in this story there is no problem with pronouns, yet, we understand Penny's statement as implying that Jack does not want a top as a present, and that is the reason why the top will be taken back. Hence, a minimum modification of (2) would be (the change is capitalized):

> (4) If we see that a person P MIGHT GET a present X, then look for X being returned to the store where it was bought. If we see this happening, or even being suggested, assert that the reason why is that P does not like X.
>
> "Proceeding in this way we can construct many variations on (1), all of which utilize the fact that (2) represents. However in each case we can construct the story so that (2), as written, does not apply. The goal of all this is to find a "correct" version of (2) which will account for all of the examples."

Charniak is also working on a very different problem-solving domain; problems in electrostatic field situations. In doing this, he is returning to attempt to apply new problem-solving methods to applied physics and analysis, the subject of his Master's thesis in which he attempted to extend the work of Bobrow to more realistic physical problems and found that the problems of ordinary commonsense reasoning posed critical obstacles.

2.8 Sussman's Program: HACKER

Consider the piece of advice:

> "If the goal is to place Block A on Block B, first check to see whether there is erough space on B; if there is not, rearrange the blocks already on B by strategy TIDYUP and try again."

Knowledge of the sort carried by this advice is successfully used by A.I. programs working in a very simple world of blocks. However, the advice is needlessly particular. The strategy suggested is obviously not special to blocks; one certainly ought to be able to formulate it as a statement about the allocation of space in more general situations.

But even this is too particular, if the word "space" is taken in a literal sense. Consider how one uses the spatial metaphor in talking about computer memories. This reflects the general usefulness, in thinking, about the abstract organization of memory, of "spatial strategies". That is to say, that information in memories can often be managed by using the same "tricks" that work for handling objects in real space! For example, rearranging (compactifying) non-contiguous blocks of data can create space for a new block of data.

Gerald Sussman has translated these very general remarks into a quite new paradigm for a program that learns from experience. Sussman gives his programs access to two different kinds of knowledge:

> Knowledge that really is specific to the particular domain (for example, a specific set of colored wooden blocks).
>
> Knowledge about strategies (for example, for allocation of resources, or for the ordering of sub-tasks) stated in a form much more general than that in which they will be used.

Sussman's ingenious idea is to represent the general knowledge in a form that can be expanded into actual domain-specific computer code when needed. This new code shows as an addition (patch) to a program that had run into an error condition (showing the presence of a bug, and thus the need for a patch). The addition of the new code can be thought of as automatic debugging of the old program!

> Consider a concrete example that Sussman has used to demonstrate the potentialities of his idea. He begins with a program written to place block A on another block, B. This program works perfectly as long as B is clear of any obstruction. Now suppose that the program is asked to place A on B when there is already a block C on B.
>
> In the usual context the program would fail, perhaps with an error comment. In Sussman's context the error comment triggers the generation of an extra piece of code which causes B to be cleared before A is placed on it. The program is now expanded into a more complex form and will stay that way until it once more falls into an error condition revealing a new bug and leading to another patch.

Sussman's demonstration makes it very plausible that quite complex programs could be grown in this way by successive macro-expansion of a small number of general strategies. Sussman will pursue this goal as the topic for a Ph.D. thesis which will be completed during the coming year.

2.9 Knowledge About Procedures and Their Likely Bugs

To appreciate the thinking behind HACKER one must realize how misleading it is to think that Sussman merely replaced the particular by the more general. Any particular statement can be generalized in many directions! Which should be chosen? A working hypothesis of much work in A.I. is that the most powerful generalization in areas such as these tend to take the form of properties of procedural structures. The meaning of this will be filled out by examples from on-going projects. The first is taken from HACKER itself.

Suppose that the program we referred to before is presented with a scene containing a blue block, a green block and a pyramid. The pyramid happens to be on the green block. The program is asked to put the blue block on the green block. It observes that the top of the green block is filled; it knows that the remedy must be to remove the pyramid, which it does, but puts it on the blue block. On the assumption that only single objects can be lifted, this is a disastrous act; it clears the green block but makes it impossible to move the blue.

Now the question is: what advice could save the program? On the more specific level one might try:

> if you want to put block A on block B, avoid putting anything on block A.

Expressed (as it easily could be) in an appropriate programming language, this would save the situation—but we'd prefer something more general. What?

One possible direction of choice might be to generalize "block" and "on" to "objects" and "relations" satisfying such and such axioms. Certain contemporary approaches to A.I. would go in this direction despite the difficulties encountered in working with appropriate axioms. HACKER actually takes a very different direction of generalization. According to Sussman, the problem is not about objects and relations but about programs—and particularly about a kind of bug that typically arises from interactions between attempts to satisfy a sub-goal (clear B) and the conditions (A being clear) still holding at the time the super-goal (move A to B) was set up. To formulate the advice properly, he makes a very general powerful addition to the programming language: an operation called "protection" whose effect is to inhibit changes in aspects of a situation to which it is applied. Using this operation, the advice about not putting block B on block A turns into something of the form:

> Protect all known necessary conditions for the super-goal before examining a sub-goal.

This transformation of advice about blocks into advice about program structure (goal—sub-goal interactions) illustrates an expanding theme of research in the Laboratory—of which HACKER is a particularly clear, but not unique, example. To see the common element clearly with the

next example, recall that the protection advice derives from reorganizing the likelihood of a certain form of interaction bug.

2.10 Goldstein's "Solution" of the Halting Problem

Suppose you have written a program, P, which is intended to make a certain decision and then stop. You suspect that a bug causes it to fall into a never-ending loop, i.e. not to halt. It would be very valuable to have a program called, let us say, SUPERMONITOR, which would examine program P and tell whether or not it will halt.

Everyone who has taken a course in automata knows that an infallible version of SUPERMONITOR cannot exist—the halting problem is unsolvable! Given any alleged version of SUPERMONITOR, it will be possible to find a program P whose failure to halt is too tricky for SUPERMONITOR to catch. But while programs can, in principle, halt or fail to halt for exceedingly complex reasons, in real life many (possibly "most" or "almost all") accidental causes of failures to halt are very simple indeed. Thus the undecidability theorem in no way precludes the possibility of a practically useful (though fallible) version of SUPERMONITOR.

Ira Goldstein, also a graduate student in the A.I. Lab, has chosen as his thesis topic the design of a monitor which could be thought of as looking over the shoulder of a novice programmer and making comments about why and when his programs work or do not work. One of the functions it will perform is detecting inadvertent infinite loops.

To appreciate the wide applicability of such work, notice that the human novice programmer could himself be a computer program generating programs. Thus, besides providing a clear paradigm of the concept of "teaching machine" and new insights into the nature of programming

knowledge, Goldstein's work is directly relevant to all areas in which programs are automatically generated.

Like Sussman, Goldstein provides his program with knowledge about the programs people (and machines!) actually write and the troubles they might get into. These projects are complementary in concentrating on disjoint aspects. The following example illustrates the aspect Goldstein studies in this part of his work; in the next section he discusses a very different aspect of programming knowledge.

Consider a simple exercise program that might be written by a beginner. (The program is in LOGO.)

```
TO COUNTDOWN :NUMBER
10 PRINT :NUMBER
20 COUNTDOWN :NUMBER-1
END
```

The effect of the command COUNTDOWN will be to print
3, 2, 1, 0, -1, -2, . . . and so on in a loop which would never end but for accidental limitations of the computer used to implement it. The non-halting of this program is easily detected by a system that knows that "recursive programs without stop rules don't stop". A more interesting situation is illustrated by the following sceraric. The program generator (human or mechanical) adds the line:

```
5 IF :NUMBER=0 STOP
```

The program now counts down to 1 and halts. The programmer then changes the program to make it countdown by twos:

```
20 COUNTDOWN :NUMBER-2.
```

ard finds himself in a simple form of a very typical bug situation. The detection of the non-halting situation is more complex in this case, but within the state of appropriate arts.

> The steps in one approach are (1) recognize from the structure of the program that it will always halt if and only if the equation K-2X=0 has a positive integral solution for X for all positive integers K, and (2) show that K-2X=0 does not have a solution in integers when K is odd.

We gain confidence in the feasibility of the project by noting that these steps lie in areas (translation of formalism and symbol manipulation) that have shown firm progress.

> From a certain practical point of view this is enough. But our theoretical orientation leads to a dissatisfaction with an "unnatural", "mathematician's" formulation of the simple fact that counting down by twos can't get you from seven to zero.
>
> A very different approach is illustrated by elevating the concept of "conservation" to the status of a fundamental primitive about which the system ought to know. The bug under discussion could then be recognized, as by a person, through the qualitative knowledge that subtracting two conserves even-ness—rather than through the manipulation of algebraic equations!

2.11 Goldstein's Program-Understanding Program

We have mentioned one aspect of Ira Goldstein's thesis goal: to make a program that will understand an object program sufficiently to decide whether it will halt and if not, why not. Besides this generalized kind of understanding, Goldstein's program will understand the intentions of programs written for a particular subject domain or mini-world. This is in the same spirit as Winograd's program, which is able to understand

complex English sentences, provided that these are about a mini-world (the BLOCKS WORLD) of which the program has thorough knowledge. Goldstein's mini-world consists of two-dimensional line-drawings, such as stick-men. His immediate problem is developing an appropriate description language for this mini-world. This work draws on ideas from work on vision and is expected to contribute to it, more particularly as the robotics projects move towards more natural vision.

Goldstein's program-understanding-program will deal with very simple graphics programs (the "object program") written in LOGO, with the intention of drawing a scene such as "men walking in file". The test of the program will be its ability to diagnose semantic bugs in the object program. The kind of knowledge required to do this could be used in many application areas including: making a more intelligent teaching machine (to teach programming); automatic debugging in Sussman's style; automatic program writing.

Part III. COMPUTER-CONTROLLED VISION AND MANIPULATION

Research on computer vision and its applications is entering a new phase in the Laboratory. In the past, the work centered around a variety of prototype problems and attempts to develop sound theoretical models of those problems. We now feel that we have a very firm understanding of these problems at all levels and want to use this knowledge to move into broader and more practical applications.

In the past, vision research in the Laboratory focussed on the "BLOCKS WORLD". This provided configurations of three-dimensional blocks, wedges, and other polyhedra simple enough to provide access to basic problems yet complicated enough to ensure that the problem solutions will have general relevance.

Even within the BLOCKS WORLD, it was necessary to work out new schemes of program organization, as well as new techniques for digital picture-processing, to be able to analyse scenes in which clearly delineated geometric objects occlude one another in three-dimensional space. The world, however, is not composed of uniformly illuminated geometric objects, and current research is pointed at the problems of natural environments.

Our applications research is being done in conjunction with work on the mini-robot development project furded by a supplementary proposal. Work on natural environment vision and manipulation is to include efforts on the following problems; all the results are to be available within our modular "heterarchical vision system" described in prior proposals and currently working at a demonstration level.

3.1 Heterarchical Vision System and Project Direction
——————————————P. Winston, B.K.P. Horn

3.2 Generalization of Labelling Theories
——————————————D. Waltz

3.3 Grouping, and Tactile Scene-Analysis
——————————————T. Finin

3.4 Analysis of Curved-Line Drawings
——————————————M. Adler

3.5 Color Vision
——————————————M. Lavin

3.6 Touch and Tactile Programming
——————————————D. Silver

3.7 "Low Level" Vision Programs
——————————————J. Lerman

3.8 Physical Knowledge and Stability
——————————————S. Fahlman

3.9 Liquids and Hand-Eye Tracking
——————————————R. Woodham

3.10 Electronic Assembly: Another Robot "World"
——————————————P. Winston and others

3.11 Analysing Complicated Objects
——————————————J. Hollerbach

3.12 Groups, Descriptions, and Conflicts
——————————————M. Dunleavy

3.1 The Heterarchical Vision System

Work on problems of natural vision situations is most closely supervised by Prof. P. Winston, whose personal concerns are with the theory of scene-analysis in the BLOCKS WORLD, automatic learning from experience of meaningful three-dimensional structures, the induction problem for meaningful groups of objects, and the problems of overall system organization that arise because of the variety of kinds of knowledge involved in "perception". B.K.P. Horn is responsible for supervising applications research and development of image processing systems.

3.2 Generalization of Labelling Theory

D. Waltz' recent thesis ties together much earlier work done at our Laboratory and others on theory of geometric scene-analysis. The sequence of events was basically this: early work on the role of "T-joints" and other local features pointed the way to programs that could solve the "figure-ground" problem of separating visual objects. This work reached a relatively successful but very complicated plateau with the work of Guzman, who used a variety of formal heuristics for linking, grouping, and link-inhibiting cues.

The behavior of Guzman's program was better "explained" with the introduction of formal semantics for the linking heuristics by Huffman and Clowes. In that model, it was recognized explicitly that different kinds of edges arise from different classes of three-dimensional situations. Applying these ideas, and using a new analysis of the situations in which Guzman's program encountered difficulties, Rattner

and others developed procedures that were simpler, more meaningful, and more effective than the older, more "syntactic" versions. In particular, the elaboration of complicated "linking" heuristics were replaced by a much simpler system of alternate linking and "splitting" of objects apart.

In his thesis, Waltz carried further the analysis of the semantics of edges, and showed conclusively that in the BLOCKS WORLD there are many surprisingly strong constraints on the line drawings derived by projection from three dimensional scenes. Once these constraints are understood, one can often get directly at the meanings of each line in a drawing, and identify it as concave or convex edge, obscuring edge, crack, shadow, or other sort of line. The techniques and implications of this appear to have significance also in other semantic areas such as natural language and automatic programming.

Waltz is now working to add features to enable the program to use more global facts than were used in the thesis work, and is developing a graphic display program to facilitate study of the labelling process. He is supervising the work of Adler (see below). He hopes to find out how far his new labelling techniques can be adapted to work on curved objects, and poorly illuminated natural scenes. In particular, he is interested in the semantics of features of civilized interior scenes—office, home, hospital, school, etc., and relations between verbal and visual scene-descriptions.

3.3 Grouping and Tactile Scene-Analysis

T. Finin is completing work on recognizing objects in contexts that are complicated by such serious occlusions that not much of some object can be seen. The methods depend on assuming that regularities consistent with the portions of the scene in direct view are continued into the

unseen portions. Such regularities might be detected, for example, by the heuristics proposed in P. Winston's thesis. Finin hopes to carry this further into the system for hand-eye coordination, so that the conjectures about the unseen portions can be confirmed or rejected by tactile explorations. In the simplest version of such an application, the robot would be programmed to completely disassemble the scene, operating at all times on objects in direct view. Obviously, this should be regarded as a last resort; and there are cases in which any disassembly constrained to so operate would pass thorough a stage of instability in which the structure would collapse, damaging delicate parts. If the program's hypothetical reconstructed image of the three-dimensional situation is exploited, one should usually be able to settle ambiguities by a very few carefully chosen delicate tactile probes that need not disturb stability, if the latter is taken into account in the selection.

T. Lozano-Perez is directly concerned with extending such grouping heuristics. In the BLOCKS WORLD, Winston proposed grouping heuristics based on chains of similar relations, and groups connected by common relations to another object. Lozano-Perez is studying criteria for grouping hypotheses in more complicated specialized domains, including the influences of sizes, colors, textures, proximities and apparent functions.

3.4 Analysis of Curved-Line Drawings

Although a great deal is now known about analysing straight-line drawings that represent three-dimensional scenes, much less is known about drawings with curved lines. Some proposals have been made by Guzman and Huffman in earlier work. Current work, at Stanford, on scene-analysis of curved three-dimensional objects is based on surface and range-finding data. Mark Adler, in our project, is working on extensions of the Guzman and Huffman ideas, to analyse drawings of outlines of three dimensional objects that occlude one another in the viewer's image.

Several problems must be confronted immediately. To recognize curved objects we need adequate descriptive methods that are not too sensitive to position changes. One might begin by adapting Winston's feature and relation network descriptions to images with curved boundaries. Ambiguities are resolved by attempts to map local configurations onto models (as in an earlier proposal by Guzman that was never implemented). To do this effectively, however, the effects of occlusion of features by objects must be handled realistically, and the proposal is to use a sequence of analyses, beginning with use of T-joints to detect occlusions, followed by conservative assumptions about intact unoccluded regions to serve as starting units for the description matchings.

The procedure for matching models will probably be an advance in sophistication, in the sense that each model will contain knowledge embedded within itself about the effects of occlusion discoveries on what next to look for. Thus, if in searching for a match to a model of a man one finds that the sleeve of an arm is occluded, one would specify that finding a "hand" in the right place is acceptable. (In the previous, knowledge-free experiments of this sort, one would have had to find a reasonably close fit to a continuation of the sleeve on the other side of the occlusion.) Eventually, such predictions could be deduced from three-dimensional models, and we hope to combine this work with that at Stanford on constructing such models. Adler plans to develop his procedures using the world of cartoon drawing conventions, before facing the more complicated situations that arise in pictures of real objects.

3.5 Color Vision

M. Lavin, one of our new students, plans to investigate color theories such as those of Land, Lettvin, et al. in the context of the robot environment. This dimension of visual experience has received relatively little attention in robotics. Implementing a color specialist program may also provide powerful tools for investigating the current theories of human color vision which, at the present time,

remain surprisingly controversial. For the Robot itself, we hope to get a better and more useful understanding of how region shape, boundary sharpness, lighting changes, and other perturbations influence programs embodying such theories. And finally, we would like to see whether there is any practical or computational advantage in using a "physiological theory" instead of a naive spectrum-characterization. If the proximity of colored objects has substantial effects on nearby ones, which is quite likely the case, it may well turn out that one of these theories, in which color-descriptions tend to be relative to what happens on inter-region boundaries rather than depending completely on spectral content, might actually be of advantage to a robot.

3.6 Touch and Tactile Programming

D. Silver has recently applied our six-axis force sensing wrist to turn cranks and screws on nuts, and plans to continue to work on applications that are difficult or impossible to do using vision alone. Such work has long been deferred in our Laboratory, even though it figured in the earliest proposal, because the problems of obtaining usable vision in a natural environment were so difficult. The time is now ripe for this, and the results will be an important part of the facilities to be supplied to the mini-robot project's system.

3.7 "Low-Level Vision" Programs

Years of work have shown convincingly that no pass-oriented single-idea line or edge-finder can work in realistic environments. High-level driving programs must have available a variety of specialists for identifying sharp background lines, dim interior lines, shadows, cracks, reflections, highlights, etc. Detecting unexpected lines requires procedures that search out from known vertices. Bringing together a wide variety of previous work, J. Lerman has concentrated on organizing a supply of primitives for such activities into a system that is better integrated than any of its predecessors. It will bring the Waltz edge-semantic labelling ideas into the very first steps of vision processing, by the way of constraints on partially labelled, partially analyzed fragments of scenes. These constraints serve to restrict places and types of lines to be looked for next. Lerman hopes to make his system able to avoid areas congested with many short segments and close-together potentially ambiguous vertices; such areas which should be left until the system has had a chance to apply more global analysis of the three-dimensional situation to the long, clear lines and isolated sharp vertices that are more easily understood.

7.8 Physical Knowledge and Stability

S. Fahlman is completing a construction-planning procedure that couples a sophisticated understanding of gravity and support principles with heuristics for coping with complicated construction tasks. Depending on the situation, his system may elect to build a structure one piece at a time, by prefabricating substructures, or by indirection— using scaffolds or counterweights to provide temporary support. His thesis will describe a "specialist" that contains advanced knowledge about such matters and the ability to propose plans for using such knowledge.

7.9 Liquids

R. Woodham is investigating little-explored problems in visual tracking, and is building up a new experimental "world" that may turn out to be ideally suited for our new steps toward practical applications. In his system the robot will pick up a container and pour a liquid (coffee) into a cup. A liquid-pouring "specialist" will track the dark edge of the liquid as it moves in space and rises relative to the supposed edge of the cup. The following is a brief excerpt from his proposal, describing the coffee-table environment:

A Problem Domain For Studying Hand-Eye Coordination

> On a table, there are several coffee cups, a coffee pot, a bowl containing sugar cubes, a small pitcher of cream and a spoon or other object suitable for stirring. There is no particular arrangement to the objects on the table. They are randomly placed within the field of view of a vidissector eye and within the reach of a mechanical arm-hand.

A human engages in a short dialogue requesting a cup of coffee in any one of its standard configurations (ie. black, cream, cream & sugar, sugar only, double cream, etc.). The arm-hand proceeds to select a cup, pour the coffee from the pot, add the required embellishments and stir the result. The human picks up his cup of coffee and says, "Thank you!"

Features Of Such A System

1) We would be demonstrating a generalized flexibility. Since there would be no specified arrangement of objects on the table nor a fixed recipe for coffee, the robot would have to both visually locate the objects and construct a plan as required. Further, we would like the system to be general enough to allow for the addition/deletion of cups while the operation is in progress.

2) We would be exhibiting a true hand-eye system in an environment realistically approaching that of the real world. In particular, the operation of pouring must accomodate a real world that changes dynamically, not just in discrete steps. Visual feedback, with the arm-hand in the visual field, would be an essential prerequisite to accomplish accurate pouring.

3) We would be exhibiting a somewhat generalized manipulative capability through the use of simple tools—a pot for pouring and a spoon for stirring.

4) We would be facing the issue of quality control. Visual feedback must certainly be used to monitor pouring. In addition, feedback must be used to protect against pouring into a cup that's fallen over or pouring into a cup that's already full. Similarly, feedback must also be used to keep from knocking over a cup when stirring its contents.

Is This A Good "Toy" System?

The idea of a robot coffee maker probably strikes one at first as being a good demonstration. It certainly would be that. However, in considering possible alternative problem domains for a hand-eye system, I believe that the robot coffee maker is also the most appropriate.

The coffee maker environment is rich enough to support the thorough investigation and development of the various kinds of feedback tools and capabilities that would be required in any hand-eye system. The processes involved in making a cup of coffee are quite characteristic of the kinds of processes required in a generalized hand-eye system.

The primitives required to monitor the rising level of coffee in a cup are seen as essentially equivalent to those that

would be required to carefully align the edges of objects in a complex assembly procedure. The primitives required to stir the contents of a cup with a spoon are essentially equivalent to those that would be required to tighten a nut with a wrench or turn a screw with a screwdriver. Similarly, the primitives required to locate a cup for pouring are essentially equivalent to those that would be required to locate a hole for inserting a bolt or screw.

Of equal significance, however, is the fact that the coffee maker environment is also simple enough to support such an investigation with a minimum amount of time required to deal with outside issues. I believe the current vision system can easily be modified to handle the specific objects required for the coffee maker. In any event, I can immediately begin developing techniques for visual feedback by restricting myself, for the time being, to polyhedral cups and pots.

The coffee making system involves an environment that is sufficiently dynamic so as to require a degree of interaction that would constitute a significant advance over previous work in machine hand-eye coordination. The primitives developed for the coffee maker would be applicable to a host of other hand-eye tasks. At the same time, the coffee maker represents a problem domain that is very accessible and manageable given the current status of the MIT VISION SYSTEM.

3.10 Electronic Assembly: Another Robot "World"

In our proposal for extension of our mini-robot development project, we describe a possible application to electronic assembly, maintenance and repair. The objects are electronic components. Here again, is an environment in which the visual problems are more complex than those in previous toy problems, but are believed to be within reach.

3.11 Analysing Complicated Objects

J. Hollerbach is making good progress in carrying us out of the domain of very simple geometric objects. In that world, identification and manipulation problems are often "too easy" because blocks and wedges are completely defined and located once a few location and dimension parameters are known. Hollerbach has proposed to develop descriptive methods for complicated polyhedra, with indentations and protrusions, in which the descriptions are structured in levels of detail, making location and identification reasonably convenient. At the same time, he is studying "simple" objects like bottles and cups with a view toward relating their descriptions to those of complex approximating polyhedra. Different approximations that are very similar in quality of dimensional fit can be vastly different in descriptional complexity, and the goal is to find techniques for describing complex objects in natural ways that make it easy to relate them to such prominent features as their global outlines and their decompositions into meaningful parts (such as "top", "handle", "legs", etc.).

3.12 Groups, Descriptions, and Conflicts

M. Dunleavy has proposed a project concerned with developing hierarchical models of how such complex structures as walls, windows, doors, and chimneys can be combined to form buildings. There are problems here that do not immediately meet the eye; the concept of "brick wall" must be defined in a somewhat indeterminate way as a repeated ("group-like") structure with, usually, some boundary-termination condition. The box or house concept of four walls meeting to form a box-like enclosure is easy to describe at a high level, but one has to engage additional knowledge to construct one—because one has to make many decisions about what happens at the intersections, where the kinds of descriptions that seem reasonable for the separate components may come into conflict. One can build a right angle in a brick wall without cutting any bricks in two, and if the wall is two layers thick, one can even make windows without cutting. How is such knowledge to be represented? Dunleavy's plans include heuristics for laying out the global forms, figuring out where the conflicts occur, and then applying local methods to attempt to remove the conflicts.

Part IV. Development of Research Methods and Tools

4.0 Programming Languages for A.I.

We have long felt that a major impediment to the progress of Artificial Intelligence is the lack of an appropriate language in which programs can be expressed for the computer. Although the situation has improved, an inordinate effort is still necessary in order to translate even a well developed theoretical idea into an experiment in computer performance. And even when the experiment is finally carried out, most of the programming effort is ultimately wasted, because of the enormous difficulty in making modifications to test new ideas suggested by it. Our goal in this respect, is to develop computer languages in which the concepts and methods of A.I. can be expressed easily and translated by the computer into flexible, intelligible and modifiable programs.

This goal cannot be separated as a "service" project with any degree of independence from the mainstream of Artificial Intelligence Research. On the contrary, constructing such languages could be redescribed as developing a formalism and fundamental set of primitive concepts for A.I. Thus, it is very close to the theoretical core of research and, in our Laboratory, language development has traditionally been integral with the substantive projects aimed at particular performance goals. Nevertheless, from time to time, issues in language design become crystallized from the general research and acquire enough clarity and

momentum of their own to become separate research topics.

We are very much at such a moment in time. The past few years has seen a dramatic change in clarity and intensity of discussion about languages for Artificial Intelligence. This wave, we believe, began in our Laboratory with the publication of early versions of the language PLANNER (and the simplified dialect MICRO-PLANNER), and has spread through the A.I. community internationally. The effect of this wave was that, for the first time, the issues related to computer language for A.I. were formulated in common terms so that experiences in many projects and many places could be compared more directly. In the past, each research project tended to develop its own ad hoc extensions to general purpose languages such as LISP.

Another interaction—from a surprising quarter—is adding to the intellectual ferment about such issues. This is the interaction between the language problems for "serious A.I. programs" and the problems that arise from our attempts to develop computer languages for the use of children in our elementary education project. Experience and new theoretical insights make it increasingly clear that the languages we use, and plan to use, for children (collectively known as LOGO) need to develop towards greater and clearer expressivity for talking about fundamental heuristic ideas. But this is exactly what is needed also for machine intelligence! In the past the work with children was too tentative for such parallels to surface and give rise to actual

interaction. This has changed, and now leads to specific suggestions for common effort. The "language for children" is being re-examined in extensive discussions within the Lab and more widely. During the current year we expect them to give rise to the design and implementation of at least one new computer language which will try to be clear enough for a child to use (quite literally!) and powerful enough for research in Artificial Intelligence.

4.1 PLANNER Progress
——————————C. Hewitt and others

4.2 Parsing Strategy Research
——————————V. Pratt

4.3 Debugging Aids
——————————S. Markowitz

4.4 Heterarchy
——————————E. Freuder

4.5 Automatic Theorem Proving
——————————A. Brown, A. Nevins, J. Geiser

4.6 Relation to our Program of Research on Education and Cognitive Development
——————————S. Papert and others

4.1 PLANNER Progress

PLANNER-like languages have now become widely accepted as important tools for research in Artificial Intelligence. The following groups are currently using such a tool:

Stanford University A.I. Laboratory
Stanford Research Institute A.I. Group
Carnegie-Mellon University
Edinburgh University
M.I.T. Artificial Intelligence Laboratory
Project MAC Automatic Programming Group

We feel that we are now in a period of consolidation for these new higher level formalisms. A variety of implementation methods and ideas have been tried. Recently we have developed a modular activation formalism which unifies PLANNER-like languages and points to how they can be efficiently implemented. We propose to do the following:

To spend the next year completing an implementation of PLANNER based on this formalism.

To investigate the formal properties of the model using the techniques developed by Hewitt and Paterson for comparing powers of different control structures.

To do a feasibility study of constructing a processor based on our model. This computer might execute PLANNER-like formalisms 40-60 times faster than a PDP-10.

Our applications research is being done in conjunction with Project MAC under a contract for the development of systems that do interactive logistics planning. These systems should make it easy to add more useful knowledge and after-thoughts. P. Bishop (hardware implementation), R. Steiger (formal properties of the model), Gary Peskin (data base hash coding), Gorden Benedict (knowledge packaging) are working with Hewitt on these plans.

4.2 Parsing Strategy Research

Prof. V. Pratt is working on a Linguistics Oriented Language, LINGOL, started at Stanford in 1970, which has been used to write a "deep structure" analyser, an English-to-German translator prototype, a comprehension and question-answering program. LINGOL implements the bookkeeping details of a context-free parsing algorithm that provides for the intervention of user-supplied code at each application of a context-free rule, allowing the user to provide semantic details outside the scope of the context-free syntax. The program needs improvement in the parsing strategy, which currently has to consider all assignments of surface structure before choosing one. This is cautious but costly; it requires several thousand words of storage and several seconds of processor time for a 15-word sentence, but breaks down altogether for 30-word sentences. Given the grammars in use, this makes it unusable for such applications as reading newspapers, journals, this sentence, and others where it might otherwise be adequate. Pratt plans to test two or three strategies to circumvent this problem without complicating the user's problems. Pratt plans a conversational program that will start out as a version of the existing question answering system and grow as deficiencies in its English are repaired; it could prove of value to people who need a good English front-end for their programs, such as the one supplied by Woods to NASA lunar geologists, or to

linguists for other purposes.

4.3 Debugging Aids

S. Markowitz is studying debugging aids for the new languages embedded in LISP as meta-languages, and preparing a set of programs to analyse files and prepare reports about function cross-references and itemizations of function-calling patterns and variables. The use of heterarchical goal-directed program invocation through pattern-matching requires a new debugging technology, and we expect to develop one in a period shorter than was usual in the past, when the importance of such matters was rarely understood at language-development time.

4.4 Heterarchy

As the results of other work come to fruition, the early shortage of methods that worked minimally well is starting to be replaced by a richness of viable alternatives; this is what makes it reasonable to begin to attack practical problems in natural environments. E. Freuder is committed to taking a new look at our vision system, to plan a new version of such a system, polarized around the notion of heterarchy from the very start. Freuder imagines knowledge available in this system to interact at all levels with mechanisms available in the new Planning

languages.

4.5 Automatic Theorem Proving

Although there is a tradition among mathematicians and philosophers of attempting to model "reasoning" as a form of "logical" procedure, we maintain that there is a deep problem in the traditional attempt to separate the kinds of knowledge in the "data base" from the kinds of knowledge used to make plausible inferences thereof. But in order to understand the situation, we feel that we have to thoroughly understand the strengths and limitations of the logistic method. We have several small investigations aimed at what we believe are important questions in this area.

A. Brown is studying the problem of automatic derivation of proofs of theorems in abstract group theory, in an attempt to combine modern heuristic techniques, classical predicate calculus and the new problem solving languages.

A. Nevins is continuing his work on the same subjects, with emphasis on making the system more intelligent about case analyses, etc.

J. Geiser is studying a variety of theories for representing empirical knowledge within classical logics, and is exploring theories of inference schemes for logical data bases that are known to possibly contain inconsistencies

4.6 Research on Education and Cognitive Development

It is relevant at this point to comment on the relation between our work on Machine Intelligence and on Natural Intelligence. (This work is funded mainly by the NSF.) We have not taken "simulation of human thought processes" as a goal, or even as a guiding principle of the work on Machine Intelligence. Rather, we have adopted a no-holds-barred approach to achieving performance, irrespective of whether the mechanisms look like what might be happening in the human brain. However, we find increasingly that the concepts which our work on Machines forces us to adopt are applicable to Human thinking. They have led us to account for known phenomena in child development and to discover new phenomena. Most important, our procedural approach to thinking about thinking is distinguished from that of traditional psychology in the simple and immediate way in which it translates into suggestions for educational practice. For example, our discussion of how to formulate powerful, qualitative principles of common-sense reasoning (using primitives such as particular conservation and general program-structural ideas) could be paralleled by a very similar discussion on how to formulate aspects of scientific thinking usually classified as "intuitive" and so usually opposed to "formal". Doing so also makes them more teachable for exactly the same reason as it makes them more programmable. Thus our approach to education involves a much more radical attempt than usual to reconceptualize whole areas of knowledge.

A particular area of educationally directed work that has been receiving growing attention in the Laboratory could be called "qualitative intuitive physics". It is very possible that during the next year this work will become sufficiently mature to branch into programming projects directed at giving machines the ability to "think physically" in a very general sense. We expect this work to have practical consequences, for example, in achieving the "people-tracking" applications.