PROGRESS ON A COMPUTER BASED CONSULTANT

Peter E. Hart
Stanford Research Institute
Menlo Park, California

Abstract

Computer based consultants are systems that incorporate specialized bodies of knowledge and make this knowledge conveniently available to users who are not computer experts. This paper summarizes initial progress on a computer based consultant project aimed at helping a novice mechanic work with electromechanical equipment. We describe some properties and abilities of consultants, and present results to date on the problem solving, vision, and natural language components of our evolving system.

Keywords in this paper are: computer based consultant, advice-giving, problem solving, trouble-shooting, scene analysis, natural language understanding.

Introduction

One of the Increasingly prominent trends in computer science research has been the emphasis on incorporating specialized bodies of knowledge in computer programs and making this knowledge conveniently available to users who are not computer experts. Such programs—which we might call computer based consultants--can be viewed as stemming from the confluence of two lines of research. One line of research has centered on formulating and encoding a great deal of knowledge about a chosen problem domain in order to produce a program whose performance rivals expert humans. Often cited examples of this research include programs that analyze chemical structure,[1,2] perform symbolic integration,[3] or play board games well.S[5,6]

The second line of research has focussed on metnods for constructing a program that can carry on a dialog with a user. Important contributions to this research have come from work in computer aided instruction, and from work in understanding typed and spoken natural language. Representative examples of this work include programs to carry out a "mixed initiative" tutorial dialog,[7*8] to engage in a dialog about a toy block world[9] and to understand spoken English sentences about such diverse topics as plumbing,[10] news stories,[11] moon rocks,[12] or submarines.[13]

The work reported herein was sponsored by the Advanced Research Projects Agency of the Department of Defense under Contract DAHC04-72-C-0008 with the U.S. Army Research Office.

Perhaps one of the best examples to date of a complete computer based consultant Is the MYCIN system.[14] This system provides advice to physicians on the diagnosis and therapy of certain classes of bacterial diseases. It solicits various kinds of medical data from a physician user, can answer his questions (expressed in a restricted natural English format), and can accept advice from him regarding generally useful rules for diagnosis and therapy.

In this paper we describe initial progress on another computer based consultant. This new consultant is aimed at helping an inexperienced mechanic work with mechanical equipment. Before describing the functional components of the system, let us first consider some of the characteristics of the problem domain.

The Problem Domain

Imagine a mechanic (whom we will assume to be relatively inexperienced) working on a piece of equipment in a "work station" like the one sketched in Figure 1. He is typically responsible for a variety of Jobs, such as troubleshooting, repairing, or modifying equipment. In order to do these jobs, he needs certain kinds of specialized knowledge; he must know about the use of various tools, about principles of troubleshooting, and about principles of assembly and disassembly, and he must also know a certain amount of detail about the construction and operation of the specific equipment on hand.

A traditional way of conveying this knowledge to a mechanic has been through the use of manuals. A more nearly ideal, though usually impractical, way would be to make an expert mechanic continuously available as a consultant. The expert could identify various components, answer specific questions about equipment details, suggest troubleshooting sequences, hypothesize causes of failure, warn of hazards, and so forth.

In order to explore what would be involved in replacing the human expert by a computer based expert, we recorded a number of dialogs between expert human mechanics and novice mechanics. The dialogs concern the air compressor shown in Figure 2. (We shall use. this compressor throughout the paper for illustrative purposes.) Two excerpts from these dialogs are presented below. At the time the dialogs were recorded, the expert and novice were in different rooms and the expert viewed the scene only by means of still pictures taken through a television camera. (We did this to simulate the

limited visual information available to a computer based expert.)

The first excerpt concerns the subtask of installing a pump pulley on the pump.

Excerpt 1

:

Expert:	**The pump pulley should be next.**
Novice:	**Yes ... uh, does the side of the pump pulley with the holes face away from the pump or towards it?**
Expert:	**Away from the pump.**
Novice:	**All right.**
Expert:	**Did you insert the key--that is, the half-moon shaped piece?**
Novice:	**Yes, I did.**
Expert:	**Be sure you check the alignment of the two pulleys before you tighten the set-screws.**
Novice:	**Yes, I'm just now fiddling with that.**
Expert:	**OK.**
Novice:	**Tightening the Allen screw now.**
Expert:	**OK, thank you.**

:

This fragment illustrates several important abilities of consultants that contrast sharply with a static information source like a manual. First, notice that a question from the novice is answered directly and in his terms. There is no need for him to search through a mass of information, or to convert information from an abstract or "standard" form into a directly usable form. Notice that the expert is checking on progress by offering warnings and reminders about critical steps. This has the function not only of minimizing errors, but also of allowing the expert to keep track of the progress of the work. The latter function is the basis for the expert^{1}s ability to present relevant advice, and to present it In a context that is familiar to the novice.

The second dialog expert concerns the same subtask, but was carried out with different participants. It offers an interesting comparison of the different demands imposed by different skill levels:

Excerpt 2

Expert:	Install the pulley on the shaft.
Novice:	What is the first thing to do in installing the pulley?
Expert:	Rotate the shaft so that the slot (keyway) is on the top.
Novice:	OK .. . now what?
Expert:	Place the key in the slot.
Novice:	**Flat side upward?**
Expert:	**Yes.**

:

This short fragment dramatically Illustrates the ability of the expert to descend into detailed instructions in order to help a very naive user. This novice needs much more help than the first one did, a situation foreshadowed by his initial question about a relatively simple operation.

The short dialog excerpts exemplify some of the abilities that a consultant needs in order to be helpful to the novice mechanic. Both introspection and protocol experiments point out a number of other required abilities, among which are the ability to provide advice about troubleshooting; to describe the use of tools; to describe the appearance of tools (or to be able to point them out); and, of course, the ability to use language.

The abilities required of a mechanic's consultant impose a number of technical requirements on a computer system designed to fill that role. It is worth mentioning just a few of these requirements to illustrate the research problems we are addressing.

The set of problems that is perhaps the most characteristic of our project centers on providing advice about a task at any of several levels of detail, and of interacting with the novice as he uses this advice. In particular, multilevel plans must be created and represented, the novice must be modeled in order to determine the level of detail he needs, his performance must be monitored as he executes the task, and internal models must be maintained to reflect the current state of the task environment. This, In turn, Imposes a need for semantic representations that can accumulate a structured discourse history that evolves as the task proceeds, and that can provide linguistic contexts and clues to the novice's competence. Also, because so much information is communicated visually, we must be prepared to use vision to answer questions from the novice; but, because the world of machinery is exceedingly complicated visually, we must exploit geometric models and semantic constrains extensively if we expect to be able to answer a reasonable range of "visual questions."

It is interesting to note that most of the foregoing technical requirements are not peculiar to a mechanic's consultant; they are likely to underlie any computer baaed consultant system. In the next sections, we will outline the progress we have made on these research problems.

Functional Components

Let us begin by considering the system organization sketched in Figure 3. The consultant

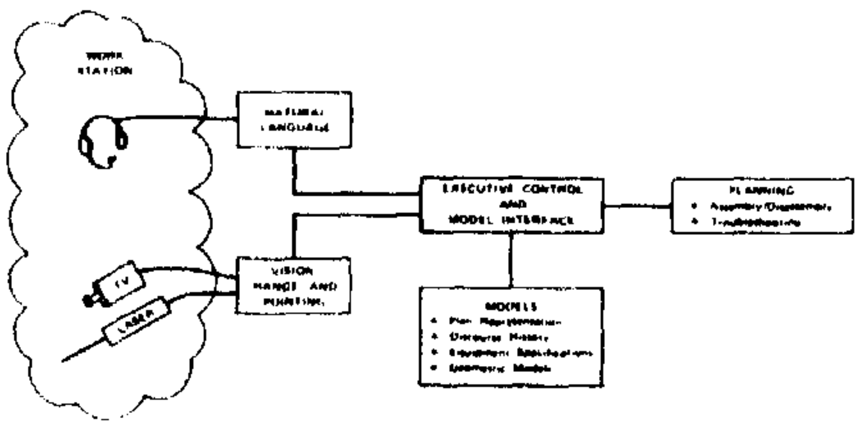


FIGURE 3 FUNCTIONAL ORGANIZATION

system interfaces with the physical domain of the work station through several devices. A headset enables the novice mechanic to talk to the system and to receive spoken replies; it represents a natural language component of the system. A visual component is represented by a color television camera and a laser rangefinder. The laser rangefinder is a mechanically scanned instrument, developed for this project, that generates an array of range values supplementing the color television picture. The range array and picture arrays (one for each primary color) can be placed in registration, providing a multisensory image that specifies the color of, and distance to, each point in the scene. The rangefinder can also be operated as a visual pointer, so that the system can answer questions like "show me the pressure switch" by pointing at it--that is, by illuminating the pressure switch with the laser beam.

The raw sensory data provided by the transducers are translated into internal representations by the natural language and visual functional components of the system. These internal representations trigger subsequent action. For example, a question about an assembly step might be answered either by referring to an assembly sequence already stored in a model, or by using the planning capability to compose a sequency if the model does not contain an appropriate one. Similarly, a question about the location of a part might be answered either by referring to a geometric model or by locating the part using new visual data. Of course, the natural language and visual components themselves need various kinds of model information in order to translate raw data into internal representations. For example, in order to understand a given sentence, it is necessary to access a discourse model in order to establish the referents for pronouns or determined nouns.

With this system overview as a background, we can review the current stock of ideas and programs for each of the functional components.

Planning for Assembly

The system component that has received the most attention to date has been the planner and associated model, for composing assembly and disassembly sequences- We have emphasized this because assembly and disassembly are subtasks of virtually all typical work station tasks. For example, many troubleshooting jobs and almost all repair jobs require some amount of disassembly and reassembly of the machine.

Let us use the task of assembling the air compressor to illustrate In a simplified way how assembly plans are produced.* Three different types of knowledge are used- a model of the specific air compressor, a procedural model that encodes more general information about how parts are fastened together (i.e., assembled), and a planner that has abstract knowledge about how plans can be represented and about how the steps of a plan can interact.

The compressor assembly model is essentially a graph whose nodes correspond to the parts of the compressor (the motor, pump, pulleys, and so on), and whose arcs correspond to the mechanical connection between parts. A considerable amount of information is usually associated with each arc. For example, the arc representing the connection between a pulley and its shaft may include information about the set screws and key. (The key prevents relative rotation between pulley and shaft.) Similarly, the arc connecting the belt cover and its support may contain information about the number and size of the sheet metal screws. This equipment-specific model also contains certain auxiliary information peculiar to the compressor, like the fact that the pump cannot be installed if its pulley is already on the pump shaft.

Each generic type of connection has associated with it a set of procedures that contain instructions about how that connection is physically accomplished. For example, a procedure associated with installing a pulley on a keyed shaft might include specific instructions about inserting the key and tightening the set screws. Note that this procedure is independent of any specific piece of equipment; it offers generally useful knowledge about how a certain job in the domain of mechanical equipment is done, and it would be invoked whenever that Job was necessary. In addition to the specific instructions, procedures of this sort contain calls to other procedures that elaborate in more detail how the given job is done. In our pulley and shaft example, we might want to call more detailed procedures for, say, describing how

* Disassembly plans are essentially similar, and will not be discussed explicitly.

to align pulley and shaft or for dealing with rusty parts. This hierarchical structuring of procedural knowledge forms the basis for producing plans that can be stated to a novice at any of several levels of detail.

The procedural model of assembly operations allows the planner to generate Instructions about how to connect two specific parts, but It does not select the order In which parts are to- be connected. This is the job of the general planning program. The planner adopts the view that if there are n connections to be made between pairs of parts, all connections are equally important and that there is no prior reason to prefer any particular order. That is, it initially assumes that all n assembly steps will be made in parallels-logically, as a conjunction. However, it then expands the steps in greater detail, and examines the preconditions and effects of these steps to see if there is any interference among them. To continue our example, it would discover that the pump can be installed only if there is no pulley on its shaft. This would interfere with a different assembly step; namely, installing the pump pulley on the pump shaft. The planner recognizes this potential conflict, and imposes an order so that the pump will be installed before its pulley is placed on its shaft. When all conflicts of this nature are resolved, the remaining steps can indeed be logically performed in any order.

This ability to recognise alternative orderlngs of steps has major Implications for any computer based consultant: A human performing a task may well take the initiative on occasion and choose an ordering for certain stepa, and it ia important for the consultant to know whether this choice is valid. Equally, the availability of alternative orderlngs affords an opportunity to appeal to other ordering criteria like ease of physical operations.

A plan is repreaented as a <u>procedural net,</u> a fragment of which is shown In simplified form in Figure 4. Each node corresponds to an assembly step at some level of detail. The net represents

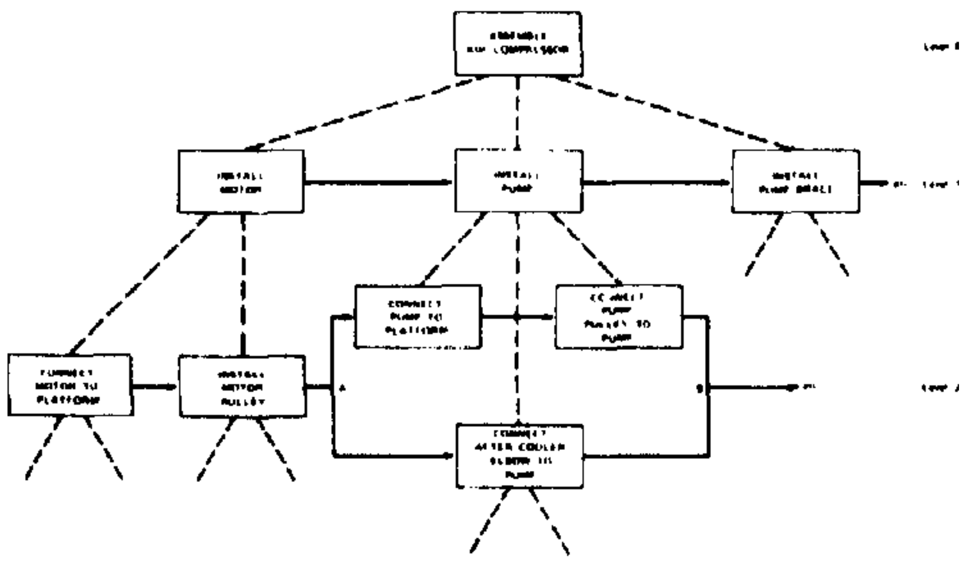


FIGURE 4 A FRAGMENT OF A PROCEDURAL NET

a hierarchy of plana, all accomplishing the same task but stated at varying levels of detail. The i^{th} row of the net represents one complete plan at the 1^{th} level of detail, and the dotted lines indicate the expansion of a step into a more detailed subplan. Notice that the Level 2 plan splits into two parallel paths at A and merges together at B in order to represent the fact that the two subplans can be performed in either order.

Procedural nets have proven useful In several ways. Perhaps the moat obvious is that it allows us to apeclfy a plan to the novice mechanic at varying levels of detail. Typically, the novice will understand some atepa at a high level and need little or no additional elaboration, whereas he will be mystified at other steps and need to have them expanded into more complete instructions. By keeping track of an execution path through the net, we can link steps at the various levels of detail. The more general problem here is to learn how to monitor the mechanic's performance as he executes a task. We would, for example, expect the system to ask occasional questions of the novice (Just as the human expert did in the dialog) in order to monitor his progress as he proceeds through the net. Thus far, our system is not that flexible and monitors progress by adhering to a more limited dialog format.

In addition to the several uses of procedural nets at plan execution time, they are also used during planning to represent partially formed plans. This allows us to restart the planner to modify an existing plan during the course of its execution, which in turn permits us to respond to information discovered as the assembly physically proceeds. Further discussion of procedural nets and arbitrary orderlngs of plan steps will be found in a forthcoming paper.[15]

Before leaving the subject of assembly planning we should mention a second type of hierarchy which is distinct from the hierarchy of plan details that we have been discussing. This second hierarchy deals with levels of equipment, and is motivated by the fact that often the major parta of a mechanical device are themselves components that can be assembled and disassembled. For example, the pump of the air compressor has a piston, crankshaft, and valves and is in fact similar in some ways to a simple one cylinder gasoline engine. We thus expect that the general scheme for assembly planning described above would be replicated hierarchically to deal with several levels of components of the equipment. Ideally, then, we hope eventually to be able to provide consultation at aeveral levels of detail about any of several levels of equipment components. We have only recently begun considering this second hierarchy, but it appears to entail a relatively straightforward extension of the ideas discussed already.

Planning for Troubleshooting

Troubleshooting Is a key element of a mechanic's job, and often represents the task requiring the highest levels of skill and experience. In spite of its obvious importance, we have given it relatively little attention thus far because of our decision to first reach a reasonable level of competence at assembly planning. Accordingly, we can offer here only tentative remarks about troubleshooting, and describe the two main approaches that we are currently pursuing.

The two approaches of interest might be termed the "engineer's approach" and the "technician's approach." The engineer's approach rests on a detailed tracing of cause and effect in order to find where the causality chain breaks down. The technician's approach eschews this time-consuming effort (except as a last resort) and instead relies on experience to suggest likely candidate faults to be investigated directly.

Let us use the example of the air compressor to contrast these two approaches. Suppose the stated problem is that the compressor can no longer power several air tools that it normally can drive. The engineering approach might begin by tracing the electrical circuits to ensure that the motor is receiving the correct voltage and current. Assuming that it is (and that the belt connecting the motor and pump is in place), the next step might entail checking the volume and pressure of air output from the pump unit. An inadequate output would pinpoint the pump as a likely suspect, and an investigation of It would continue in the same vein. In contrast to this approach, a skilled maintenance mechanic familiar with the air compressor knows that the reported symptoms are often caused simply by a lack of lubricating oil in the crankcase of the pump. He would fill the crankcase immediately, and if the air compressor was then able to power the usual air tools he would assume that his suspicions were correct and that the problem was now corrected.

It seems clear that a computer based consultant needs to be able to employ both of these approaches, and be able to switch between them when appropriate. An implementation of the engineering approach has begun in a simple way; it relies on a simulation model of the equipment to suggest a sequence of tests or observations corresponding to a sequence of causes and effects. An indication of a malfunctioning component is obtained whenever an observed effect differs from the effect predicted by the simulation. An alternative to this implementation Is suggested by Shortllffe's[14] rule-driven system, in which each rule corresponds to a simply stated fact or rule of thumb. We are currently investigating the extent to which some sort of rule-drive system can be adapted to mechanical troubleshooting.

Vision

The domain of electromechanical machinery is an extraordinarily difficult one In which to do automatic scene analysis. Equipment and components usually have only a limited range of hue and saturation values. Visual texture similarly is very limited; specular highlights are often complex, and can vary depending on such vagaries as "oil-canning" of sheet metal parts; shadow patterns can be very complex and depend in complicated ways on the particular stage of assembly; and, finally, machines and components assume a very wide variety of shapes. Indeed, scene analysis In the domain of machinery is likely to be more difficult than, say, in the domain of offices or even landscapes, because in the latter two domains there is a much richer variety of perceptual clues. For these reasons, we have elected to approach the vision problem by capitalising on prior knowledge of visual appearances and geometric relations, and by limiting our aspirations at the outset to a set of subproblems that are both Important and tractable.

We have implemented thus far several modular vision packages to perform specific tasks. One interesting module is a tool recognizer that can accept limited semantic descriptions of tools, build a model of the tool from the description, and aubsequently use this model to discriminate an Isolated hand tool from a set of alternatives. Let us outline briefly how this is done.

Consider the open-end wrench shown in Figure 5. This wrench is a member of a large class of

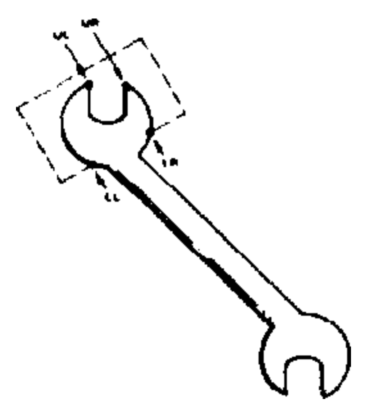


FIGURE 6 AN OPEN END WRENCH

hand tools characterized by one or more connected shafts or handles, with an optional "business end" that is applied to a fastening; screwdrivers, nutdrivers, hammers, and many varieties of wrenches fall in thi,s class. The basic operation of the tool recognizer is first to find a shaft, or handle, based on typical aspect ratios for tools, and then to concentrate attention on the end of the shaft in order to determine the tool type. Model information about the endplece is provided in advance by an operator, who may use a circumscribed rectangle (as shown In the figure) as a reference for his description. In the case of the open-end wrench, the operator could specify that the end-piece has a convex curved outline between its upper left (UL) and lower left (LL) endpoints, is again curved convex between the upper right and lower right, and is U-shaped between the upper left and upper right endpoints. Relative acale information would typically also be added to ensure that various parts were reasonably proportioned.

In operation, after finding a tool shaft or handle, the tool recognizer module constructa a loose bounding rectangle in order to determine whether:

1) an endpiece is attached to the end of the shaft/handle;
2) the end of the shaft/handle is connected to another shaft/handle; or
3) nothing is attached to the end of the shaft/haudle (e.g., the ends of an. alien wrench, the free end of a tool handle).

Coarse size and shape tests are performed to eliminate broad classes of tools. For example, hammer-like tools are easily distinguished from screwdrivers and wrenches on the basis of aspect ratio. In the case of an endpiece, the loose rectangle also gives gross upper, lower, left, and right limits, which are then refined by shrinking the rectangle to a minimum enclosing size. Figure 6(a) illustrates an intermediate processing step using a combination wrench as an example; the program has located the tool shaft, and has circumscribed a tight bounding rectangle around one of its ends. On the basis of this information, the tool may be an open-end wrench, a box-end wrench, or a nut-driver. Included in the model information for these tools are tests for distinguishing features possessed by the tools. For example, the open-end and box-end wrenches have an opening in the center of the endpiece, whereas the nutdrlver is solid in the center. The program generates a square grid of points within the bounding rectangle and determines whether any of these points is within the brightness range of the background region (Figure 6a). Since some of the points pass the test, the nutdrlver is ruled out as a possibility and the program looka for a test which will distinguish the open-end wrench from the box-end wrench. Using the test "Open-at-top," the program attempts to scan an uninterrupted straight line from the center of the bounding rectangle to a point on the top segment of the rectangle, as shown in Figure 6b.

Using the distinguishing features tests, the program has refined its hypothesis about the identity of the tool to a single candidate, the open-end wrench. It now attempts to verify this hypothesis using model information which describes the shape of the endplece, in this case by finding the curved convex sides and the U-shaped opening between the upper right and upper left endpoints of those sides.

It is interesting to contrast this approach to tool recognition with a brute force template matching approach. On pragmatic grounds, template matching is not very attractive simply because of the variety of tools (and of sizes of tools), the large number of translations and rotations that a tool can assume, and the inadequacy of template matching if a tool is partially occluded. On conceptual grounds, the approach outlined above is interesting chiefly because of the ease with which new tools can be described by their functional characteristics. In the wrench example, the description is a primitive attempt to say that "anything that can be used as a wrench Is in fact a wrench." Tools are a particularly good domain in which to pursue this philosophy because they are artifacts with clear functional purposes.

Two other interesting vision modules entail the ability to point to specific components of machinery. Both rely on an underlying geometric model of the equipment at hand. The first module enables the consultant to answer requests of the form "Show me the X" by pointing at X with the laser rangefinder. This is accomplished using a hidden surface algorithm of a very simple variety to locate the outline of a visible surface of the desired component. Once the surface has been determined, a little care is needed to guard agalnat the possibility of pointing at the component by pointing through a hole or a concavity. (For example, we would not want to point at a doughnut-shaped part by pointing at the hole.) A simplified form of the medial axis transformation is used to find a thick region of the part that can serve as a target.

The second pointing ability is intended to allow the novice technician to ask questions of the form "What part is this?*' by pointing at the unknown part himself. We currently uae a wand with a small light at the tip to simplify processing, yet retain a natural form of pointing for the novice. A ray in space is defined by the wand tip and the camera lens; intersecting this ray with a geometric model of the equipment provides the Information needed to answer the question.

The modules described above have some direct extensions that we expect to pursue in the near future. For example, the pointing modules rely heavily on the use of geometrical models of equipment. Using these models, we expect to develop means for finding and determining the orientation of a machine or component of Interest to the novice. An open question centers on the extent to which range data will simplify the problem. We have already devoted a good deal of attention to a formalism for combining multlsensory data;[16] we will need to explore the ease of applying the formalism to the complicated collection of shapes typical of most machinery.

Having said something about our plans for vision, we should perhaps also mention that we are not planning in the near future to use vision to answer fine grained mensuration questions like "Are the pulleys aligned sufficiently well?" We expect that most realistic questions of this form will tax the resolution of our transducers and prefer not to devote our energy to this class of problem.

Language Communication

We were persuaded at a very early stage that natural language communication would have to be an integral part of a mechanic's consultant. Aside from the unfamiliarlty a mechanic presumably has with a computer terminal, it seems unreasonable to ask, say, an auto mechanic to crawl out from under a car in order to ask how to replace a U-joint.* Thus, our ultimate goal is to allow the novice to use natural English speech to talk to the computer based consultant. Symmetrically, a second goal is to enable the consultant to talk to the novice using ordinary speech. In recognition of the difficulty of these ultimate goals–chiefly the first one--we have set as intermediate goals the development of more restricted language components that will still allow our experiments to proceed.

Our current language ability rests heavily on a commercially available device known as a phrase recognizer.* The phrase recognizer is able to recognize an isolate speech fragment up to two seconds long, once the device has been trained by listening to the novice say each phrase (or word) several times. In our experimental work, a typical vocabulary of roughly 50 words is evenly divided between object names and control words. Control words include items like "Why," "How," and "Show-me-the," while object names are mainly part names.

To enable the computer based consultant to talk to the novice, we use a commercially available phoneme synthesizer.§ The synthesizer accepts a sequence of phoneme specifications from the computer and converts these to audible form, producing speech output. A programmer-defined output vocabulary is implemented by selecting a phonemic representation for each word; once this has been done, that word (which is reasonably understandable) may be used in any context. There is thus no intrinsic limit to vocabulary size, and the system is much more convenient and compact than say, a direct digital representation of acoustic waveforms.

The combination of phrase recognizer and phoneme synthesizer has been notably useful in allowing us to experiment with fragmentary versions of a consultant system. They permit us, at the very least, to gain some intuition about the "live" behavior of such a system. On the other hand, the phrase recognizer in no sense "understands" linguistic content, there is a very limited ability to handle even simple sentences, and there is no representation of an utterance other than its recognition as one of a limited number of alternative phrases. Consequently, we view the current capability as being only an experimentally useful (and easily achieved) interim one, and are devoting our energy to developing a more adequate natural language component for the consultant.

Our Immediate goal in this regard is to implement a language component able to deal with natural text input. Simplifying matters for discussion purposes, we need three things in order to accomplish this: a sentence-by-sentence translation facility, an internal representation of input sentence meanings, and a discourse analysis model. Sentence-by-sentence translation Is driven by Paxton's "best-first" parser.[17] This parser was originally intended for, and indeed is used in, a natural speech understanding system, but It has been modified to accept text input while work in acoustics continues. Accordingly, let us focus attention on the Internal representation, which doubles as a target language for the parser.

We are planning to use a semantic net representation that follows roughly along the lines suggested by Norman[16] and Simmons.[19] Let us use a few simplified examples to convey both the general approach and its application to our particular needs.

A semantic net representing some information about simplified air compressors is shown in Figure 7. Each node of this particular net

Our intuition recently received some support when we learned of a simple program providing diagnostic advice to auto mechanics. When field-tested by a major auto manufacturer, it was found to be "A disappointment, because the men wouldn't go near a keyboard."

Ours, the VIP-100, is manufactured by Threshold Technology, Inc. Other suppliers are also entering the market.

We use a Votrax voice synthesizer, manufactured by Federal Screw Works.

We plan to extend the system from text to speech at a later date.

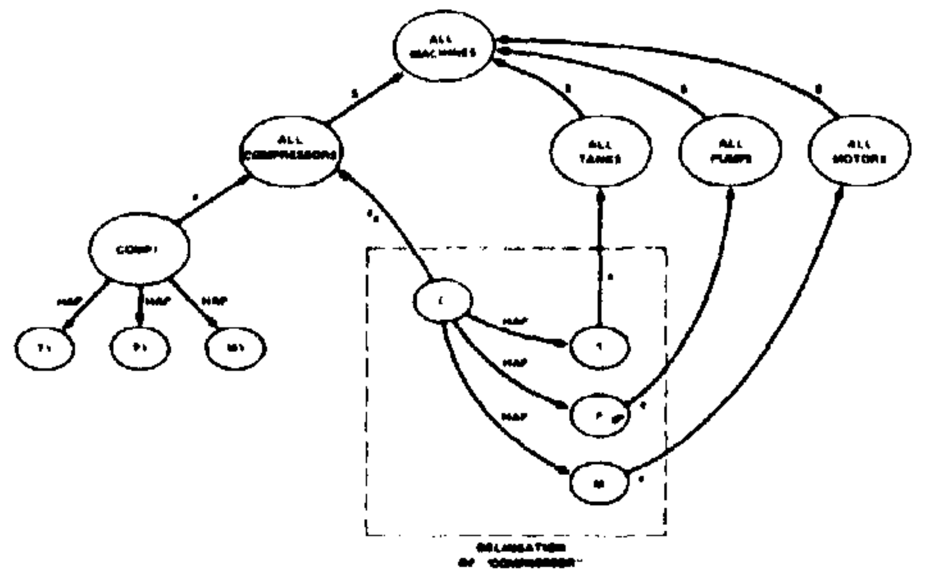


FIGURE 7 SEMANTIC NET REPRESENTING AN AIR COMPRESSOR

represents an object or a set of objects, and each arc a binary relation between nodes. The upper nodes indicate that the set of air compressors, of tanks, of pumps, and of motors are each subsets (the S relation) of the set of all machines. The set of air compressors is partially defined--we will say delineated to indicate the definition is only partial—by the subnet enclosed in the box. This subnet represents a prototypical member C of the set of air compressors; we use e_d to show that C is the delineating element of the parent set. Compressor C is shown as having three parts (HAP means has-as-part): T, P, and M. The e (element-of) arcs emanating from these nodes indicate that they are respectively elements of the set of tanks, pumps, and motors. Thus, the portion of the net we have discussed so far represents the fact that a typical air compressor is composed of a tank, a pump, and a motor, and that all of these objects are machines. The remaining portion of the net represents the fact that there is a particular compressor called COMPI (it is an element of the set of all compressors), and that it has as parts a particular tank Tl, a pump PI, and a motor Ml.

Nodes can also represent abstract entitles like relations. The net in Figure 8 is a representation of the "drive" relation (drive in the sense of "to power") between two pieces of machinery. It shows that drive relations are a subset of the family of all relations, and it delineates drive by displaying a subnet of a typical drive relation D. The delineation shows that drive relates two objects, a driving object M' and a driven object P'. By following the e arcs out from these nodes, we see that M' must be a motor, and that ?' must be a component that can be driven. The net also shows that pumps are among

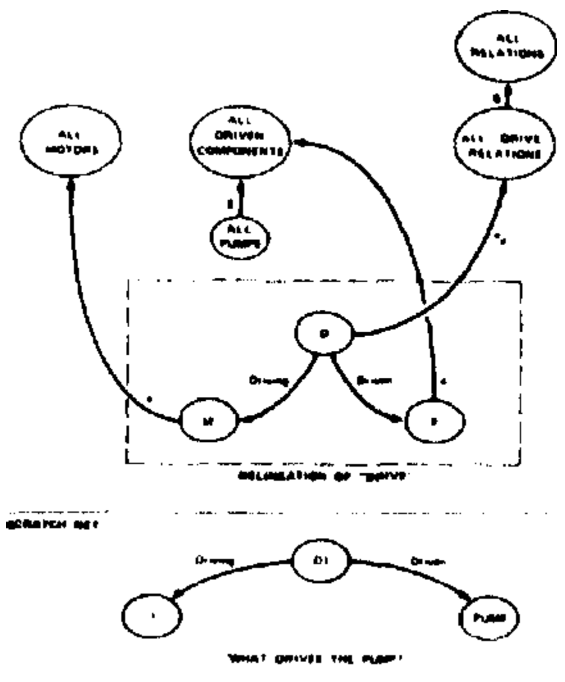


FIGURE 8 REPRESENTATION OF "DRIV

Che set of components that can be driven, whereas motors are not.

If we are now given the question "What drives the pump?" the net of Figure 8 provides the semantics that enable us to parse the question satisfactorily. In particular, it shows that drive is a binary relation, and that "pump" is something that can be driven. (In contrast, the question "What drives the motor?" would be rejected because motors are not shown by the net as being drivable.) The "scratch net" at the bottom of Figure 8, which is essentially a copy of the delineation subnet, represents the parsed question "What drives the pump?".

Although the net in Figure 8 enables us to parse the question, it does not contain enough information for us to construct an answer. For this, we need the additional information represented in Figure 9. Figure 9 shows fragments of

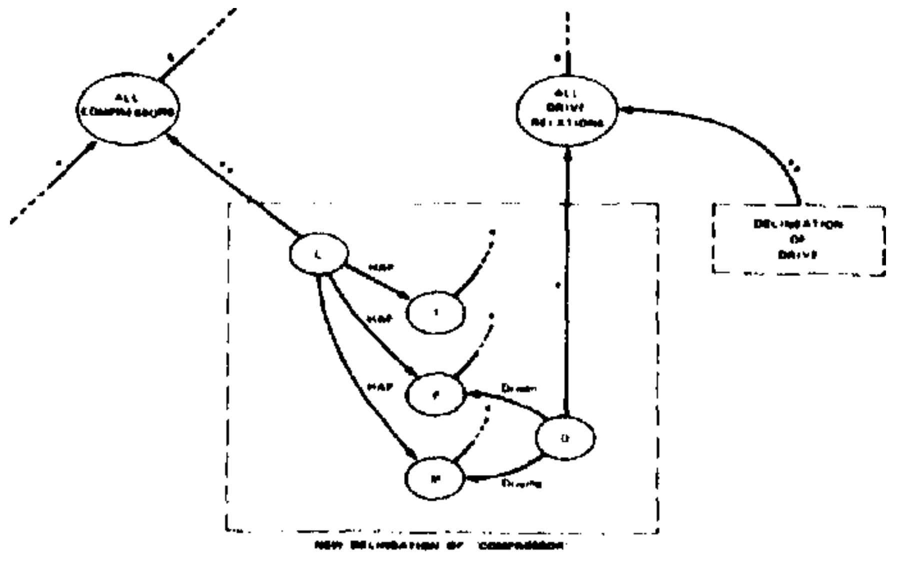


FIGURE 9 AUGMENTED REPRESENTATION OF AN AIR COMPRESSOR

the previous two nets, but we have augmented the delineation of the typical compressor C. The new delineation uses the relation D' (an element of the set of drive relations) to include the information that motor M drives pump P. Using this information, it is possible to match the scratch

We are simplifying some bookkeeping details needed in practice to associate T1 with all tanks, P1 with all pumps, and so forth. Conceptually, the reader can think of COMP1 and C as having matched subgraphs.

net in Figure 8 with the net partially shown in Figure 9 and thus to construct the correct answer. Further details are available in a forthcoming paper by Hendrix.[20]

The preceding discussion sketches our current design for semantic representation. The third part of the language component has to do with the use of a discourse history to provide the context in which to understand a sentence--for example, to resolve references. In the preceding example, we passed over the issue of how "the pump" is associated with the particular pump at hand (say pump PI of Figure 7). We expect to rely on an accumulation (or a summary) of previous scratch nets, with links to procedural nets, for this purpose, but do not yet have a detailed design for this aspect of the system.

A Brief Example

We are still some distance away from having a smoothly running consultant system containing all the functional components described above. Nevertheless, we include the following fragmentary example, from a transcript of a live voice experimental run, in order to give a little of the flavor than a more nearly complete system would have. For ease of comparison, we again have specified the task to be "Assemble the air compressor."

System: Try to achieve assembled aircompressor.
Novice: How?
System: The top level plan is . .. install pump ... install pumpbrace ... install pumppulley ... install belt ... install belthousingcover ...
System: Ready for questions.
Novice: Show me the pump.
(Laser beam moves to point at pump.)
System: Ready for questions.

Novice: What is this? (Pointing with wand at tank platform)
System: Tankplatform.
System; Ready for questions.
Novice: None.
System: Install pump.
Novice: How?
System: Connect pump to platform.
Novice: How?

System: Get 4 l&1/4-inch #5/16 hexagonal bolts and 4 #9/16 flat washers and 4 sire 5/16 hexagonal nuts.
Novice: OK.

In this example, the system has composed a hierarchical plan for transforming an initial state of the compressor to the desired final state of complete assembly. The novice then executes the plan, while the system keeps track of the current state by using the procedural net representation of the plan. The simple replies of "OK" and "How?" (or their equivalents) tell the system either to move to the next step at the current level of detail, or to expand a step into a number of more detailed actions. If the novice were to ask "Why?'' some step was suggested, the system would use the procedural net to construct an answer that might involve either supergoals of the current step or subsequent steps of the current subplan.

Future Problems

It is obvious that a great deal of work re* mains to be done to bring the individual components of the system to a uniformly high level of sophistication, and to integrate these components into a smoothly running system. It is interesting to note, however, that even in its present rudimentary state the system exhibits a surface behavior that Impresses outside observers favorably. (We will forgo the opportunity to speculate on the implications this observation has for the presumed mechanisms of Intelligence.) In the remaining few paragraphs, we will briefly discuss the outlook for continued progress on the consultant system.

It seems to us that the situation is reasonably optimistic, at least as far as the individual components of the system are concerned. Planning for assembly/disassembly appears to be quite well in hand, and the only real uncertainties center on how much work la needed to encode how much detail for machines of Interesting complexity. Planning for troubleshooting is less advanced, but there is a good stock of ideas available and at least one demonstration program (Shortliffe's MYCIN program) to encourage us that quite sophisticated troubleshooting abilities are within reach. Vision, however, is clearly difficult in the domain of real machinery, and a general and powerful capability here is not likely to be forthcoming in the near future. However, by relying heavily on geometric and other models, which we assume would typically be available, it appears that a vision component can be evolved that, though limited, would still be very useful. Natural speech understanding is not yet a reality in our system, but good progress is being made on that topic at a number of laboratories.[10, 11 12, 13] By carefully matching our interim language designs (e.g., the semantic net representation) to this body of work, we expect to be able to make use of forthcoming speech understanding systems with minimum effort.

It is less easy to assess the difficulty of integrating all of the functional components into one smoothly running system. One obvious problem is the current multiplicity of models; at the moment, we have at least one model each for assembly planning, troubleshooting planning, vision, snd language. Some of these models may be combined—for example, the procedural net may play a role in modeling a discourse history. The fundamental problems, however, are much deeper, and involve issues of coordinating very diverse knowledge sources,[11] of thinking versus acting, and of human novice effort versus "expert" computer effort. All this persuades us that computer based consultant systems are likely to continue to be a fruitful domain for artificial intelligence research, in addition to offering promise as a means for deploying knowledge usefully in an increasingly complicated world.

Acknowledgment

The ongoing project described in this paper is of course the work of a large number of people. I would like to especially acknowledge the contributions of G. Agin, A. E. Brain, B. Deutsch, R. 0. IXida, R. E. Fikes, T. D. Garvey, J. Gaschnig, G. G. Hendrix, N. J. Nllsson, W. Paxton, E. Sacerdoti, G. L. Sutherland, J. M. Tenenbaum, and H. C. Wolf.

References

1. B. G. Buchanan, G. Sutherland, and E. Felgenbaum, "Heuristic DENDRAL: A Program for Generating Explanatory Hypotheses in Organic Chemistry," Machine Intelligence. Vol. 4, B. Meltzer and D. Michle, eds., pp. 209-254 (American Elvesier Publishing Company, New York, New York, 1969).
2. B. G. Buchanan and J. Lederberg, "The Heuristic DENDRAL Program for Explaining Empirical Data," Proc. IFIP Congress, Vol. 71, Ljubljana, Yugoslavia, 1971; also Stanford Univ. AIM 141.
3. Joel Moses, "Symbolic Integration: The Stormy Decade," Proc. ACM 2d Symposium on Symbolic and Algebraic Manipulation. S. R. Patrick, ed., Los Angeles, California (March 23-25, 1971).
4. A. L. Samuel, "Some Studies in Machine Learning Using the Game of CHECKERS II--Recent Progress," IBM J. Res. Dev,. Vol. 11, 601-617 (1967).
5. R. Greenblatt et al., "The Greenblatt Chess Program," Proc. AFIPS Fall Joint Comp. Conf., 801-810 (1967).
6. A. L. Zobrist and F. R. Carlson, Jr., "An Advice-Taking Chess Computer," Scientific American. Vol. 228, No. 6, pp. 92-105 (June 1973).
7. J. S. Brown, R. R. Benton, and A. G. Bell, SOPHIE: A Sophisticated Instructional Environment for Teaching Electronic Troubleshooting (An Example of AI in CAI), BBN Report No. 2790 (Bolt Beranek and Newman, Inc., Cambridge, Massachusetts, March 1974).
8. A. M. Collins, J. R. Carboneil, and E. H. Warnock, "Semantic Inferential Processing by Computer," Proceedings of the International Congress of Cybernetics and Systems. Oxford, England (August 1972).
9. T. Winograd, "Procedures as Representation for Data in a Computer Program for Understanding Natural Language," Tech. Report AI TR-17, MIT, Cambridge, Massachusetts, 1971. Published as Understanding Natural Language (Academic Press, New York, New York, 1972).
10. D. E. Walker, "The SRI Speech Understanding System," IEEE Symposium on Speech Recognition, pp. 32-37 (April 1974).
11. Victor R. Lesser et al., "Organization of the Hearsay II Speech Understanding System," IEEE Symposium on Speech Recognition, pp. 11-22 (April 1974).
12. W. A. Woods, "Motivation and Overview of BBN Speechlls: An Experimental Prototype for Speech Understanding Research," IEEE Symposium on Speech Recognition, pp. 1-10 (April 1974).
13. H. B. Rltea, "Voice-Controlled Data Management System," IEEE Symposium on Speech Recognition, pp. 28-31 (April 1974).
14. E. H. Shortliffe et al., "An Artificial Intelligence Program to Advise Physicians Regarding Antimicrobial Therapy," Computers and Biomedical Research. Vol. 6, 544-560 (1973).
15. Earl Sacerdoti, "The Non-Linear Nature of Plans," submitted to IJCAI IV.
16. J. M. Tenenbaum, "On Locating Objects by Their Distinguishing Features In Multlsensory Images," Comp. Graphics and Image Processing. pp. 308-320 (December 1973).
17. W. H. Paxton, "A Best First Parser," IEEE Symposium on Speech Recognition, pp. 218-225 (April 1974).
18. Donald A. Norman and David E. Rumelhart, "Active Semantic Networks as a Model of Human Memory," Third International Joint Conference on Artificial Intelligence, pp. 450-457 (August 1973).
19. R. F. Simmons, "Semantic Networks: Their Computation and Use for Understanding English Sentences," Computer Models of Thought and Language, pp. 63-113, R. Schank and K. M. Colby, eds. (W. H. Freeman and Company, San Francisco, 1973).
20. G. G. Hendrix, "Expanding the Utility of Semantic Networks Through Space Partitioning," submitted to IJCAI IV.

FIGURE 1 A WORKSTATION

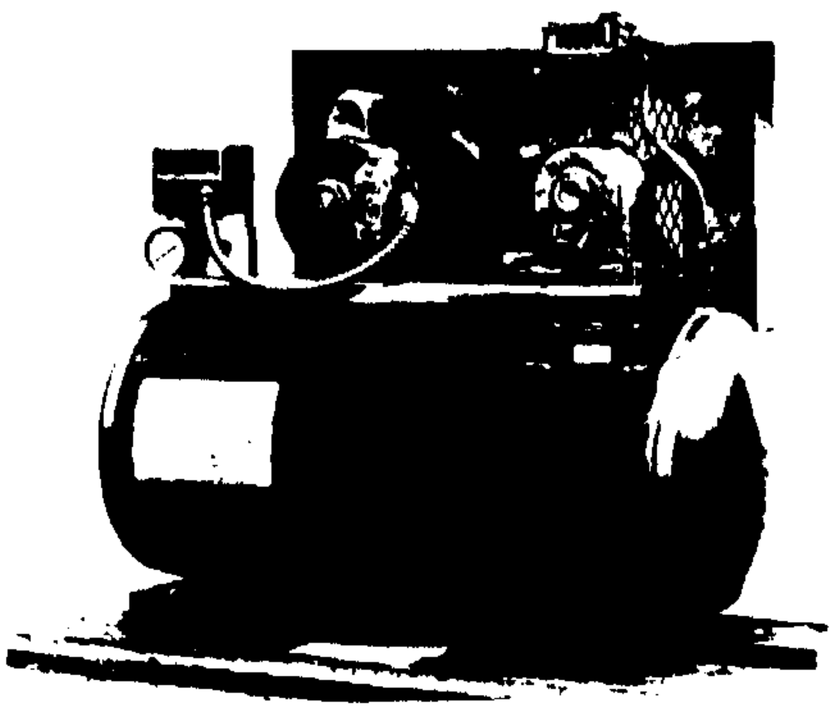
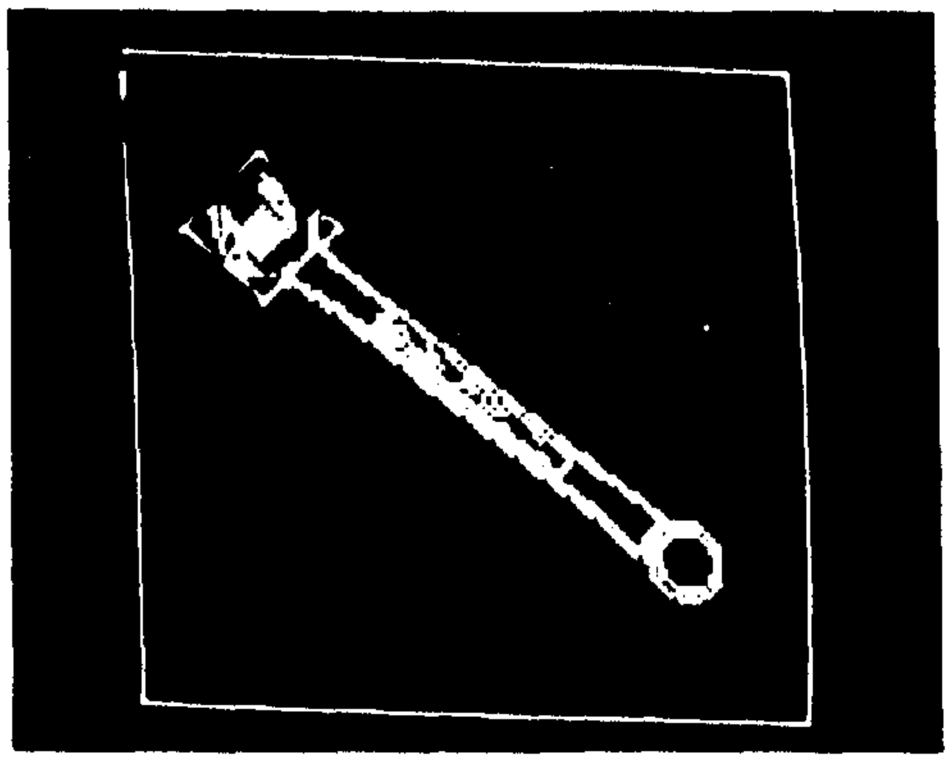
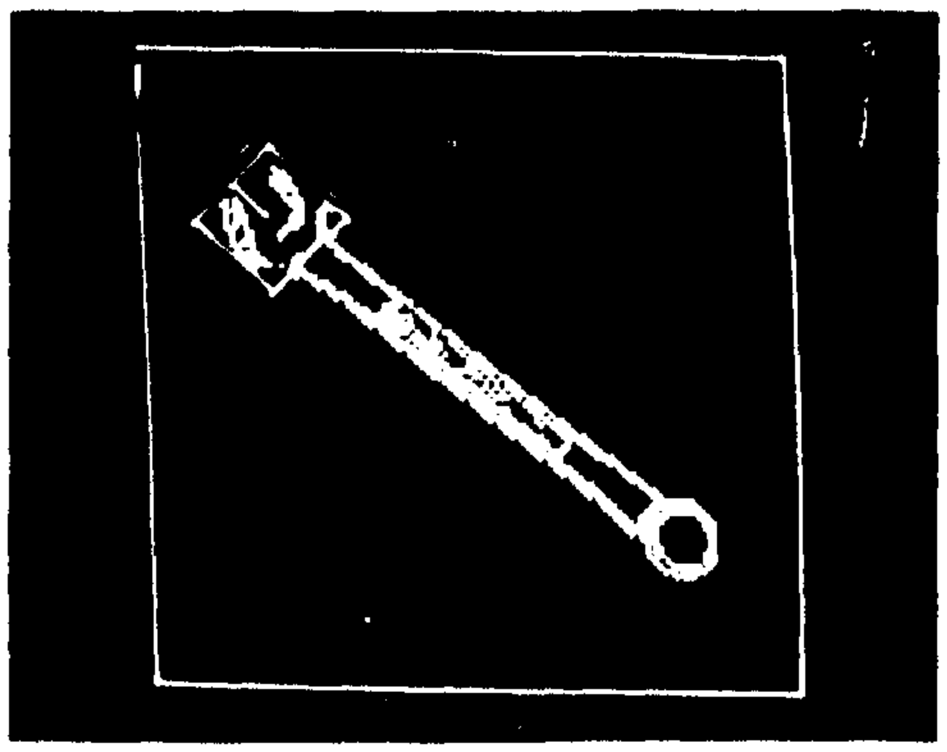


FIGURE 2 A SMALL AIR COMPRESSOR



(a) ENDPIECE LOCATED



(b) ENDPIECE SHAPE CHECKED

FIGURE 6