SOPHIE - A PRAGMATIC USE OF ARTIFICIAL INTELLIGENCE IN CAI

John Seely Brown and Richard R. Burton

Bolt Beranek and Newman Inc.
Cambridge, Mass. 02138

Key Words and Phrases: Artificial Intelligence, CAI, Natural Language Understanding

Abstract

This paper describes a fully operational AI-CAI system (accessible over the ARPANET) which incorporates Artificial Intelligence techniques to perform question answering, hypothesis verification and theory formation activities in the domain of electronic troubleshooting. Much of its logical or inferencing capabilities are derived from uses of simulation models in conjunction with numerous procedural specialists. The system also includes a highly tuned structural parser for allowing the student to communicate in natural language. Although the system is extremely large it is sufficiently fast to be thoroughly exercised in a training or classroom environment.

Introduction

Although digital computers have become increasingly more powerful and versatile, their use in instruction has grown primarily in but one dimension: that of finding cost-effective ways of providing more students with access to frame-oriented CAI systems. The purpose of this paper is to describe the basic mechanisms and design philosophy of a different kind of CAI system: one that provides a qualitatively new type of instructional environment which has been made possible by taking fuller advantage of the symbol manipulation capabilities of a computer in conjunction with some recent advances in Artificial Intelligence (AI).

We had two main motivations for building this system (named SOPHIE**). First we wanted to demonstrate that the notion of using AI techniques to build an "intelligent" CAI system was not purely a pipedream but that in fact a system could be built that was sufficiently complete and efficient that it could be used as an experimental tool in a classroom environment. Second we wanted to use this system to explore the teaching of problem solving skills such as electronic troubleshooting without being constrained to pose only problems which contained extensionally defined solution sets. For example, in troubleshooting (or theorem proving) there are many logically reasonable sequences of measurements (or proofs) that a student could make. We wanted to allow the student freedom in choosing the way in which he would go about solving his problem while still expecting SOPHIE to be able to monitor all his decisions and provide useful feedback whenever he wished it. In summary we wanted SOPHIE to have sufficient symbolic knowledge, problem solving strategies and natural language capabilities that it could mimic many of the capabilities of a human tutor in a problem solving situation. In particular, it had to "understand" its subject domain so that it could reason <u>on its own</u> about situations not <u>pre-stored or</u> programmed in.

The idea of using AI techniques in CAI was originated by Carbonell with his mixed-initiative SCHOLAR systems (Carbonell 70, 73). Since then, other systems for teaching symbolic logic (Goldberg 73), meteorology (Brown et al 73) and for interpreting nuclear magnetic resonance spectra (Sleeman 74) have explored how to augment the mixed initiative system with considerably more problem solving and inferencing capabilities. Koffman's recent article (Koffman and Blount 73) provides a review of the underlying structure and the inherent limitations to some of these systems. SOPHIE reflects a major effort to produce a CAI system that on

---

*This research was supported by the Advanced Technology Training Program of ARPA-HRRO and AFHRL of Lowry AFB under contract F41609-74-C-0015.

**A <u>So</u>phisticated <u>I</u>nstructional <u>E</u>nvironment one hand produces deep logical inferences on a domain less formal than symbolic logic and on the other hand is sufficiently complete that it can answer nearly all questions posed to it. As such it overcomes two of the major limitations inherent in all the above-mentioned systems. However these capabilities are by their very nature complex and as such require a sophisticated set of strategies and procedures. SOPHIE represents approximately 3ØØ,ØØØ words (36 bit) of INTERLISP and FORTRAN code running on a virtual memory TENEX. Although it is an immense program, it is surprisingly efficient exhibiting a typical response delay of three seconds on a lightly loaded system.

Basic Scenario

The basic scenario around which SOPHIE is built is that of a student attempting to isolate a fault in a given piece of electronic equipment while having a lab instructor standing over his shoulder to answer questions, evaluate hypotheses and pose alternatives. SOPHIE not only provides the student with a simulated electronics laboratory, but much more important it provides him with a consultant who will, if asked, give him suggestions and comment on the logical consistency of his measurements and ideas.

In this setting, SOPHIE presents the student with a circuit schematic of the instrument under study and automatically selects and inserts a fault of some specified degree of difficulty. The student then tries to isolate the fault by requesting various measurements under any instrument settings that he desires. At any time he can offer a hypothesis about what he thinks is wrong with the instrument and have the system evaluate his hypothesis. This evaluation reports to the student whether his hypothesis is consistent with what he should have learned from the measurements he has taken. (Of course, the particular set of measurements the student has made is not known to the system ahead of time.) The student can also, at any time, replace any component, but before a part is replaced, the student is queried as to what he thinks is wrong with it. Only if his answers are correct is the component replaced. In those cases where he has only discovered a part which was blown because of a deeper fault, the replaced component will be reblown until he discovers and fixes the fundamental fault. If the student becomes stuck and cannot think of any faults which would explain the measurements he has made, he can ask for help. SOPHIE then looks at his measurements and generates possible hypotheses which he can explore.

In order to illustrate some of SOPHIE's capabilities, we have included in Figure 1 a protocol generated by a student using SOPHIE. The instrument SOPHIE currently models is the Heathkit IP-28 regulated power supply. The IP-28 is a reasonably sophisticated, six transistor power supply with both current and voltage limiting behavior. At the start of the session (1)*, the student indicates that he would like to troubleshoot a broken instrument**, so SOPHIE randomly selects a fault, inserts it into a model of the instrument and tells the student how the controls are currently set. The student takes some measurements (2,3,4,5,6,7) in the broken unit and eventually discovers what he believes to be the fault. Upon trying to replace the part that he thought was bad (8), he learns that it is not faulted the way he thought (9,1Ø,11). Undaunted, he asks whether his hypothesis was reasonable (12,13) and is told exactly why it is not! Unable to think of what to do next, he asks for assistance (14) and is given some possible hypotheses which are consistent with the symptoms he has observed. Exploring these possibilities, he asks a hypothetical question (15), makes more measurements (16,17) and reaches another hypothesis. He is told that this hypothesis is reasonable (18,19) and checks to confirm it (2Ø,21) before fixing the problem (22) and making sure that the instrument is indeed fixed (23,24).

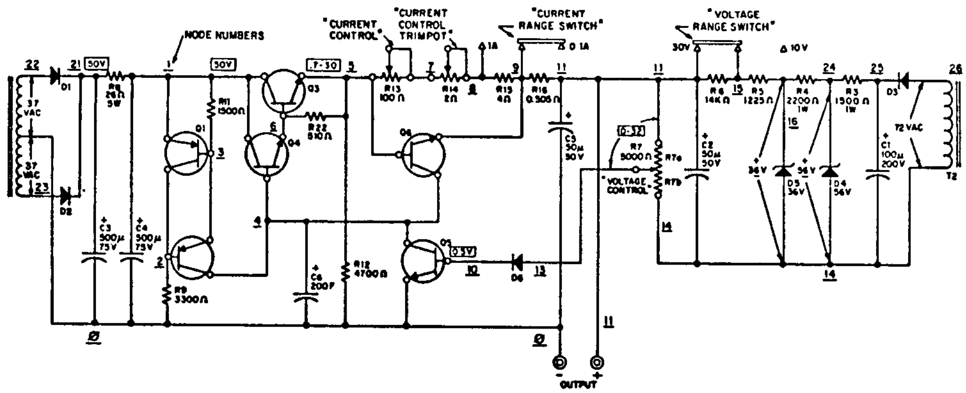
There are several points worth emphasizing about the dialogue in Figure 1. First the student is not restricted to any particular order or number of measurements. This means that SOPHIE must be able to evaluate a student's hypothesis or generate a collection of hypotheses relative to any possible set of measurements in the circuit. Second the student in the dialogue was not a beginning electronics student. While SOPHIE is potentially useful as a simulated laboratory for beginning students, its tutorial feedback assumes that the user has the requisite electronic knowledge expected of someone beginning troubleshooting. Third the dialogue occurred in real time. In the next section, we will discuss the inferencing mechanisms which allow SOPHIE to carry on such a dialogue.

----

*Numbers in parentheses refer to the student's lines in Figure 1.
**SOPHIE can also be used in other modes of operation such as to explore the actions of a working circuit.

FIGURE 1

WELCOME TO SOPHIE, A REACTIVE LEARNING ENVIRONMENT.
THE PIECE OF EQUIPMENT YOU ARE STUDYING IS THE IP-28 REGULATED POWER SUPPLY. WOULD YOU LIKE A CIRCUIT SCHEMATIC?*
> YES **



WOULD YOU LIKE TO START WITH A FAULTED INSTRUMENT WHICH NEEDS FIXING?
1) > YES
THE INSTRUMENT HAS A FAULT AND ITS PRESENT CONTROL SETTINGS ARE:

CC	1.Ø	(CURRENT CONTROL-FROM Ø.Ø (MIN CURRENT) TO 1.Ø)
CR	HIGH	(CURRENT RANGE SWITCH, HIGH=1 AMP, LOW=.1 AMP)
LOAD	2Ø	(LOAD RESISTANCE IN OHMS)
VC	1.Ø	(VOLTAGE CONTROL-FROM Ø.Ø (MIN VOLTAGE) TO 1.Ø)
VR	HIGH	(VOLTAGE RANGE SWITCH, HIGH=3Ø VOLTS, LOW=1Ø VOLTS)

2) >>WHAT IS THE OUTPUT VOLTAGE?
THE OUTPUT VOLTAGE IS 11.7 VOLTS.

3) >>WHAT IS IT IN A WORKING INSTRUMENT?
IN A WORKING CIRCUIT THE OUTPUT VOLTAGE IS 19.9 VOLTS.
THAT VALUE IS INCORRECT.

4) >>WHAT IS THE VOLTAGE BETWEEN R8 AND Q1
THE VOLTAGE AT N1 IS 47.6 VOLTS.

5) >>WHAT IS THE BSE EMITTER VOLTAGE OF THE VOLTAGE LIMITING TRANSITOR?
=BASE
=TRANSISTOR } *spelling correction*
THE BASE EMITTER VOLTAGE OF Q5 IS -.Ø37 MILLIVOLTS.

6) >>WHAT ABOUT FOR THE CURRENT LIMITING TRANSISTOR
THE BASE EMITTER VOLTAGE OF Q6 IS .349 VOLTS.

7) >>WHAT SHOULD IT BE?
IN A WORKING CIRCUIT THE BASE EMITTER VOLTAGE OF Q6 IS .593 VOLTS.

8) >>REPLACE Q6
I AM GOING TO ASK YOU SOME QUESTIONS ABOUT HOW Q6 IS FAULTED.
ARE ANY JUNCTIONS SHORTED?
9) > I DON'T KNOW
ARE ANY TERMINALS OPEN?
10) > YES
WHICH TERMINALS ARE OPEN?
11) > THE BASE
Q6 IS NOT FAULTED IN THAT MANNER. Q6 NOT REPLACED.

----

*The schematic has been photo-reduced to save space and is included here to give an idea of the complexity of the circuit.
**Parts typed by the student are underlined.

```
12)  >>IS IT POSSIBLE THAT Q6 HAS AN OPEN BASE
     IS THE COMPLETE FAULT MODE EB/EC/BC = OP/OK/OP WHAT YOU INTENDED?
13)  > SURE
     WHEN THE SETTINGS WERE: CC=1.Ø, CR=HIGH, LOAD=2Ø, VC=1.Ø, VR=HIGH
     THAT FAULT CONFLICTS WITH THE FOLLOWING MEASUREMENT.
     YOU OBSERVED THAT THE OUTPUT VOLTAGE WAS 11.7 VOLTS.
     IF Q6 HAD FAULT MODE EB/EC/BC = OP/OK/OP IT WOULD BE 23.9 VOLTS.
     IN A WORKING CIRCUIT IT IS 19.9 VOLTS.

     ALSO:
     THAT FAULT CONFLICTS WITH THE FOLLOWING MEASUREMENT.
     YOU OBSERVED THAT THE BASE EMITTER VOLTAGE OF Q6 WAS .349 VOLTS.
     IF Q6 HAD FAULT MODE EB/EC/BC = OP/OK/OP IT WOULD BE .712 VOLTS.
     IN A WORKING CIRCUIT IT IS .593 VOLTS.

     THAT FAULT DOES NOT EXPLAIN ANY OF THE SYMPTOMS YOU OBSERVED BUT IT IS CONSISTENT
     WITH THE VALUES FOR THE BASE EMITTER VOLTAGE OF Q5 AND THE VOLTAGE AT N1.

14)  >>WHAT COULD BE WRONG?
     HERE ARE SOME FAULTS WHICH EXPLAIN THE MEASUREMENTS YOU OBSERVED:
     THE BETA OF Q3 IS LOW
     THE BETA OF Q4 IS LOW
     R22 HAS THE WRONG RESISTANCE
     OR THE CONSTANT CURRENT SOURCE IS ANEMIC

15)  >>WHAT WOULD THE OUTPUT VOLTAGE BE IF THE BETA OF Q3 WERE 8?
     IF THE BETA OF Q3 WERE 8
     UNDER THE PRESENT SETTINGS, THE OUTPUT VOLTAGE WOULD BE 1Ø.5 VOLTS.
     IN A WORKING CIRCUIT IT IS 19.9 VOLTS.

16)  >>IS THE EMITTER CURRENT OF Q4 CORRECT
     THE EMITTER CURRENT OF Q4 IS -61 MILLIAMPS.
     IN A WORKING CIRCUIT THE EMITTER CURRENT OF Q4 IS -49 MILLIAMPS.
     THAT VALUE SEEMS TO BE INCORRECT.

17)  >>WHAT IS THE CURRENT THRU R22
     THE CURRENT THRU R22 IS 35 MILLIAMPS

18)  >>COULD R22 BE LOW
     IS IT OKAY IF I USE 25 OHMS FOR THE RESISTANCE OF R22?
19)  > YES
     THAT FAULT AGREES WITH ALL OF THE MEASUREMENTS YOU HAVE MADE.

20)  >>WHAT ARE THE SPECS OF R22
     THE SPECIFICATIONS OF R22 ARE:
     VALUE -- 51Ø OHMS
     POWER RATING -- .5 WATTS

21)  >>WHAT IS THE RESISTANCE OF R22
     THE MEASURED RESISTANCE OF R22 IS 25 OHMS.

22)  >>REPLACE R22
     R22 REPLACED
     NOW YOU SHOULD TAKE MEASUREMENTS TO SEE IF THERE IS ANYTHING ELSE WRONG.

23)  >>NOW  WHAT IS THE OUTPUT
     THE OUTPUT VOLTAGE IS 19.9 VOLTS.

24)  >>WITH A 1ØØØ OHMS
     WITH THE LOAD NOW SET TO 1ØØØ
     THE OUTPUT VOLTAGE IS 3Ø VOLTS.

     >>IS ANYTHING ELSE WRONG?
     NO
```

Endowing SOPHIE with Some Intelligence

SOPHIE manifests most of its "intelligence" through its question answering and hypothesis evaluation and generation abilities. These abilities are achieved through a set of special purpose inferencing procedures each of which performs a certain class of inferences extremely efficiently. The centralizing component of the inferencing system is a simulation program which models a "piece of knowledge" which in this case is an electronic instrument*. The underlying idea of how simulation can be used to perform inferencing is both straight-forward and extremely powerful. Let us first consider the problem of answering a question (always with respect to a given circuit) of the form:

"If X then Y?"

where X is a proposition about some component in the given instrument and Y is a proposition about its behavior or symptoms. An example of such a hypothesis might be:

"If C2 is shorted, is the output voltage zero?"

The answer to the question can be found by invoking the simulator: First the simulation model of the instrument must be modified so that C2 is shorted (i.e. the proposition X must be made true on the model). Then the simulation of the modified model is executed. Since the results of the simulation run contain all the consequences of the modification (C2 being shorted), the hypothetical consequent (the output voltage being zero) is simply checked against these simulation results.

The above paradigm glosses over several logical difficulties concerning which boundary and/or input conditions should be used for the simulation runs. If it is necessary to determine all the logically possible consequences of a hypothetical modification, then the simulation must, in principle, be run over a potentially infinite collection of the instrument's control settings, etc. While for most practical situations there are only a finite number of cases "worth" considering, this number can still be quite large. It is clearly necessary to have an additional inferencing mechanism which can determine what the worthwhile cases are for any particular question. This additional mechanism must embody electronic knowledge of a different sort than is represented in the simulator. Thus, metaphorically, the simulator may be interpreted as creating examples whereas this additional mechanism tries to guarantee that these examples will be useful.

The inferences that fit most simply into the simulation paradigm concern requests for measurements. It is through this mechanism that SOPHIE can create the laboratory environment within which the student is working. Whenever the student requests a measurement, the simulation is called to compute the voltage at every node in the circuit. From the voltages, procedural specialists derive answers to additional questions about the current through any component, the resistance of a component, the power dissipation of a component, the beta of a transistor, etc. Whenever the student wishes to explore the circuit under different conditions, he can change the controls which is reflected in a corresponding change to the simulation model.

Hypothesis Evaluation:

The first non-obvious use of simulation concerns the task of hypothesis evaluation. Remember that hypothesis evaluation requires a technique that can check the logical consistency of a hypothesis against the measurements the student has taken. For example, hypothesis evaluation is required when a student, after making several measurements, develops an idea (hypothesis) about what is wrong, e.g. "Is it possible that Resistor R9 is open". The question at this point is "does the idea which the student has about what's wrong, conflict with any of the measurements he has observed?" If so, these discrepancies must be pointed out to the student as logical inconsistencies in his hypothesis.

The evaluation strategy makes extensive use of simulation. First, the simulation model is modified to reflect the given hypothesis, i.e. the fault hypothesized by the student is inserted into the model. Then all the student's measurements are repeated under this "hypothetical" model. For each measurement there are four cases that might occur. (1) The observed and hypothetical values may agree. (2) The observed value may represent a symptom (i.e. be wrong), while the hypothetical value is normal. In this case the fault proposed by the student does not account for this particular symptom. (3) The

---

*More precisely, it models a schema of electronic instruments with one element of the schema being the working instrument and the other elements representing various ways the instrument can be faulted.

observed value may be normal while the hypothetical value is wrong. In this case the proposed fault would have created symptoms which the student did not observe. Or (4) the observed value and the hypothetical value may both be symptomatic but not the same. In every case but the first, the student must be told how the measurements disagree. The student's hypothesis is consistent if all of his measurements fall into case (1).

The comparisons needed to separate the above cases require knowing not only the values of a measurement in the hypothetical and malfunctioning circuits but also the value in a working circuit as well. The value in a working circuit is used to determine when the other two values differ significantly. For example, if the value that the student observed was 25 volts and the value under the hypothesized fault was 30 volts the difference between these two may or may not be significant. If the value in the working circuit is 30 volts, the proposed fault does not account for the lower voltage observed in the faulted circuit. However, if the working circuit voltage is 3 volts, the hypothesis is doing a pretty good job of explaining the behavior implied by this measurement. Therefore, in addition to using simulation to determine the above quantities, a metric is involved to "judge" the qualitative distance between these values. (See (Brown et al 74) for a complete specification of this process.)

Hypothesis Generation:

One of the most difficult tasks performed by SOPHIE is determining the set of possible faults that are consistent with the observed behavior of the broken instrument. At any time during his troubleshooting session, the student can ask for help. SOPHIE must then generate the set of hypotheses which would explain the behavior exhibited by the particular set of measurements the student has thus far made. The method of generating this set of hypotheses (possible faults) combines simulation with both backward and forward types of reasoning.

First, a backward working specialist examines an output voltage measurement (taken by the student) and generates a list of possible hypotheses that "vaguely" explain that measurement. Each hypothesis so generated is evaluated by a forward working specialist who invokes a simulation of that hypothesis to see if it really accounts for all the known output voltages and internal measurements.

In some cases the backward specialist generates as a hypothesis a fault schema, i.e. a "fault" that has some unspecified parameters. The hypothesis "the beta of the Darlington amplifier (of the IP-28) is low" is an example of a fault schema as is the hypothesis "C2 is leaky". Rejecting such hypotheses requires some subtle reasoning: given two measurements it is possible that each measurement, by itself, could be explained by instantiating a fault schema to a particular value. However, it may turn out that the instantiations required by each measurement are in fact different and hence this fault schema cannot explain both measurements simultaneously. Although a sophisticated forward deduction system might be able to detect this inconsistency, we eventually settled on using simulation in conjunction with a procedural specialist who tries to find an instantiation value of the fault schema by "intelligently" manipulating the simulation model (i.e. using successive refinements).

The exact values for these fault schemas can only be found if at least one output voltage measurement is made in which the voltage is not the correct value for the settings. However, even when a measurement is correct (in the sense of being the same value as would be found in the working circuit), it is possible to determine a range of values for these schema. For example, SOPHIE has a specialist who can determine the range of values for the beta of the Darlington (in the IP-28) which could account for the observed output current. By successively refining this range, it is sometimes possible to rule out certain faults. These specialists do not use simulation but instead have enough built-in intelligence to be able to deduce these ranges for any of the fault schemas that arise in this context.

Simulation Models

As we have seen, DC simulation models form part of the basis of SOPHIE's understanding of electronics. There are currently two very different types of models in use. One is a general purpose circuit simulation package, called SPICE (Nagel 71, 73), which accepts a description of an arbitrary circuit and produces exact quantitative results in both working and faulted versions of the circuit. While we were fortunate enough to be able to borrow the bulk of the simulation package, there were many problems unique to our use of simulation. Methods of modeling a circuit which facilitate the insertion of faults had to be developed along with explicit models of faulty components. In addition, the faulting of one component will very often overload one or more other components leading to fault propagation. This required a special monitoring mechanism which "sits on top" of the general purpose simulator and looks at the results of the simulation to decide if and how additional parts would blow. In fact, this mechanism, by making successive calls to the simulator, grows a fault propagation tree which captures the chain of events of several parts being blown by one initial fault.

The second simulation system is a circuit-dependent, functional simulator which runs several orders of magnitude faster than the general purpose one. This system incorporates much specialized knowledge about the internal functioning of the IP-28 instrument and is used by the hypothesis generation specialist which can need the results from dozens of simulations. This simulator is the only section of SOPHIE which would require extensive rewriting to capture the DC properties of new circuits.

SOPHIE's Linguistic Capabilities

Since the student using SOPHIE is engaged in a problem solving task, it is imperative that he be able to request measurements and state hypotheses in a reasonable subset of English. If he is constrained by the way in which he expresses his ideas, he may spend his energy searching for ways of saying or asking something, rather than getting involved in solving the problem. On the other hand, the student must not wait all day for his question to parse. While keyword analysis schemes have tremendous speed potential, they also have serious limitations and in many cases can become more cumbersome and inefficient than a well designed structural parser. For example, although a keyword analysis might suffice in decoding the utterance:

"What is the output voltage?"

such a technique begins to get messy when the objects being referred to are further distinguished by modifiers. Consider the problem of distinguishing the following two questions:

"What is the output voltage of the power reference transformer?"
"What is the output power of the voltage reference transformer?"

If the modifiers can in themselves be modified, the situation quickly grows out of hand for even the most advanced ad hoc keyword system. Consider for example the question:

"What happens to the voltage between the anode of D6 and the collector of the voltage limiting transistor when the emitter of the current limiting transistor opens?"

After studying numerous protocols of students using a mocked-up version of SOPHIE, we noticed that powerful constraints existed in the relationships between the various semantic/conceptual entities making up a question. For example, if one asks for a voltage measurement it is either between two nodes, across a particular component, or across the output terminals. It seemed that this high degree of semantic predictability could be utilized by a predictive analyzer ("parser") by refining the usual syntactic categories such as noun phrase into relevant semantic/conceptual categories such as "measurement". In general, such refinements could lead to a phenomenal proliferation of categories to be captured by the "grammar", but an analysis of our data indicated that such an approach was feasible. These and other considerations led us to build a highly efficient top-down (goal-oriented), context free, fuzzy parser which makes its predictions on the basis of semantic rather than syntactic categories*.

While we were aware of the potential limitations of such an approach, the parsing systems required to handle more complex grammars are significantly more complex and much slower.** In order to study how easily users could adapt to the imposed linguistic limitations, we have collected well over a hundred hours of protocols of people using SOPHIE over the ARPA network. Each user had seen a prologue which gave him some idea of the system's linguistic and logical capabilities. Any time SOPHIE encountered a sentence which it could not parse, that sentence was automatically stored on a file which was continually used to provide data for expanding our grammar. A point has now been reached in which SOPHIE handles nearly all sentences generated by users

---

*See (Burton 74) for a complete description of the parser and its capabilities.

**A typical 12 to 15 word statement is parsed by our system in about 1ØØ milliseconds.

who first read the prologue.

These experiences convinced us that for our highly constrained domain our approach to parsing is viable. Although the handling of paraphrases allowed by more complex parsing systems would surely have helped SOPHIE appear more natural, three other issues seemed at least as important. The first was accepting an extensive number of abbreviations such as "VBE" standing for "base emitter voltage". The second was the need for spelling correction and separation of run-on words (e.g. "whatis"), which can greatly reduce the amount of concentration, and hence effort, that a poor typist expends in addressing the system.* The third concerned the issue of handling context dependent anaphoric references and ellipses, e.g. pronoun references referring to a prior sentence. Having this capability appears to be especially crucial when one has become totally immersed in using the system as opposed to simply trying it out. An example will illustrate how natural it is to use anaphors and how complex the problem of handling such entities can be:

> What is the current through the base of Q6?
>
> What is it through the emitter? ("It" refers to "current" and "Q6" is implied but not mentioned.)
>
> What about through the collector? (In this case, "current" and "Q6" are both implied but neither is mentioned.)

Our semantically based grammar has enabled us to include a powerful context facility which can handle the pronoun references and ellipses such as exist in the above examples. See (Burton 74) for a further description.

Possible Uses of SOPHIE

Although SOPHIE may be viewed as a stand alone CAI system, we prefer to view it as a set of powerful tools with which to implement and experiment with various teaching strategies. One can also imagine this system being used in conjunction with an efficient frame-oriented system which would be responsible for presenting textual material about the operational principles of a given instrument, etc. Once this material had been mastered, problems could be presented to the student for which the unique power of SOPHIE would best be suited. With the ARPA computer network, a combination of resources is now feasible where a computer system optimized for frame-oriented CAI could "automatically" invoke another computer system optimized to handle an "intelligent" CAI problem solving session.

Outside the domain of frame-oriented CAI are other exciting possibilities for using SOPHIE. One particular use currently under consideration involves using SOPHIE in a gaming situation. For example, after students are exposed to the fundamentals of how a given circuit operates, they would participate in a two-person game. One student introduces a subtle fault into the circuit and is scored on how well he predicts the consequences of his modification. The other student must then discover the modification by performing a series of measurements. Each measurement has a cost* and the total cost is computed for his sequence of measurements. After the fault is isolated the roles are reversed and the game is played again.

Although the gaming scenario may seem of primary relevance for diagnostic training, it is of far greater importance in providing the student with an intuitive understanding of the qualitative and causal behavior of system components. In fact, one of the best ways of discovering the purpose for a particular component is to "alter" or remove that component and see what aspects of the circuit's behavior change.

Conclusion

SOPHIE has demonstrated that practical uses of AI techniques exist for CAI systems. In light of the rapid development of powerful, fast mini-computers which are capable of supporting systems as complex as SOPHIE, we believe that this kind of research should help expand our view about innovative uses of technology in CAI. It is not difficult to see that many of the ideas underlying SOPHIE could also be used in helping an author prepare lessons. Research along these lines plus the expansion of the logical (e.g. providing causal explanations, checking the validity of measurements etc.) and linguistic capabilities of SOPHIE are currently in progress.

---

*The language processing system uses the spelling correction routines provided by the INTERLISP DWIM facility (Teitelman 74).

*The cost could be varied to encourage students to learn different methods of troubleshooting but would usually reflect the difficulty involved in making the measurement in a real electronics laboratory, i.e. external measurements are the cheapest while ones requiring the removal of a component from the circuit are expensive.

ACKNOWLEDGEMENTS

We would like to thank Alan G. Bell for his extensive work on the simulation module and other parts of SOPHIE and Edward Gardner for his many constructive suggestions and patient exercising of earlier versions of SOPHIE.

REFERENCES

Brown, J.S., Burton, R.R. and Zdybel, F. "A Model-Driven Question Answering System for Mixed-Initiative Computer Assisted Instruction", IEEE Transactions on Systems, Man and Cybernetics, Vol. SMC-3, May 1973.

Brown, J.S., Burton, R.R. and Bell, A.G. "SOPHIE: A Sophisticated Instructional Environment for Teaching Electronic Troubleshooting (An Example of AI in CAI)" BBN Report No. 279Ø, Bolt Beranek and Newman, Inc. Cambridge, Mass. March 1974.

Burton, R.R. "A Semantically Centered Parsing Strategy for Mixed-Initiative CAI Systems" Paper presented at the 12 Annual Meeting of the Association for Computational Linguistics, Amherst, Mass., July, 1974.

Carbonell, J.R. "AI in CAI: An Artificial Intelligence Approach to Computer-Assisted Instruction," IEEE Transactions on Man-Machine Systems, Vol. MMS-II, No. 4, December 197Ø, pp. 19Ø-2Ø2.

Carbonell, J.R. and Collins, A.M. "Natural Semantics in Artificial Intelligence," Proceedings of Third IJCAI, SRI Publications Department, Menlo Park, California, August 1973, pp. 344-351.

Goldberg, A. "Computer-Assisted Instruction: The Application of Theorem-Proving to Adaptive Response Analysis," Technical Report No. 2Ø3, Institute for Mathematical Studies in the Social Sciences, Stanford University, Stanford, California, May 1973.

Koffman, E.B. and Blount, S.E. "Artificial Intelligence and Automatic Programming in CAI," Third International Joint Conference on Artificial Intelligence, Stanford University, Stanford, California, August 1973.

Nagel, L.W. "Transient Analysis of Large Non-Linear Networks," IEEE Journal of Solid State Circuits, Vol. SC-6, August 1971.

Nagel, L.W. and Pederson, D.O. "SPICE: Simulation Program with Integrated Circuit Emphasis," Memorandum ERL-M382, Electronics Research Laboratory, College of Engineering, University of California at Berkeley, April 1973.

Sleeman, D.H. "A Problem Solving Monitor for a Deductive Reasoning Task," International Journal of Man-Machine Studies. (In press.)

Teitelman, W. INTERLISP Reference Manual, XEROX PARC, Palo Alto, California, February 1974.