DOCUMENT RESUME

ED 034 410 | 56c | EM 007 572

AUTHOR Roy, Rob
TITLE Computer Aided Instruction for a Course in Boolean Algebra and Logic Design. Final Report (Revised).
INSTITUTION Rensselaer Polytechnic Inst., Troy, N.Y.
SPONS AGENCY Office of Education (DHEW), Washington, D.C. Bureau of Research.
BUREAU NO BR-5-1081
PUB DATE Aug 68
GRANT OEG-1-6-051081-0660
NOTE 21p.

EDRS PRICE EDRS Price MF-$0.25 HC-$1.15
DESCRIPTORS Academic Performance, Achievement Gains, *Computer Assisted Instruction, Computers, *Concept Teaching, *Cost Effectiveness, Feedback, Indexes (Locaters), Input Output, Instructional Design, Machine Translation, Man Machine Systems, Mathematical Logic, *Multimedia Instruction, *Programed Instruction, Systems Analysis, Time Factors (Learning)
IDENTIFIERS Rensselaer Polytechnic Institute

ABSTRACT

The use of computers to prepare deficient college and graduate students for courses that build upon previously acquired information would solve the growing problem of professors who must spend up to one third of their class time in review of material. But examination of students who were taught Boolean Algebra and Logic Design by means of Computer Assisted Instruction (CAI), in comparison with students who had the conventional classroom instruction, showed that the control group's retention of conceptual material was markedly superior. Further drawbacks to CAI are the enormous effort required to prepare even the simplest course material, the large cost of both hardware and software, extremely difficult man-machine communication, poor software support from IBM, the inability of the machine to judge partially correct answers and to ascertain whether the student has successfully grasped the concept involved, decreased student concentration span, inefficient use of student time, the lack of an indexing system to compensate for the forward-structured system, and the undue complexity which requires a student to work with several media simultaneously. The conclusion that CAI is unsuitable for engineering education is narrow due to experimental conditions. (MM)

FINAL REPORT (REVISED)
Project No. 5-1081
Grant No. OEG-1-6-051081-0660

COMPUTER AIDED INSTRUCTION FOR A COURSE IN BOOLEAN ALGEBRA AND LOGIC DESIGN

August 1968

U.S. DPEARTMENT OF
HEALTH, EDUCATION, AND WELFARE

Office of Education
Bureau of Research

COMPUTER AIDED INSTRUCTION FOR A COURSE IN BOOLEAN ALGEBRA AND LOGIC DESIGN

Project No. 5-1080
Grant No. OEG-1-6-051081-0660

Dr. Rob Roy

August 1968

The research reported herein was performed pursuant to a grant with the Office of Education, U. S. Department of Health, Education, and Welfare. Contractors undertaking such projects under Government sponsorship are encouraged to express freely their professional judgment in the conduct of the project. Points of view or opinions stated do not, therefore, necessarily represent official Office of Education position or policy.

Rensselaer Polytechnic Institute

Troy, New York

Introduction

As the fund of knowledge increases, increasing demands are placed upon undergraduate and graduate instruction. As a result, topics and courses that were primarily the concern of a graduate school now fall into the curriculum of the undergraduate school. Consequently, certain prerequisites for a graduate course are assumed to have been obtained in a student's undergraduate work. This may not be the case, especially where there are foreign or transfer students entering into a graduate school. Often, the professor must spend up to one third of a course reviewing material that is essential for the new material of the course. The current alternatives to this approach are one, the student is denied admission to the course until he has completed the prerequisites. This solution is often impractical, due to financial or fellowship arrangements that the student has made. Second, the student is admitted to the course with the provision that he learn the background material on his own time. Without any guidance in this direction, the student soon falls behind in the new material and eventually must drop the course.

This problem has reached such proportions that Rensselaer Polytechnic Institute has decided to take a serious look at Computer Assisted Instruction, in the hope that this approach may solve this problem.

The field of computer aided instruction is relatively new. Much of the work has been devoted to either the physical hardware required to implement this instruction, or the programming of courses which require learning of a rote nature. Although tests have been made using control groups learning the same material under standard classroom conditions, there has been no examination of the success of computer-taught students in courses which relied upon the material learned. This project proposed the examination of this aspect of computer instruction.

The research which is most closely related to the work which will be performed is that which was reported by Grubb and Selfridge.[15] This pilot study was concerned with the computer tutoring of statistics. The primary development of this study is the Coursewriter language. This pseudo-language permits the instructor to easily write and modify computer programs for a given course.

The objective of this pilot study is to obtain sufficient evidence to support an administrative decision on the use of computer aided instruction. In particular, Rensselaer Polytechnic Institute is interested in the use of computer aided instruction for entering students, both entering freshman and entering graduate students. This instruction would provide these students with the prerequisites necessary to continue their curriculum.

This pilot study would be devoted to the subject of Boolean Algebra and Logic Design. This course is a prerequisite for the graduate sequences in the fields of Digital computers, Automata, and Learning machines. If computer aided instruction is proven successful in a course of this type, then a program of greater scope will be undertaken.

Historical Review

At the request of IBM, a demonstration of Computer Assisted Instruction was held at Thomas J. Watson Research Center, Yorktown Heights, N. Y. on December 21, 1964. At this demonstration the CAI program for the IBM 1401-1440 computer was described, as well as the original work which was developed on the IBM 650 computer.

After a report of this demonstration was made to the administration of R.P.I., an interest was displayed as to the feasibility of CAI for our Institute. At this point an investigation of the various avenues of support suggested by IBM was made and a proposal written to HEW. Before writing this proposal IBM had promised to supply the additional equipment for our IBM 650 computer as well as the necessary software. Thus it appeared as if our only problem would be writing the suitable CAI program.

Consequently the proposal was approved in October 1965 and finally granted 15 April 1966. During the period from October 1965 to July 1966 it was assumed that the IBM equipment was imminent.

Operating under the assumption that the necessary equipment additions were to arrive, a great deal of planning was undertaken to determine how the equipment would interface. Coupled with the equipment interface problem was a necessity for thorough understanding of the software required. This effort took us into July 1966. At this point all the required electrical connections were ready, as well as the software knowledge. However it became evident, although never explicitedly stated, that the promised equipment would never arrive.

Consequently we decided to use our newly installed IBM 360 Mod. 50 as the central computer, coupled with a remote station. Since IBM had a CAI program underway at Poughkeepsie, N. Y. they promised to provide the required software for this system. Our efforts then turned to obtaining the remote terminal and the required software for the terminal.

During this period we continued to write the program for the course in Boolean Algebra and Logic Design, outlining the desired topics and determining the areas of maximum student difficulty.

By October 1966, after direct contact with the CAI group in Poughkeepsie, it was clear that IBM would not have their 360 version of CAI prepared in time for this grant. The group in Poughkeepsie did have a working version, but they would not release it as they felt it was not ready for general use.

Consequently we decided to write a single terminal compiler for the 360 which would emulate the 1410-1440 CAI system. This would enable us to use the present CAI compiler. Under this arrangement the 360 would appear as if it was a 1410-1440 system. A letter was written on November 28, 1966 to Dr. Kenneth Brown of HEW requesting an extension of the April 15, 1967 deadline to September 15, 1967. This extension would enable us to write the compiler and properly test the CAI program which had been written. The request for an extension was granted.

In addition a group of 20 undergraduate students were tested using the programmed text "Boolean Algebra and Logic Design" written by Hoernes and Heilweil, and published by McGraw-Hill Book Co. The results of this test were highly encouraging. The students were able to rapidly comprehend the material and appeared to enjoy working with the programmed text. Although there was no noticeable difference in performance with this group as opposed to the conventional class group, there was a marked improvement in learning time. This test with programmed text gave rise to a projection of significant improvement or at least equal performance using CAI.

The 1410-1440 emulator was completed by 15 January 1967 and a checkout phase was begun.

The checkout phase was completed by 15 March 1967. During this period negotiations with IBM concerning the installation of remote terminals continued, and we were assured that the proper software would be available when required. Unfortunately, although the 360 was initially equipped with the necessary hardware for remote terminals the software was not completed. Consequently without time sharing facilities, our emulator could not be utilized except in a highly impractical manner which would not establish the proper man-machine communication.

After considerable additional negotiations with IBM, time was made available (for purchase) on the IBM 7010-1440 experimental CAI system at the Thomas J. Watson Research Center. This negotiation had been held in conjunction with our emulator design, with the thought that one of the two possibilities would prove fruitful. When the time sharing software for the 360 was not available, it appeared that the time shared setup with the 7010-1440 was the only alternative. The contract specified that 4 hours/day of CAI time would be made available between March 1, 1967 and April 30, 1967. Although this arrangement was a minimal one for R.P.I. in terms of time and student availability, it was the only possible solution to an already impossible situation.

However IBM failed to deliver the system on the scheduled dates, and the agreement had to be rewritten for the period May 1, 1967 to June 30, 1967. This latter arrangement was less than satisfactory since it meant that we could purchase CAI time (paying for computer time, terminal rental, and telephone charges) only during the period when there would be no undergraduate students available to provide testing of the material. Indeed, the limited time available precluded any testing and debugging of our original complete course material. Consequently we were faced with a situation where only a limited set of course topics could be checked and tested by a small group of graduate students who were available. However, since this was the only possible means of obtaining any of our initial goals, we proceeded along this path.

During the period when CAI time was available there were other minor difficulties among which were a further two week delay in equipment delivery and a period of two weeks in the middle of our project when IBM "borrowed" our terminal for display to a prospective customer. These events made it difficult to maintain our group of volunteer graduate students.

However as the remainder of this report indicates, a limited set of goals were achieved and the project was completed in January 1968 when we tested the original group of students for retention of both concept and factual material.

Method

Proposed Procedure: The first step in this study is to prepare a text for the course. The programmed text "Introduction to Boolean Algebra and Logic Design" by Hoernes and Heilweil[17] (both of IBM) will serve as the framework for the course. Additions to this text will be made by programming certain portions of

"Logic Design of Transistor Digital Computers" by Maley and Earle (both of IBM). The COURSEWRITER Language developed at the IBM Watson Research Center will be used.

The class size in the technical elective Logic Design I and Logic Design II runs from 75 to 100 students, being mixed with both graduate students and seniors. From this population, a sample size of 40 students will be drawn. This sample will be divided into three categories.

1. 10 newly entered graduate students - these students will take the classroom courses Logic Design I and Logic Design II in the normal sequence.

2. 10 newly entered graduate students - these students will take both the computer aided course and the classroom course Logic Design II simultaneously.

3. 10 seniors - chosen to represent the spectrum of students who have registered for the course. These students will take both the computer aided course and the classroom course Logic Design II simultaneously.

4. 10 seniors - chosen to represent the spectrum of students who have registered for the course. These students will take both the classroom courses Logic Design I and Logic Design II in the normal sequence.

Thus, twenty students will be used to take the computer aided course. Their performance will be examined in several different ways.

1. A comprehensive examination covering Logic Design I will be given to all 40 students.

2. Their relative performances in Logic Design II will be examined.

In addition, student comments and their evaluation of the computer aided instruction will be sought. Standard statistical methods will be applied to all quantitative measurements.

Actual Procedure: Due to the extreme limitations which were placed on this investigation by the lack of CAI software and computer time, a curtailed version of the proposed method was employed. During the period of CAI availability a group of ten graduate students were available, and these students were extensively used to evaluate user response of this system. The system used consisted of a Western Electric 103A data set and an IBM 1052 remote terminal equipped with a slide projector and a tape recorder. Communication to the Watson Research Center was made via the data set.

The following is an outline of the actual topics covered.

I. Introduction to CAI

a. 1052 terminal
b. Procedures: sign on, sign off, etc.
c. Conventions: capitalization, underlining, etc.

II. Number Systems

a. General number system
b. Base 10 system
c. Binary system
d. Conversion between different bases
e. Coding: BCD, Gray, Biquinary, etc.

III. Binary Arithmetic

a. Addition
b. Subtraction
c. Multiplication
d. Division

IV. Logical Statements

a. And - Intersection
b. Or - Union
c. Negation
d. Venn Diagrams

V. Truth Tables

VI. Axioms of Boolean Algebra (DeMorgans Theorem, etc.)

VII. Minterms and Maxterms

VIII. Minimization

a. Using axioms of Boolean Algebra
b. Using Karnaugh map

Each of the students took the entire course over a one month period. During this period they were tested three times. The tests were designed to test rote learning, conceptual understanding, creativeness, and ingenuity. The results of these tests were correlated with a similar group of students who took the same tests but took the classroom course. After a period of six months this same group of students were retested for retentivity of the course material.

Results

The results of this study are largely negative. After a considerable amount of program writing and revision, this author has become disillusioned with the use of CAI for engineering courses at the senior level. Actually, the implication is far broader than this, but due to the limitations of our testing the following remarks will be restricted to the narrow area stated. Briefly the reasons behind this negative reaction are:

1. The enormous effort required to prepare even the simplest course material.

2. Coupled with (1) is the large cost in both providing the terminal equipment (hardware) and the course program and associated slides, tape, manuals, etc. (software).

3. Extremely difficult man-machine communication. The standard 1052 keyboard is simply not capable of transmitting the desired information in a suitable form.

4. Poor software support from IBM. There appeared to be a definite lack of interest in CAI on the part of IBM.

5. The (correct/incorrect) structure of student answers. Using this system it is difficult to judge partially correct answers or ascertain whether the student has successfully grasped the concept involved.

6. Decreased student concentration span. The students found it difficult to sit at the keyboard for long lengths of time. Although the material was interesting, the students found CAI to be boring.

7. Inefficient use of student time, partly due to the amount of time required to type in mathematical answers and

partly due to the inherent choppiness of CAI. Most of the students preferred the free flow of a textbook for the comprehension of conceptual material. Consequently there was student impatience in many areas of the course material.

8. The system is forward-structured, preventing the student from looking back to refresh his memory. This is a serious drawback, as the student who forgets a formula must refer to a textbook or guess the answer. An indexing system is required, with additional software and hardware (multiple visual display) support.

9. Since this is computer assisted instruction, other media must be used. This requires the student to work in several media nearly simultaneously. Consequently the use of the computer display, hard copy, and pencil and paper scratch pad often constitutes an unduly complex system.

10. Tests of the students ability to retain conceptual material show CAI in a poor light. Although it may have been the fault of the computer program, conceptual material was not retained too well. The student group which had the conventional classroom instruction showed markedly superior conceptual retentivity than the CAI alone group. These tests were performed six months after the completion of instruction.

Discussion

The results of this study indicate that CAI remains to be developed to a higher degree of sophistication before it can be considered to be useful in a concept oriented engineering course. The difficulties arise in two principal area;

1. Man-machine communication

2. Effort required to provide an adequate course of instruction.

The first difficulty can be partially overcome by redesigning the 1052 keyboard and the inclusion of additional input-output facilities. A redesigned keyboard would have not only mathematical symbols but 1/2 line spacing so that super and subscripts could be employed. One additional input-output facility which would be desirable is a character console by which the student could recall past formulae and facts. The basic terminal devices are not adequate for course presentation. A recent NASA report (CR-917) on a feasibility study for CAI lists the following requirements for the terminals:

"The display system must be capable of displaying textual information, diagrammatic information, and graphic representations. An analysis of the types of displays indicates that the display system must provide:

A. A repertoire of 128 characters, symbols and numerals with two character sizes.

B. 1000 to 1200 displayable characters when in a textual or typewriter mode of operation.

C. Approximately 600 randomly plotted points or symbols when in the random mode of operation.

D. Approximately 20 full-length vectors and 300 to 400 randomly plotted vectors capable of extending up to a quarter length of the display surface. Continuous point vector generation is also required.

E. A mixed mode of operation in which vectors, characters and symbols are used for display construction.

The input device must have the following capability:

1. Character entry to enable English sentence construction.

2. Numberic entry to enable operation with computationally oriented languages.

3. Symbolic inputs to enable the student to identify unique functions related to various disciplines."

The second primary disadvantage, that of the enormous programming effort to produce a CAI program appears to be inherent in this type of system. The author must account for or forsee almost every type of difficulty that the student will encounter, from conceptual difficulties to spelling errors. This latter point causes a large part of the increased effort. A computer program will check each answer on a true or false basis. Consequently a large number of partially correct answers have to be listed as a true answer. Even more perplexing than simple spelling or typing errors is the programming involved when a decision has to be made as to whether the error is simply an error in typing or a conceptual error. For example, if the student typed in an incorrect sign, there are several ways in which this error can be handled, and is partially dependent upon the students previous performance. Thus, at each point where the student must make a response there is buried beneath the correct answer a multitude of incorrect, partially correct, or ambiguous answers, each of which must be handled by a responsible and informative reply.

It is interesting to note that a recent report on CAI in Engineering Education had similar results, as a result of which the CAI approach has been discontinued at the U. S. Naval Postgraduate School.

The students who participated in this experiment were asked for written comments concerning this system of education. One of the students written comments appears as an appendix.

Conclusions

The conclusion reached by this study is that CAI (in its present form) is unsuitable for engineering education at the senior-graduate school level. A number of the present objections can be removed by additional hardware. However this additional hardware (display terminals) would inflate the already prohibitive cost of CAI. Even if the physical conditions were ideal, there is an enormous effort required to produce an hour of CAI program, roughly 100 hours for each hour of program. In light of the rapidly changing course structure of an engineering curriculum it appears that utilization of CAI would not be practical.

Summary

This study was an investigation of Computer Assisted Instruction for use in a concept-oriented engineering curriculum. Principally, the thought was to utilize CAI for those students who lacked certain undergraduate prerequisites for first year graduate courses. Consequently, the course "Boolean Algebra and Logic Design" was programmed and tested on a small group of students. This effort supplied both student comments and author experience from which a judgement was made as to the effectiveness and utility of CAI.

The results of this study was that CAI is not sufficiently developed from either a hardware or software point of view for use in senior year engineering courses. The hardware required for adequate man-machine communication could be obtained, but the software requirements place a limiting cost on any particular endeavor, which greatly diminishes the effectiveness of CAI. Consequently, it was felt that the effort and money which would be required for an adequate set of CAI programs could best be utilized in more direct teaching formats.

References

1. Balabanian, Norman. "The Educational Engineering Called Programmed Instruction," IEEE Transactions on Education. Vol. E-9, No. 2, June 1966. pp. 49-57.

   This article is primarily concerned with programed tests. The author comments that the reasons for using a machine are not compelling. An adaptive machine with imaginative displays that include visual and audio stimuli has considerable potential, but there is of yet no such machine available.

2. Balogh, Richard L., and Purdum, Don L.. Computer Assisted Instruction, Feasibility Study, NASA CR-917, January 1968.

   This is the result of a 15 month study by Philco-Ford on the feasibility of a large scale CAI effort for the Manned Spacecraft Center in Houston, Texas.

3. Engel, Gerald L. Computer Assisted Instruction. A selected bibliography and KWIC Indes, U. S. Naval Weapons Laboratory, Technical Memorandum No. K-3/67, January 1967.

   This is a fairly complete listing of the pertinent references on CAI.

4. Jacobs, James N., Yeager, Helen, and Iford, E. Jean T. "An Evaluation of Programed Instruction for the Teaching of Facts and Concepts," The Journal of Programed Instruction. Vol. III, No. 4, 1966. pp. 29-38.

   The noteworthy results of this evaluation are that;

   a. Conventional method classes had the least information decay.

   b. Programed instruction for long usage results in student boredom. Topics of relatively short duration hold the most promise.

5. Strum, R. D., and Ward, J. R. "Some Comments on Computer-Assisted Education in Engineering Education," IEEE Transactions on Education. Vol. E-10, No. 1, March 1967. pp. 1-4.

   This article is a result of a study of CAI at the U.S. Naval Postgraduate School. The result of this study is that authors have abandoned the CAI approach and are now using programmed instruction in book format.

6. IBM 1401/1440/1460-1026 and 1440-1448. Operating Systems Computer Assisted Instruction. Author and Proctor Manual, May 1965. File No. GENL-48, form C24-3384-1.

7. IBM 1401, 1440, or 1460. Operating System, Computer Assisted Instruction. March 1965. File No. 1401/1440-48, form C24-3253-1.

8. Introduction to the Coursewriter Language, IBM, January 1967.

Bibliography

Bitzer, D. L., and Braunfeld, P. G. "Computer Teaching Machine Project: Plato on Illiac," Computers and Automation, Feb. 1962.

Bitzer, D. L., and Braunfeld, P. G. Problems and Prospects of Teaching with a Computer, Automated Instructional Techniques. Bedford, Mass.: The Mitre Corporation, 1962.

Bitzer, D. L, Braunfeld, P. G., and Lichtenberger, W. W. Plato: II A Multiple-Student, Computer Controlled, Automatic Teaching Device, Coulson, J. (Ed.) Programed Learning and Computer-Based Instruction, New York, Wiley and Son, 1962.

Bushnell, D. D. Computer-Based Teaching Machines, J. of Educ. Res., 1962, 55, 236-245.

Bushnell, D. D. The Computer as an Instructional Tool. A summary, Systems Development Corporation, SP-1554, February 1964.

Bushnell, D. D. and Cogswell, J. F. A Computer-Based Laboratory for Automation in School Systems, Santa Monica, Calif., Systems Development Corporation, 1961.

Chapman, R. L., and Carpenter, J. T. Computer Techniques in Instruction, Coulson, J. (Ed.) Programed Learning and Computer-Based Instruction, New York, Wiley and Son, 1962.

Computer-Aided Teaching, Technology Review, February 1964.

Coulson, J. E. Computers in Programed Instruction and Educational Data Processing, Automated Instructional Techniques, Bedford, Mass.; The Mitre Corporation, 1962.

Coulson, J. E. A Computer-Based Laboratory for Research and Development in Education, Coulson, J. (Ed.) Programed Learning and Computer-Based Instruction, New York, Wiley and Son, 1962.

Coulson, J. E. Paper presented at the ONR-SDC conference on Digital Computer Teaching Machines, Oct. 1961.

Demonstration Manual for Computer Assisted Instruction Executive Presentations - IBM Watson Research Center Report.

Dunwell, S. W. Using a Computer to Teach Programming, IBM Watson Research Center, RC-854, December 1962.

Four Case Studies of Programed Instruction, The Fund for the Advancement of Education, June 1964.

Grubb, Ralph E., and Selfridge, Lenore D. The Computer Tutoring of Statistics: A Preliminary Report, IBM Watson Research Center Report RC 724.

Hjelm, H. F., Beaton, A. E., Flanagan, J. C., and Cooley, W. W. A Symposium: The Computer and Educational Research, <u>Har. Educ. Rev.</u>, 31, 3, 1961.

Hoernes, G. E., and Heilweil, M. F. "Introduction to Boolean Algebra and Logic Design," A Program for Self-Instruction, McGraw-Hill Book Co., 1964.

IBM 1401 or 1440 Operating System Computer Assisted Instruction - IBM Systems Reference Library Form C24-3253-0.

Licklider, J. C. R. Paper presented at the ONR-SDC conference on Digital Computer Teaching Machines, Oct. 1961.

Licklider, J. C. R. Preliminary Experiments in Computer-Aided Teaching, Coulson, J. (Ed.), <u>Programed Learning and Computer-Based Instruction</u>.

Lumsdaine, A. A. Teaching Machines and Auto-Instruction Programs, Educational Leadership, February 1961, 271-277, 314.

Lumsdaine, A. A. Criteria for Evaluating Self-Instructional Programs, School Life, November-December, 1962, Vol. 45, No. 2, 15-17, 39.

Lumsdaine, A. A. Some Theoretical and Practical Problems in Programed Instruction, Coulson, J. (Ed.) <u>Programed Learning and Computer-Based Instruction</u>, New York, Wiley and Son, 1962.

Mager, R. F. On the Sequencing of Instructional Content, Psychological Reports, 1961, 9, 405-413.

Mager, R. F. and Clark, C. Explorations in Student-Controlled Instruction, Psychological Reports, 1963, 13, 71-76.

Maher, A. Computer-Based Instruction (CBI): Introduction to the IBM Research Project, IBM Watson Research Center, RC 1114, March 1964.

Markle, S. M. Teaching Machines Versus Programers, AV Communication Review, Vol. 10, No. 4, July-August 1962, 286-292.

Morrison, H. W. Computer Processing of Responses in Verbal Training, IBM Watson Research Center, RC-1133, February 1964.

Rath, G. J, Anderson, N. S., and Brainerd, R. C. The IBM Research Center Teaching Machine Project in Automatic Teaching (Ed. by Eugene Galanter) Wiley, 1959.

The Research in Programed Instruction, An Annotated Bibliography, Bulletin, 1964, No. 35, OE 3403, U.S. Dept. of Health, Education and Welfare, Office of Education.

Schurdak, J. J. An Approach to the Use of Computers in the Instructional Process, IBM Watson Research Center, RC-1237, June 1964.

Stolurow, L. M. and Davis, D. J. Teaching Machines and Computer-Based Systems, Training Research Laboratory Tech. Rep., Univ. of Illinois, No. 1, August 1963.

Swets, J. A., Feurzeig, W., Harris, J. R., and Marill, T. The Socrates System: A Computer System to Aid in Teaching Complex Concepts, AMRL Memorandum P-43, June 1963.

The Use of the COURSEWRITER Language - IBM Watson Research Center Report.

Uttal, W. R. My Teacher has Three Arms, IBM Watson Research Center, RC-788, September 1962.

Uttal, W. R. Paper presented at the ONR-SDC conference on Digital Computer Teaching Machines, Oct. 1961. (Also: IBM Research Report RC-532.)

Uttal, W. R., Charap, M., and Maher, A. The Computer Tutoring of Stenotypy: A preliminary Report, IBM Research Center Report, RC-663, April 11, 1962.

APPENDIX A: Student Comments

Computer Assisted Instruction

Since 1958 extensive research has been employed in the field of computer assisted instruction. The objective of this research has been to develop a method of presenting information to an individual in such a manner that he will be able to learn new material from a machine rather than another person.

The main driving forces behind this research are the realizations that teachers are becoming more and more in demand and classroom space is not keeping up with the expanding population. With the aid of computers, it should be possible, in theory, to present material to many students at the same time through typewriter terminals or display units and let each student learn at his own rate of speed.

The technique that has been developed so far is contained in the computer language of COURSEWRITER. This language is used by the instructor to present his course to the students, check their answers to questions, and control the flow of course presentation to the students.

The basic coursewriter language consists of twelve commands. These commands are divided into three categories and each category is assigned a priority in its control of the computer. The three categories are as follows:

1.	Primary op codes	rd	qu				
2.	Major op codes	ca	cb	wa	wb	nx	un
3.	Minor op codes	ty	ad	br	fn		

The primary op codes are used to present text and ask questions. They always require a response from the student, either an answer or a signal that he has completed reading the material presented to him and is ready for the computer to proceed. The major op codes are used to control the checking for correct and incorrect as well as uncrecognizable responses from the student. The third group of minor op codes control the action to be taken by the computer in the event that a correct, incorrect, or unrecognizable response has been detected. With the use of these op codes the instructor is able to program his course material in a manner that can be presented in a logical fashion to the student.

Programming in coursewriter is similar to programming in assembly language, except that the operand field is the actual text to be printed or checked. As an example, here is a program that demonstrates the use of the coursewriter language.

```
start     qu Read Page 35 of your book, and when you are
             finished type in -     ready     -.

          ca ready

          ty Let's see what you have learned!

          un If you are ready, type in -    ready

ques      qu 2 * 4 = ?

          ca 8

          cb eight

          ty Good, 2 * 4 = 8

          wa 6

          ty No, * does not mean 'add'

          un No, * means to multiply

          un No, re-read page 35 and try again.
          un
          br example
```

As you can see from the previous example many different teaching techniques can be programed into CAI. The first instruction caused a reference to a text book to be typed to the student. By having him enter the word, ready; the student is given practice in communicating with the computer. Also a discipline is imposed on the student in that the program will not proceed further until the student has followed instructions and entered the correct code word as requested.

After a question is asked, correct answers as well as anticipated wrong answers can be checked for and appropriate messages typed back to the student. If the student's response does not match a right or wrong answer the "un" command will type the first hint to the student and then wait for another response. (See the last page of this report for detailed explanation of each command.)

If the student can not match the correct answer after the last hint has been given the program will then branch to a "subroutine" and display an example that may be beneficial to the student. When the student signals that he has read the example, the same question can be asked again and the same hints will be presented as before if he still can not determine the correct answer. It should be noted at this time that this is a very simple example, and that through the use of counters and sense switches more elaborate teaching programs can be written. For example, a counter and a conditional branch would allow a different example to be presented to the student until finally he could be given the correct answer or told to seek aid from his instructor.

One of the most powerful commands available to the instructor is the fn command. The command is used to call in special function subroutines from the system library to process the student's answer or to supply additional data for the program. One use for the fn command is to edit out shift characters, commas, periods, and extra blank spaces from the student's response so that his answer will compare equal to the stored correct answer. Another use is to supply random numbers to the program so that different numbers will be presented to the student each time the same problem is used.

The basic problem with CAI is the tremendous responsibility it places on the teacher to write a good course. This is difficult and requires a great deal of time and careful planning. If the instructor does not give good hints or does not present the material clearly the student may never be able to understand it at all. For example:

Good	Bad
qu Write $\frac{1}{2}$ as a decimal fraction	qu $\frac{1}{2}$ =
ca .5	ca .5
ty Yes, $\frac{1}{2}$ = .5	un No
un Divide 2 into 1.	un No, try again.
un 2 divided into 1 = .X? What should X be?	un Come on now!
un The answer is .5, enter it.	un You can divide better than I can.

The experience that I have had with CAI programming has led me to believe that this use of a computer has its greatest potential as a teaching aid rather than as a replacement for the teacher. Sitting for hours in front of a terminal can become very boring, and it isn't long before you start talking to yourself. Along with the solitary confinement you are stuck with the inability to ask questions at the time when you are most receptive to the answers. Also you are unable to get hints or helpful comments until you have already done something wrong. In some cases you may accidently guess the correct answer and the program will proceed, leaving the student behind wondering how he should have really analyzed the problem.

As a teaching aid, however, where the student could come to see extra examples and work additional problems, the CAI program could be a great help. This would allow the teacher to program several examples in detail and let each student study them at his own rate. This would give the teacher more free time, incourage a student to seek additional help because he would not feel the embarrassment of demonstrating to the teacher his lack of understanding, and could serve as a reinforcement of the student's knowledge since he could work the example problem one step ahead of the teacher and verify his own work, getting help where he needed it.

rd Computer types text and waits for student to signal completion (send an EOB).

qu Computer types text and waits for student to respond.

ty Computer types text and continues without waiting for any response from student.

ca Correct answer to be stored in memory for comparison with student's answer.

cb Similar to ca -- used to identify one of a set of correct answers when the subsequent action is the same regardless of which answer in the set is given by the student.

wa Wrong answer to be compared with student's answer.

wb Similar to wa -- used to identify one of a set of wrong answers when subsequent action is the same for all answers in the set.

un Text to be typed if student's response is not one of the specified correct or wrong answers.

br Branch -- alters the sequence of execution.

nx Signals the computer to execute all minor op codes following it.

ad Used to add or subtract constants or counter values from a counter.

fn Used to call in a function to supply data to the program or to process the student's answer.