DOCUMENT RESUME

ED 089 794 IR 000 519

AUTHOR	Countermine, Terry A.
TITLE	The Development and Evaluation of a Teaching and Coursewriting Computer Language (TACL).
INSTITUTION	Pennsylvania State Univ., University Park. Computer-Assisted Instruction Lab.
REPORT NO	PSU-CAI-57
PUB DATE	Jun 73
NOTE	125p.; Ed.D. Dissertation, Pennsylvania State University
EDRS PRICE	MF-$0.75 HC-$5.40 PLUS POSTAGE
DESCRIPTORS	*Computer Assisted Instruction; Computer Programs; *Computer Science; *Curriculum Design; *Curriculum Development; Doctoral Theses; Program Descriptions; Program Evaluation; Programing; Programing Languages
IDENTIFIERS	TACL; *Teaching and Coursewriting Computer Language

ABSTRACT

A description is provided of the design and development of an author language for computer-assisted instruction (CAI). This Teaching and Coursewriting Computer Language (TACL) is described as being easy to learn for newcomers to computers and as providing efficiency and time savings in course development without sacrificing power or flexibility. Individual chapters of the report discuss: 1) general aspects of CAI; 2) programing languages for CAI course design; 3) the computer science aspects of CAI author languages; and 4) the implications of TACL. (Author/PB)

TACL: A Teaching and Coursewriting Language

by

Terry A. Countermine

An Abstract of a Thesis

in

Computer Science

Submitted in Partial Fulfillment

of the Requirements

for the Degree of

Doctor of Education

August 1973

The Pennsylvania State University

The Graduate School

ABSTRACT

Computer-assisted instruction is one of the new, exciting and dynamic branches of educational technology. In the best case, CAI combines the advantages and sophistication of computer technology with the latest theories and knowledge of human learning to provide a stimulating and effective instructional program for individual learners. Well developed CAI courses take advantage of the power and flexibility of the computer to produce dynamic student-computer interactions.

The design of such CAI courses, however, is a time consuming process that involves a great deal of computer programming and testing. To a great extent, the development of CAI has been hindered by the absence of a programming language suitable for educators and authors of CAI courses. The need for such a language is directly attributable to the high costs of developing a non-trivial CAI course.

This document describes the design and development of an author language that is easy to learn by persons naive to computers, is efficient and time saving for course development and does not sacrifice the power or flexibility of existing CAI languages.

ACKNOWLEDGEMENTS

I would like to thank and acknowledge the contributions of Mr. Terry Bahn without whom this project could not have been completed. His insight and expertise in the field of computer-assisted instruction was invaluable. Also thanks to Dr. Karl Borman, Mr. James Watts and Mrs. Bonnie Shea for their suggestions, cooperation, help and encouragement. A special note of thanks to Dr. G. Phillip Cartwright for helping to initiate this project and to Dr. Neil Jones for teaching me the computer techniques that were needed to write the software.

TABLE OF CONTENTS

Chapter Page

APPENDICES

LIST OF TABLES

LIST OF FIGURES

CHAPTER I

STATEMENT OF THE PROBLEM

Computer-assisted instruction is one of the new exciting and dynamic branches of educational technology. In the best case, CAI combines the advantages and sophistication of computer-technology with the latest theories and knowledge of human learning to provide a stimulating and effective instructional program for individual learners. As will be shown in the later sections of this dissertation, CAI has been shown to be capable of producing superior learning in shorter time periods than conventional instruction. Well developed CAI courses take advantage of the power and flexibility of the computer to produce dynamic student-computer interactions. However, the adaptability of computer-assisted instruction to individual students needs is not easy to achieve. Course preparation for sophisticated CAI is a time consuming process that involves a great deal of computer programming and testing. To a great extent, the development of CAI has been hindered by the absence of a programming language suitable for educators and authors of CAI courses. The need for such a language is directly attributable to the high costs of developing a non-trivial CAI course. Using currently available languages and techniques, the ratio of preparation time to online student time for tutorial CAI instruction is in excess of 100 to one. (28)

This document describes the design and development of an author language that is easy to learn by persons naive to computers, is efficient and time saving for CAI course development and does not sacrifice the power or flexibility of existing CAI languages.

CHAPTER II

COMPUTER ASSISTED INSTRUCTION

What is CAI?

Computer-assisted instruction (CAI) is often confused with academic programs teaching courses in computer science. There is a distinct difference between instruction about computers and instruction by computers. As an example, a university might offer a curriculum in computer science which enables a student to learn programming concepts, systems design, information retrieval, and other computer related topics. Such courses might very well be taught in a traditional classroom mode of instruction. At the same time, that same university might have the facilities to use computer-assisted instruction as a method of teaching any topic for which a CAI course was available. Courses in human development, mathematics, Fortran programming, and many other fields might be taught through the use of computer-assisted instruction.

Another source of confusion is the liberal use of the term computer-assisted instruction (CAI) when computer-managed instruction (CMI) is meant, and vice versa. CAI refers to a mode of instruction in which the student interacts with the computer and receives instruction directly from the computer program. Because of its extraordinary memory and logic, the computer program can store a student's past responses and use such information to individualize instruction for that student. CMI differs in that it is the instructor that interacts with the computer. He uses the computer mainly as a management tool for record keeping and information retrieval.

The computer software makes statistical information available which the teacher can use to individualize instruction. With CAI the individualization takes place automatically. CMI, however, requires the teacher to intervene between the computer and the student and to determine the instructional sequence.

A significant part of any computer-assisted instruction application is the design and development of the course material which is presented to the student through a computer terminal. Depending on the objectives of the instruction and the student's background and level of achievement in a given area of study, certain modes of instruction would be more effective than others.

The most common mode is that of problem-solving. Students must first learn a programming language in order to write programs related to the course work which they are taking. In this mode the computer is being used as a problem-solving and exploratory tool.

Drill and practice assumes that students need a great deal of practice in order to master certain basic knowledge, procedures, vocabulary, nomenclature or mathematical skills. Drills to provide this practice can be presented by the computer in a fairly standardized fashion. The patterns for student-computer interaction are generally limited to simple correction and retrial. Utilizing the extensive memory, the endless patience and the ability of the computer to adapt to student performance, this mode of CAI has been very effective. The level of difficulty and rate of presentation can be modified to meet the needs of each student. This potential to individualize instruction is a very strong argument for developing the use of CAI. (10)

A third form of CAI can be defined as simulation, with the computer responding adaptively to learner input. An artificial but realistic environment is established which enables the student, through interaction and feedback, to investigate the simulated configurations. To implement this mode of CAI the teacher(s) must be able to define the model sufficiently to permit it to be programmed. For example, at Bolt, Beranek and Newman, Inc. a computer has been programmed to simulate the conditions of a patient brought into a hospital emergency room. A physician in training sits down at a teletype terminal and, by requesting information, tests and symptoms from the computer regarding the "patient," is able to provide a diagnosis of the specific injuries that the "patient" has received. (14)

Gaming simulation is different in that the student plays problem-oriented games instead of investigating real-life situations. Various games have been designed to develop certain thought processes which are useful in other fields of study.

Three economics games have been developed: the Sumerian game, the Sierra Leone Development Project game, and the Free Enterprise game. These games simulate current economic and business situations in an attempt to teach the students the thought processes necessary in making related decisions. The Sierra Leone Development Project game simulates the economic problems of a newly formed African nation. Situations are taken from actual problems that Sierra Leone has faced.

The student assumes the role of Second Assistant Affairs Officer at the United States Embassy in Freetown. He proceeds from problem to problem and, if successful, is promoted to Assistant Affairs Officer, and finally to Chief Affairs Officer. Each position brings up problems of a broader scope. (37)

The inquiry mode of CAI is used in situations where files and search algorithms have been established in the computer enabling students to ask questions about various topics. In this mode the system responds to the student inquiry with answers which have been stored by the authors. That is, the course authors must anticipate the questions which will be asked so that the answer may be stored in a file accessible by the computer. Many library management systems use this type of CAI.

A final definition of CAI involves the computer in the role of a tutor. This mode tends to simulate the natural dialog between a teacher and a student. Instructional sequences that use remedial and skip-ahead pathways selected on the basis of previous student responses are incorporated extensively by computer programs to move the student toward the attainment of a set of specifically defined behavioral objectives. Such programs are complicated and difficult to write, but when done correctly this mode of CAI is very effective.

Effectiveness of CAI

Two forms of evaluation--formative and summative--are common in the CAI literature.

Formative evaluation is evaluation at the intermediate developmental stages of a program. The results of formative evaluation are intended to serve as the basis for altering the nature of the program in its formative stages. Formative evaluation and the resultant curriculum revision improve the probability that future students who use the program will achieve mastery of the material.

Summative evaluation is terminal evaluation concerned with the comparative worth or effectiveness of a CAI program and alternative instructional procedures. The results of summative evaluation are not intended to serve directly in the revision, improvement or formation of a program; rather they are gathered for use in making decisions about support or adoption. (6) Summative evaluation of computer-assisted instruction has been increasing each year as the field of CAI matures. (10) Most of these studies have indicated that CAI can be a viable instructional technique. It has potential for becoming a substantial instructional innovation.

Cartwright and Mitzel (7) described the summative evaluation of a three-credit course, "Early Identification of Handicapped Children," designed for regular classroom teachers primarily in rural areas. On-campus students who registered for "Introduction to the Education of Exceptional Children" were randomly assigned to conventional instruction (CI) and to CAI. Objectives for both courses were the same; in fact, the teacher of the CI class had been one of the authors for the CAI course. Using the time to complete the course and the score on the 75 item final exam as variables, the authors reported that the CAI students (N=27) scored significantly (23%) higher than CI students (N=87) on the final exam and completed the course in twelve hours (33%) less time than the CI students. (7)

A group under Donald Bitzer of the University of Illinois has done several studies comparing CAI to conventional instruction. Using the PLATO system and various subject matter (computer programming, clinical nursing, foreign language; mathematics) the resu ts indicated that the CAI students did as well as and in many cases better than those taught through CI. The results also showed that the desired criterion levels were achieved in less time by the CAI groups. (2)

A very detailed study conducted at the Florida State University by Hansen, Dick, and Lippert compared CAI to CI in the teaching of college level physics. (15) During an eleven week term, 69 students scheduled to take Physics 107 were randomly divided into three groups; those taught by CAI, those taught by CI and those taught by a combination of both. The CAI students completed the lessons in 17% less instructional time. Since there was a fixed total time for all students, the extra time saved by the CAI students was used mainly for repetition of material which the students felt was difficult. Table 1 shows the grade distribution for the three groups.

Table 1

Final Grade Distribution in Three Instructional Conditions

	Frequencies of Final Grades					
Conditions	A	B	C	D	Mean Grade	Total Students
Total CAI	11	6	6	0	3.22	23
Partial CAI	6	7	10	0	2.83	23
CI	4	5	13	1	2.52	23

Personal interviews conducted after the term revealed that the CAI participants felt that they had a greater concept mastery in comparison with their peers. For example, the CAI students claimed to be better explainers of homework problems than their dorm-mates who attended the conventional course.

Using the Stanford Achievement Test (SAT) to measure achievement in mathematics in the California schools during the 1967-68 school year, Suppes and Morningstar found significant differences between CAI (in the drill and practice mode) and conventional instruction (CI) favoring CAI in the second, third, and fifth grades. No significant differences were found in the first, fourth or sixth grades. In a concurrent study in McComb, Mississippi, significant differences were found at all grade levels. (35) Suppes and Morningstar attribute the overall superiority of the experimental program in Mississippi more to a lesser increase in performance level for the CI groups in Mississippi than to a greater change in performance level for the Mississippi CAI groups relative to the California CAI groups.

There are many more studies available reporting summative evaluations of computer-assisted instruction. Much of the literature cited has been summarized by F. M. Dwyer (10) as follows:

> 1. CAI appears to be a viable instructional technique having its capabilities thoroughly grounded in current learning theory. It has the potential for becoming a very substantial instructional innovation; however, it must be emphasized that CAI is still in its experimental (infancy) stage and a long way from actualizing its inherent capabilities.

2. The available evidence indicates that CAI can teach as well as live teachers or other media, that students can learn in less time, and that students respond favorably to CAI.

3. The empirical research reported so far concerning the instructional effectiveness of CAI (in terms of experimental design, number of students participating and duration of the instructional treatments) appears to be less than desirable. It may be that since CAI systems are often being developed and perfected at the same time that research is being conducted, adequate time and money may not be available for implementing well-designed experimental evaluation.

Costs of CAI

By far the biggest criticism against CAI is the cost involved. Studies, however, are beginning to show that when done correctly, the cost of CAI can be brought within acceptable limits. (29) Probably the most careful cost analysis as applied to possible CAI systems was made by Kopstein and Seidel who concluded that using specified but reasonable assumptions, the cost per student hour of CAI in higher education can be about $2.60 per hour, which compares favorably with conventional university level instruction calculated to average about $2.75 per hour. (22) Conventional instruction at the primary grade level costs about 30¢ per student hour, so CAI may not be economically favorable for that market. However, D. Bitzer, at the University of Illinois' PLATO project, is working toward of goal of 30¢ per student hour and hopes to achieve it by 1976. (2)

One of the main costs is initial program development. Two things must be considered in this respect. First, as more is learned about the teaching-learning process with respect to effectiveness (and even efficiency) the initial stages of course development will be shortened.

A second consideration in judging over-all cost must be that as more effective courses are becoming available they may be shared by others who have only the operational costs, i.e., a course developed in Pennsylvania may be used anywhere in the world where the required computer system exists.

Finally, it is difficult to "cost out" CAI. Expenses must be amortized across other uses. Some of the economic problems associated with the use of computers in education can be solved through more effective use of time-sharing and satellite computers, ranging from increased off-line applications to scheduling pupils via computer into homogeneous or alternate, logistical groupings. Utilization of the computer system at a level near its full capacity and capability is necessary. Finding this level is a goal of CAI advocates.

There are several things that can be done to aid in lowering the costs of CAI. As has been mentioned, research and development costs are high. Effective methods to share CAI courses among various CAI centers is needed. This would cut down the duplication of effort, amortize the developmental costs over more students, and increase the total availability of courses throughout the country.

Improved authoring languages and input procedures would also help to lower costs. Such authoring languages and input procedures would result in a reduction in the ratio of author time to student time. The less time it takes to produce a usable CAI course (author time) the lower the cost of that course.

The ideal situation would be to develop a program that would convert a CAI course written in a language for one CAI system into an equivalent course in some other language for another CAI system (see Figure 1). Such a development would essentially result in CAI courses that were machine independent.

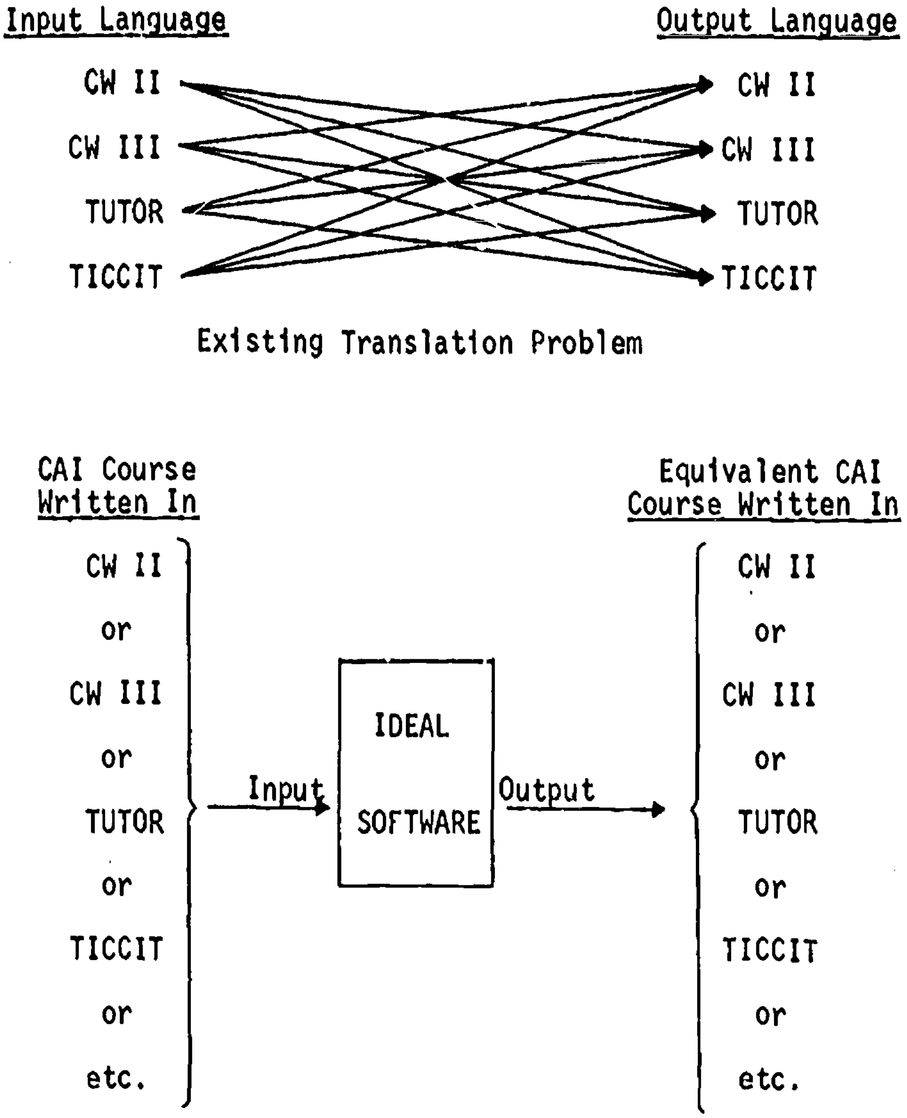
Hardware Configurations

General Purpose Systems. Computer-assisted instruction is sometimes judged on the basis of articles read or demonstrations seen several years ago. In many cases, general purpose systems were useo to attempt CAI. That is, business oriented machines were adapted slightly to enable cathode ray tubes and teletypes to be added to the configuration. Promoters of these systems could then advertise the available CAI possibilities in addition to other applications on the same system. Such "piggyback" systems are not generally adequate CAI applications.

Hewlett-Packard (HP), Philco-Ford (PF), and Digital Equipment Corporation (DEC) have invested time and money in such dual purpose systems. Although they offer CAI at a fairly attractive price, their over-all CAI capabilities are limited. Generally, those that use a cathode ray tube only have upper case characters and very limited or no graphics. DEC uses a teletype terminal which offers a hard-copy for the student but limits the teaching strategies that may be used.

A serious problem with piggyback systems is the limited number and quality of student stations that can be handled by a single system ranging from a low of five (DEC Edusystem 10) to not more than 16 (HP 2001A) with purchase costs ranging from $10,000 to $100,000. Student stations range from teletypes to limited cathode ray tubes. (26)

If the IBM system 360-370 and other similar machines (Burroughs B550, G.E. 635, SDS 940, PDP 10) are added to the list of general



Existing Translation Problem

Ideal Translation Solution

Fig. 1. CAI language translation; problem and ideal solution.

purpose systems sometimes used for CAI, the number of possible terminals is increased to a maximum of 200, but the costs may exceed $1,000,000.

Piggyback systems are important and necessary to the development of CAI, but their usefulness is limited to problem-solving, drill-and-practice, and simulation applications. These applications offer a genuine aid to the students without degrading the system as a batch processing unit.

Special Purpose Systems. The following systems have been designed especially for use in computer-assisted instruction. Each has some unique features which distinguish it from each of the others.

IBM 1500 Instructional System. The IBM 1500 Instructional Computer System was designed specifically for providing individualized instruction at each student station (maximum of 32). Each student station is equipped with a small cathode ray tube (CRT) on which is displayed alphameric information plus a wide variety of graphics including animated illustrations. Sufficient information to fill the 640 display positions of the CRT (16 horizontal rows and 40 vertical columns) is available in micro-seconds from a random access disk. Student response components of the CRT include a typewriter-like keyboard with upper and lower case characters plus a wide variety of special characters and a light-sensitive pen used by the learner in making responses to displayed material. In addition to the CRT, each student station has a rear-screen image projector on which are displayed color photographic images from a 1,000 frame 16mm film with each frame randomly accessible by the computer with a search rate of 40 frames per second. The third display component is an individual audio play/record device with randomly accessed, pre-recorded messages on standard 1/4 inch audio tape. A pictorial diagram of the 1500 system is presented in Figure 2.

At this stage of development, the IBM 1500 instructional system is very good for research and experimentation in CAI. The limitation of 32 terminals per system is a serious one if large scale CAI is to be attempted. More will be said about this system in Chapter 4 of this paper.

PLATO. The PLATO system, developed at the University of Illinois under Professor Donald Bitzer is one of the earliest CAI developmental systems. Originally the system operated with only facilities for a single student. Using the ILLIAC computer with high-speed memory of only 1,024 words, a two terminal operation developed (PLATO II).

In 1964, transition was made from PLATO II to the PLATO III system based on the CDC 1604 computer. PLATO III had a theoretic limit of 1,000 terminals, but only 20 were implemented. Using the PLATO III system, more than 70,000 student contact hours have been produced in electrical engineering, geometry, nursing concepts, library science, chemistry, algebra, computer programming, and foreign languages.

In a recent report, Bitzer projected that response times would not exceed a maximum of 1/10 of a second and projected a cost

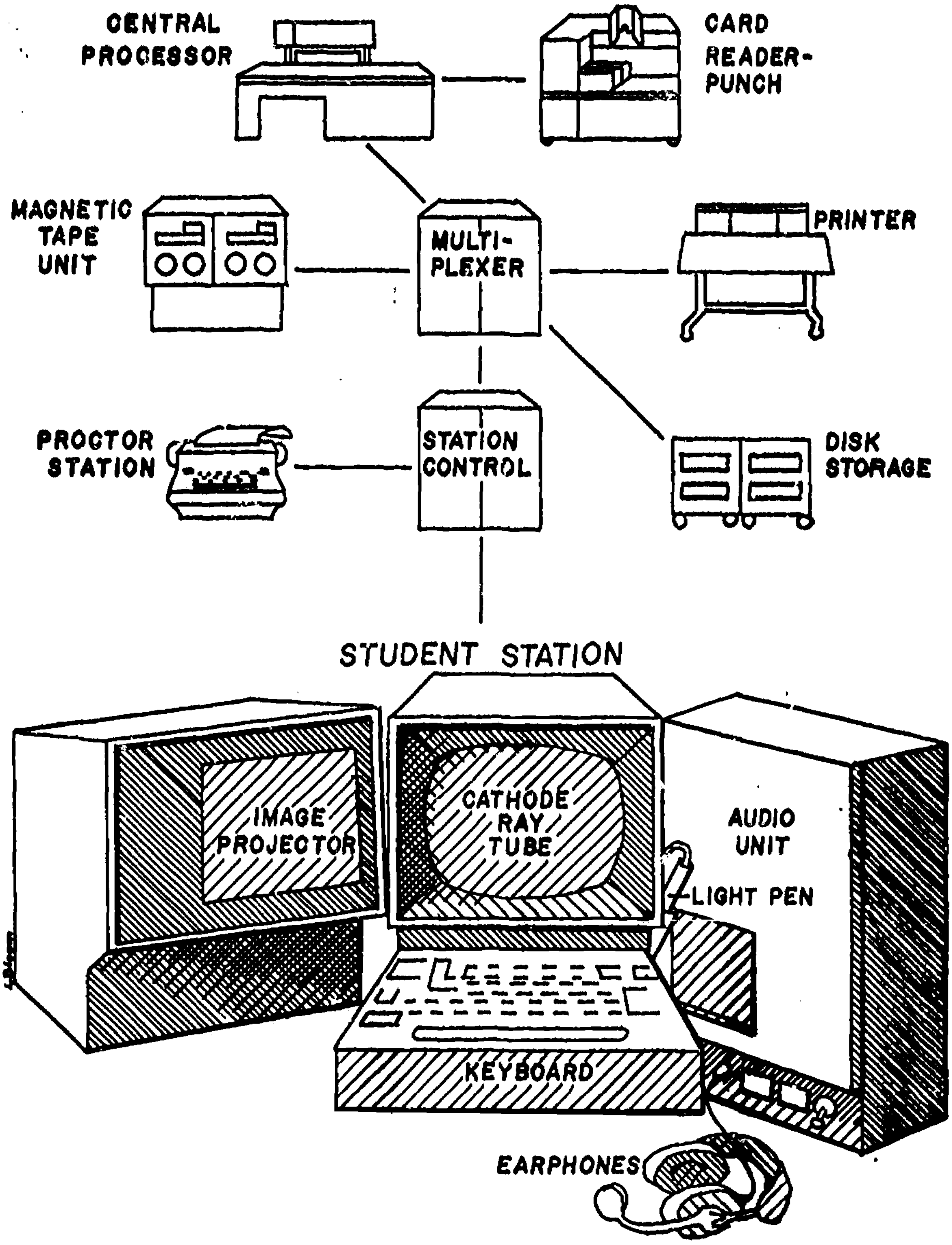


Fig. 2. IBM 1500 Instructional System with 1 of 32 Student Stations.

of 34¢ per student hour of use. This calculation is based on an eight hour student contact day and does not include the use of the CDC 1604 as a batch processor during other hours. (2)

The main hardware development associated with the PLATO system is the plasma tube. In fact, the economic feasibility of Bitzer's proposed teaching system is dependent upon the newly-invented plasma display panel now under development at the University of Illinois and other laboratories. This device combines the properties of memory, display and high brightness in a simple structure of potentially inexpensive fabrication. In contrast to the commonly-used cathode ray tube display, on which images must be continually regenerated, the plasma display retains its own images and responds directly to the digital signals from the computer. This feature will reduce considerably the cost of communication distribution lines. (2)

Results of PLATO studies dealing with the evaluation of instructional effectiveness parallel the studies cited earlier in this paper. Generally, they show that students do at least as well as, and in many cases better than similar students receiving traditional instruction. As stated earlier, the most significant difference is the time required to complete the instruction.

TICCIT. Time-share, Interactive, Computer-Controlled Information Television (TICCIT) is the most recent effort to make CAI a market success. It is different from most other systems in that it uses an off-the-shelf, commercially available, color television screen for its main information display. From the student's viewpoint, the terminal consists of a color TV, a headphone set, and a typewriter-like keyboard. Under computer control alpha numerics and line graphics in seven colors, as well as full color movies, can be displayed on color TV monitors. Up to 17 lines of 41 characters each may be displayed. The character set is completely programmable, with up to 512 distinct different characters available at any single time.

The main processor is a DATA General Nova 800 configured as a time-sharing minocomputer with 32,768 words of core storage, special hardware time-sharing protection features, and the usual host of standard peripherals, in addition to three large moving-head disk drives containing up to 50 million characters. Another disk memory is used for student records.

A second NOVA 800 is used as the student terminal processor. It services the TICCIT terminals by receiving and processing keyboard entries and by generating new displays to be sent to the terminal. The buffered computer-to-computer link uses both a fixed-head disk, accessible by both minicomputers, and a direct-memory-to-memory data transfer system to provide intercomputer queuing capability and fast data transfer. (34)

The TICCIT system is self-contained and supports a maximum of 128 terminals located up to 1500 feet from the computer. Video and audio information transmitted to the terminal and keyboard signals transmitted to the computer are frequency multiplexed on the same coaxial cable.

Another new aspect of the TICCIT system is its capability to use standard coaxial cable. Since the TICCIT terminal display is a television receiver and requires a signal similar to and compatible with that of normal television, a cable TV system can carry TICCIT signals. Several techniques to deliver CAI to the home via a cable television system have been developed and are being studied. (34) The projected commercial cost including hardware, equipment maintenance and CAI programs is less than $1.00 per student contact hour. This is more than the projected costs of the PLATO system but is certainly within acceptable limits if it can deliver a high quality of CAI.

CHAPTER III

PROGRAMMING LANGUAGES FOR CAI COURSE DESIGN

Overview

A review and comparison of the existing languages which are commonly used for CAI is presented in this chapter with the intent of identifying the desirable features of a CAI language which will lead to the development of an author language for enhancing the ease and flexibility of authoring CAI courses.

Interactive Computer Languages

APL. A Programming Language (APL) was designed by K. E. Iverson in 1962 and has since been further developed in collaboration with A. D. Falkoff and L. M. Breed. APL is a mathematical language dealing with transformations of abstract objects, such as numbers and symbols, whose practical significance, as is usual in mathematics, depends upon the interpretation placed upon them. Although APL is relatively easy for a computer scientist or mathematician to learn, it is definitely not oriented towards non-mathematical oriented CAI authors.

APL employs the use of primitive functions which are provided by the system, or defined functions, which the user provides by entering their definitions on the input terminal in addition to many library functions. Such concepts as scalor and vector constants, scalor and monadic and dyadic functions, local and global variables--just to mention a few--must be understood before using APL effectively.

Another source of confusion for the non-scientific author is the APL character set. Many of the symbols are mathematical in meaning and appearance. A few examples are <, ≤, ≥, >, α, ω, ρ, ~, ⌈, ⌊, ⊂, ⊃, ∩, ι, and ○. A clever user can form almost any function definition, not necessarily mathematical, that he desires, but such manipulation is far from trivial.

The Florida State University has implemented a CAI program using a PDI-8 computer and the APL language. To do this, however, they had a few APL programmers write many functions which could be used by the other authors simply by inserting the necessary parameters. These functions perform many non-numeric operations such as text processing and display. In effect, they have created a different language consisting of APL functions.

There is no doubt that APL is a powerful interactive computer language. It has the capability of doing almost any type of computing that one would like to do. It lacks, however, the ease of learning that is necessary for a CAI authoring language. A summary of APL features is given in Table 2.

Coursewriter II. The Coursewriter II (CW II) language was designed by IBM to enable a course author to communicate with his students through the use of the IBM 1500 Instructional System. It was intended that CW II would be an easy-to-learn language for any educator who had the desire to write a CAI course. Since the language is not oriented toward any special instructional methodology, facilities were to

Table 2

Comparison of the Features of APL, CW II, Dowsey Author Entry System, and VAULT

Language Features	APL	CW II	VAULT	DOWSEY
ease of learning	difficult	difficult	easy	easy
human vs. computer time	high human	high human	low human	low human
programmer time	high	high	moderate	low
key puncher time	moderate	high	high	low
graphic	no	yes	no	no
error occurrence for new programmer	high	high	moderate	moderate
co-ordination input	moderate	easy	difficult	easy
flexibility of screen display	adequate when function has been designed	very flexible	depends on the logic division used	very flexible
creating text	must use function definitions	tedious	punch on cards	punch on cards
branching capabilities	flexible	flexible	limited without card stuffing	flexible
use of implicit branching	difficult in answer processing	good	not used in answer processing	not used in answer processing
frame identification on CRT	none	not unless programmed	none	yes, automatic
diagnostics	adequate, but mathematically oriented	mostly in terms of parameters	poor, many errors hard to find	good
immediate execution possible	yes	yes	no	no

allow many different teaching strategies to be programmed into any given course. The result is a very powerful language, but one that is not easy to learn as had been intended.

The language is broken down into major and minor instructions. This classification is important in the use of some of the opcodes and can result in serious errors if not completely understood. For example, consider the following from the Coursewriter II manual describing how to control course flow by use of the answer set:

> "The analysis of responses by Coursewriter II begins with the first encountered member of the answer set and proceeds as follows:
>
> 1. If the anticipated and actual response do not match, all minor instructions up to the next member of the answer set are ignored. Comparison of the next member with the answer in the response buffer will then take place.
>
> 2. When a match of any type is found, all minor instructions are executed until the next defined correct answer (ca), wrong answer (wa), or additional answer (aa), is encountered. Or, in the case of the last group of a set under consideration, minors are executed until the next major instruction which is not in the group is encountered.
>
> 3. If no match occurs for the ca's or cb's, comparison will proceed with the aa's. All minors in between are ignored."

When a programmer has mastered the CW II language, such descriptions are meaningful. They are not, however, easy to grasp by beginning authors.

Another source of difficulty in CW II is the use of many parameters. To display a line of text on line 5 starting in column 10 the author might use:

```
dt   5,10///  This is a line of text.  e
```

Almost every instruction has a field where one or more parameters must be inserted. Even to write a very simple course segment, these parameters must be understood.

It should be emphasized that Coursewriter II is a very powerful interactive computer language, but it is not author-oriented. A summary of CW II features is given in Table 2 and a sample of a course listing written in CW II is presented in Appendix 3.

Author Oriented Languages

Several compilers, preprocessors, and authoring aids have been created specifically for the CAI author in response to the need for simple languages and procedures for the non computer-oriented authors. The following two, VAULT and Dowsey Author Entry System, were designed mainly for use with the IBM 1500 Instructional System.

VAULT. A Versatile Authoring Language for Teachers (VAULT) was designed by E. W. Romaniuk, R. R. Jordan, and W. Birtch of the University of Alberta. It runs on an IBM System/360 Model 67 computer and produced Coursewriter II source code as output. The source deck (cards) must then be assembled on the 1500 system before use.

The main aspect of VAULT is its division into two separate and distinct parts, the LOGIC division and the DATA division. The LOGIC division specifies the type of presentation and logical strategies to be used in the program. The DATA division consists of the actual course content which is to be presented in a manner defined by the

LOGIC division. The course was divided this way to decrease author time and improve the quality of the resulting courses. (21)

Both the LOGIC and DATA divisions may be subdivided into three types of units, BLOCKS, LESSONS, and PROBLEMS. The PROBLEM is the smallest unit of division of VAULT. Within the PROBLEMS are located specific instructions and/or specifications that define the course that the student will receive. PROBLEMS in LOGIC contain VERBS (or action instructions) while PROBLEMS in DATA contain KEYWORDS and the associated course material. One or more PROBLEMS make up a LESSON, and one or more LESSONS make up a BLOCK.

A disadvantage of VAULT is that the resultant courses are very repetitive since much of the same LOGIC is used over and over again. A given LOGIC division will produce essentially the exact same sequence of code regardless of the content of the DATA division. A summary of VAULT features is presented in Table 2.

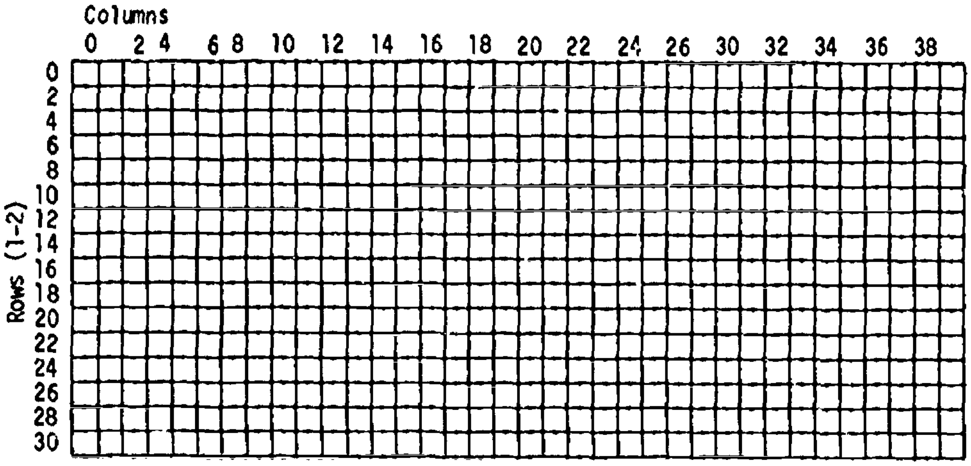
Dowsey Author Entry System (DAES). This system was developed by M. W. Dowsey to work in conjunction with the coding form developed by Peter Dean of the IBM Corporation. It will run on either the IBM 1130 or 360 computer. The form is divided into four sections: identification, presentation, decision, and response analysis (see Figure 3). The author fills out these forms specifying such things as rows to be erased, CRT image, possible responses, what to do for various responses, and other course specifications. Once the author has designed the course using these forms, a corresponding deck of cards is produced which can be run on the Dowsey pre-processor. The format of these cards is very rigid (fixed) depending on the way the given forms

PAGE LABEL ____________
(1-12)

From row [ ][ ] to row [ ][ ] erased. [ ] Restart point? Check if required.
(16-17) (19-20) (22)

TEXT
(6-71)
(72 is continuation)

Pause Time
in seconds
(75-80)



After this frame, the student should go to:

□ The Return Point

□ The Next Logical Frame

□ The Last Question

(1)

□ A Frame Named ...

(1)

Enter E,K,N, P or U.

(You may enter a 2-character response identifier)

__________ if his □ response was ______________________ □ □
__________ if his □ response was ______________________ □ □
__________ if his □ response was ______________________ □ □
__________ if his □ response was ______________________ □ □
__________ if his □ response was ______________________ □ □
__________ if his □ response was ______________________ □ □
(1-12) (14) (16-71) (75) (76)

Fig. 3. DAES Coding Form

were filled out. See Figure 4 for a sample listing of the card input to the preprocessor. Once the pre-processor deck has been produced it is submitted as data to the pre-processor and a CW II deck is generated. This CW II deck is then assembled on the 1130 system in order to produce an executable CAI course.

The most frequent cause of errors in the use of this system is in the preparation of the input deck to the pre-processor. A look at Figure 4 indicates the rigidity of the card format. A mispunched character or a punch in the wrong card column would result in an error. A summary of DAES features is given in Table 2.

An Ideal Authoring Language. Each of the previously mentioned languages had some shortcomings when analyzed from the non computer-oriented author's viewpoint. An ideal author oriented language would include at least the following features:

1. Easy to learn
2. Easy to use
3. Versatile enough to allow many instructional strategies
4. Clear communications between author and programmer
5. Operational

Table 3 compares the desirable features of an author language for non computer-oriented authors and compares those features with APL, CW II, VAULT, and Dowsey accordingly. The last column gives the rating for an ideal authoring language.

TACL. Teaching and Coursewriting Language (TACL) was designed for use on the IBM 1500 Instructional System. The main design goal was

```
---------------------------------------------------------------
1                 0 31 R
 4    <W>HEN WAS THE <B>ATTLE OF <H>ASTINGS?
---------------------------------------------------------------
1
1A           N 1066                                          RI
1B           N 1000-1099                                     NR
1C           U
---------------------------------------------------------------
1A                0  0
 8       <J>OLLY GOOD  <S>POT ON
---------------------------------------------------------------
N
1B                0  0
 8       <Y>ES,RIGHT CENTURY.
      <B>UT WHAT ABOUT THE EXACT DATE?
---------------------------------------------------------------
Q
1C                   0
---------------------------------------------------------------
4                                      1066                   5
8        <N>OT EVEN THE RIGHT CENTURY
10    <W>ELL,IT WAS 1066.
2                 8 11 R
---------------------------------------------------------------
6     <N>AME THE KING WHO WAS KILLED.
1
---------------------------------------------------------------
2A           E <H>AROLD                                      EX
2B           K ARLD                                          SP
2C           U
10           U
---------------------------------------------------------------
2A                0,0
 8       <E>XACTLY RIGHT                                      2
---------------------------------------------------------------
10    <N>OW FOR A MORE DIFFICULT QUESTION.
N
2B               0   0
 8    <Y>ES,<I> BELIEVE YOU'VE GOT IT BUT FOR
---------------------------------------------------------------
      THE SPELLING     <T>HE CORRECT VERSION
      IS <HAROLD>.
---------------------------------------------------------------
N
2C                0  0
 8     D OES THIS QUOTATION HELP YOU RE-ANSWER?               2
      ' A ND ------ GOT SHOT IN THE EYE'.
---------------------------------------------------------------
Q
2D                0  0
---------------------------------------------------------------
8     <W>ELL,IT WAS <HAROLD> WHO GOT SHOT IN
1/    THE EYE.
N
3                 0  0 R
```

Fig. 4. Listing of Card Input to DAES Pre-processor.

Table 3

Language Features and Ratings of
APL, CW II, VAULT, Dowsey, and
An Ideal Authoring Language

Scale: 1 - Undesirable; 2 - Acceptable; 3 - Very Desirable
(Low) (Moderate) (High)

Language Features	APL	CW II	VAULT	Dowsey	An Ideal Authoring Language
ease of learning	1	1	3	3	3
human vs. computer time	1	1	3	3	3
programmer time	1	1	2	3	3
key puncher time	1	1	1	1	3
graphic	1	2	1	1	3
error occurrence for new programmer	1	1	2	1	3
co-ordination of input	2	3	1	3	3
flexibility of screen display	2	3	2	3	3
creating text	2	1	2	2	3
branching capabilities	3	3	2	2	3
use of implicit branching	1	2	1	1	3
frame identification on CRT	1	2	1	3	3
diagnostics	2	2	1	2	3
immediate execution possible	3	3	1	1	3
Total Rating	22	26	23	29	42

to create a language with characteristics and features of the "Ideal Authoring Language" presented in Table 3. In the case where one person writes the course (non computer-oriented author) and someone else does the actual programming (computer-oriented programmer) it was intended that the communication between the two would be improved by providing a common language that both could understand. Other objectives were to reduce the amount of time needed to construct a workable CAI course and, therefore, to increase the quantity and quality of course material generated. Essentially, the design goals were those presented in Table 3 and identified as an "ideal authoring language."

One of the characteristics of TACL that is an improvement over Dowsey and VAULT is that the input is done on-line through the cathode ray tube. The author or programmer sees on the screen the way a line of text will actually look to the student. This cuts down the number of subsequent revisions.

There is no division of TACL into different parts. Certainly, the author may write a course composed of several chapters or segments, but there is no distinction between the course content and how it is presented. That is, the author writes a single program in which he presents the course in whatever manner he desires.

Another characteristic of TACL which increases usability is that it may be run completely on the 1500 instructional system. There is no need for another computer or any other special equipment. The elimination of punched cards also seems to increase efficiency since there are no lost decks or misplaced cards. The source input is written on magnetic tape and may be listed, processed or updated at any time. Table 4 gives a comparison of TACL and an ideal authoring language. TACL will be explained in detail in Chapter IV.

Table 4

A Comparison of TACL and An Ideal Authoring Language

Language Features	TACL	An Ideal Authoring Language
ease of learning	3	3
human vs. computer time	3	3
programmer time	3	3
key puncher time	3	3
graphic	2	3
error occurrence for new programmer	3	3
co-ordination of input	3	3
flexibility of screen display	3	3
creating text	3	3
branching capabilities	3	3
use of implicit branching	3	3
frame identification on CRT	3	3
diagnostics	3	3
immediate execution possible	1	3
Total Rating	39	42

CHAPTER IV

COMPUTER SCIENCE ASPECTS

Introduction

TACL was designed as a step towards the ideal CAI language. To understand the overall TACL system it is necessary to first understand the structure of the IBM 1500 Instructional System, course design using CW II, and the relationship of each of these to TACL. This chapter provides the necessary background to clarify the logic behind the TACL software and TACL opcodes.

The 1500 instructional system uses the IBM 1131 central processor. This central processing unit (CPU) is interfaced with other hardware which enables the use of Cathode Ray Tubes (CRTs), audio units, and image projectors. The 1131 central processor can also be used independently of the 1500 system.

The IBM 1130 system is embedded within the 1500 system. Thus, batch processing jobs can be processed which are written in FORTRAN or 1130 assembler language. Since the 1500 system uses the 1131 central processor, both systems cannot be "up" simultaneously.

IBM 1500 Instructional System

From the student's viewpoint, the IBM 1500 instructional system appears as in Figure 5. He need not be aware that there is a computer involved at all. To use the system, he sits down at the student station and signs on to the course he is taking. Each student who is taking a

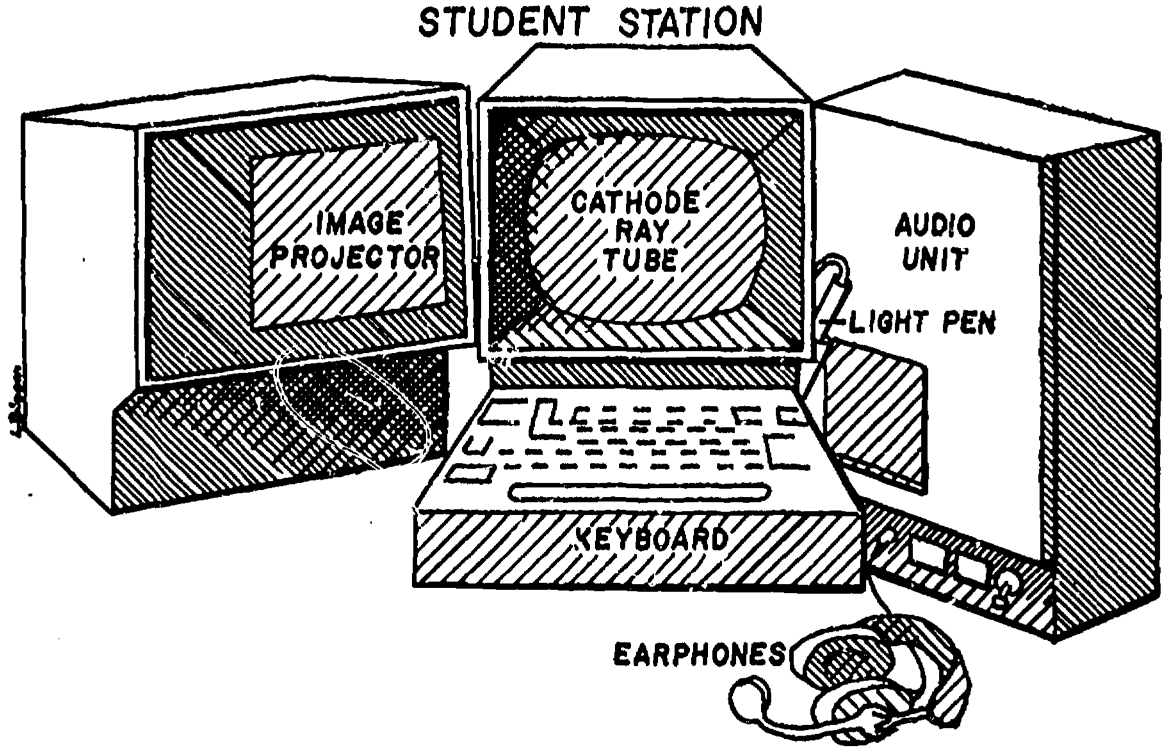


Fig. 5. 1500 System from the Student Viewpoint.

course is assigned a student number. This number, along with the course name, are the only codes needed to sign on. Once signed on, the student is given instruction using one of the many available modes discussed earlier.

In every course there are certain places designated as restart points. Each time the student signs on he begins instruction at the last restart point which he passed through during his previous instruction. Thus, the flow of the course from one time to the next can be controlled by the course author.

The system is said to be operation in the student mode when a student is signed on to a course and is receiving computer-assisted instruction. During this mode, each student is continually interacting with the computer; when a question is presented to him by the system, he makes a response and waits for the computer to react. The delay in time between a student response and the next visible computer action is referred to as the latency time. The shorter the latency time, the better the system is operating.

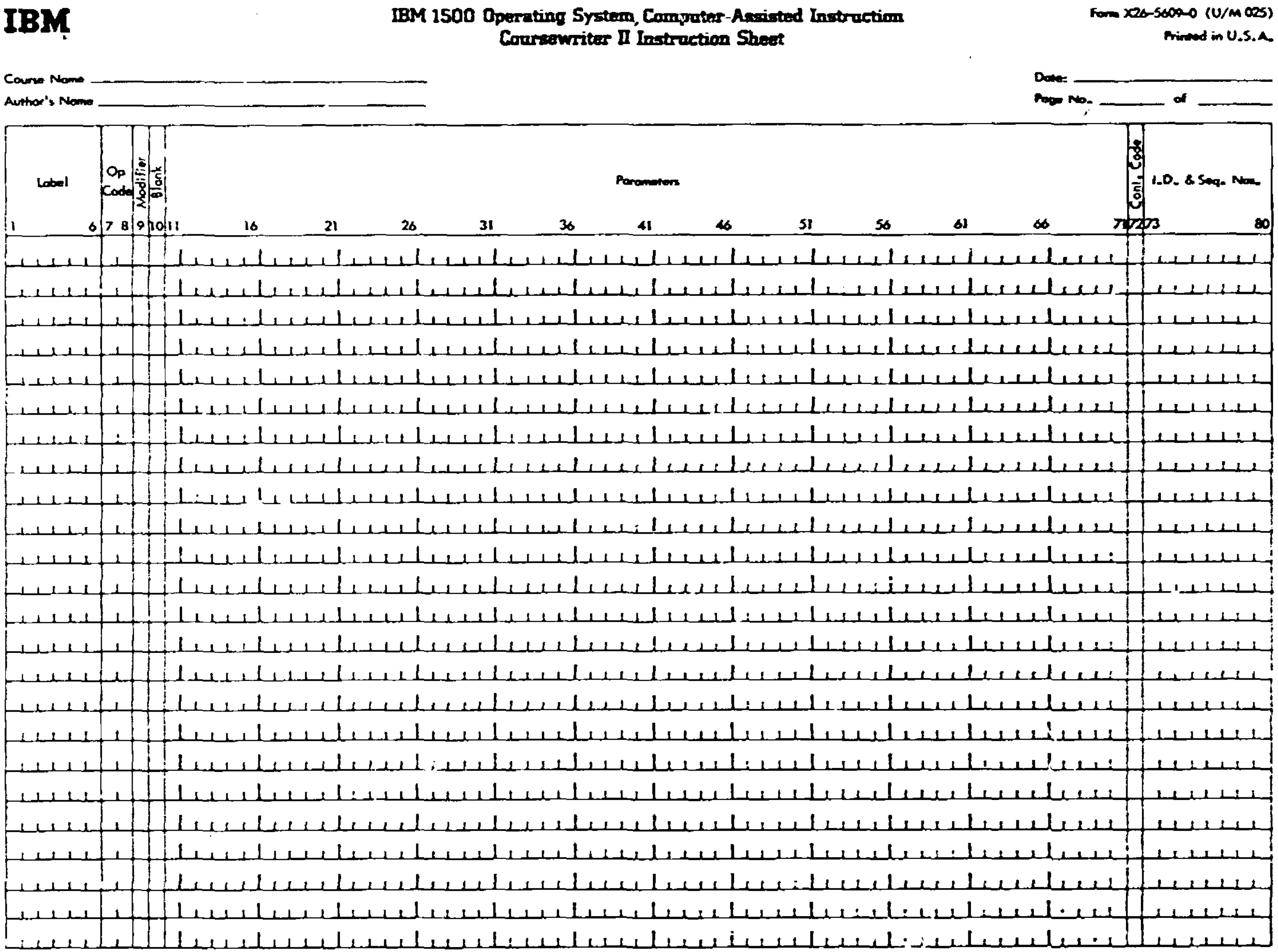
The author may or may not want to have every student response recorded. While designing the course he decides which responses are most important and through his program causes these to be written onto tape when any student takes that section of the course. Since many students may be on line at the same time, responses on this performance tape are intermingled. When the teacher who is running a course wants to look at the performances of the students, he has a service program run which extracts this information from the tape. Thus, trouble spots may be found in the course such as a poorly worded question or a badly phrased point of information. These may then be changed by modifying the program which comprises the course.

Course Design by Coursewriter II

When an author decides to write a course for use on the IBM 1500 instructional system, he would probably design a small segment of instruction, using CW II coding forms (Figure 6). At this point he has two choices: (1) punch the code using a specific format onto cards; or (2) enter the code, again using a set format, through the CRT using the 1500 system. If he chooses (2) the system is said to be operating in CW II author mode. This differs from student mode (from the system viewpoint) in that the on-line CW II assembler must be used. The principle problem with the author mode is that the latency time increases due to the use of the assembler.

If (1) is chosen, the forms are usually given to a keypunch operator who punches the code on cards. Since many of the characters available on the CRT are not available on the IBM 029 keypunch, many single characters must be punched as two separate characters on the input card. These special combinations are interpreted by the software and are reconverted internally to a single character. The problems with this approach are apparent.

Once the source deck has been punched it is submitted to the system operator who has it assembled by the off line CW II assembler on the 1130 system. This approach is cheaper and does not increase the latency time. However, it can only be done when the 1500 system is not operating.



IBM

IBM 1500 Operating System, Computer-Assisted Instruction
Coursewriter II Instruction Sheet

Form X26-5609-0 (U/M 025)
Printed in U.S.A.

Course Name ______________________

Author's Name ______________________

Date: ______________

Page No. ______ of ______

Label	Op Code	Modifier	Blank	Parameters	Cont. Code	I.D. & Seq. Nos.
1 6	7 8	9	10	11 16 21 26 31 36 41 46 51 56 61 66 71	72	73 80

Fig. 6. Coursewriter II Instruction Sheet.

At this point in the course design procedure, regardless of whether (1) or (2) was used, there is an executable form of the course on one of the available disks. The author may then sign on to his own course and evaluate it. If it is not satisfactory, he would change the existing code and/or add new code using the procedure just described. See Figure 7 for a flowchart of the course design procedure using CW II.

Course Design Using TACL

Designing a course using TACL is in some ways similar to the preceding CW II explanation. There are several different time modes in the TACL procedure that should be understood. The off-line author mode is the time when the author is designing the content and presentation of the course. When this is finished the author would fill out TACL coding forms (Figure 8) explicitly defining a particular segment (chapter) of the course. (See Appendix 1)

At this point in time the author would enter the on-line author mode. That is, he would sign on to the 1500 system to a "course" called AUTHOR. It is the purpose of this program to accept TACL commands and write them onto tape so that they may be processed later. Thus, TACL commands and course content are entered through the CRT with the aid of TACL coding forms. The tape that is produced will be referred to as the raw TACL tape.

INIT/EDIT

The raw TACL tape is next processed by either the INIT or EDIT software. If it were the very first time that this course segment were being processed, INIT would be used. INIT produces as an output the first version of the master TACL tape. If the raw TACL tape contains

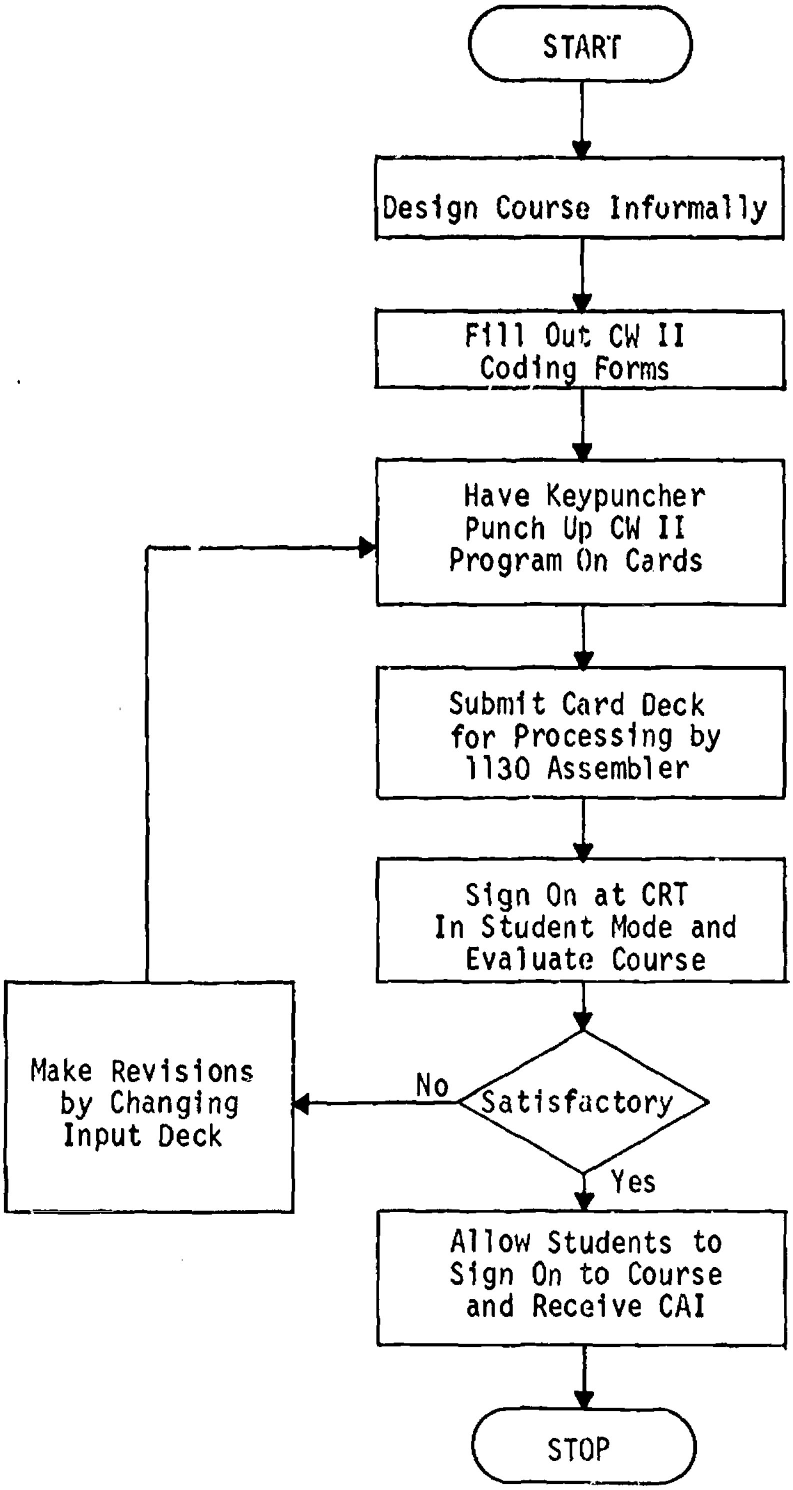
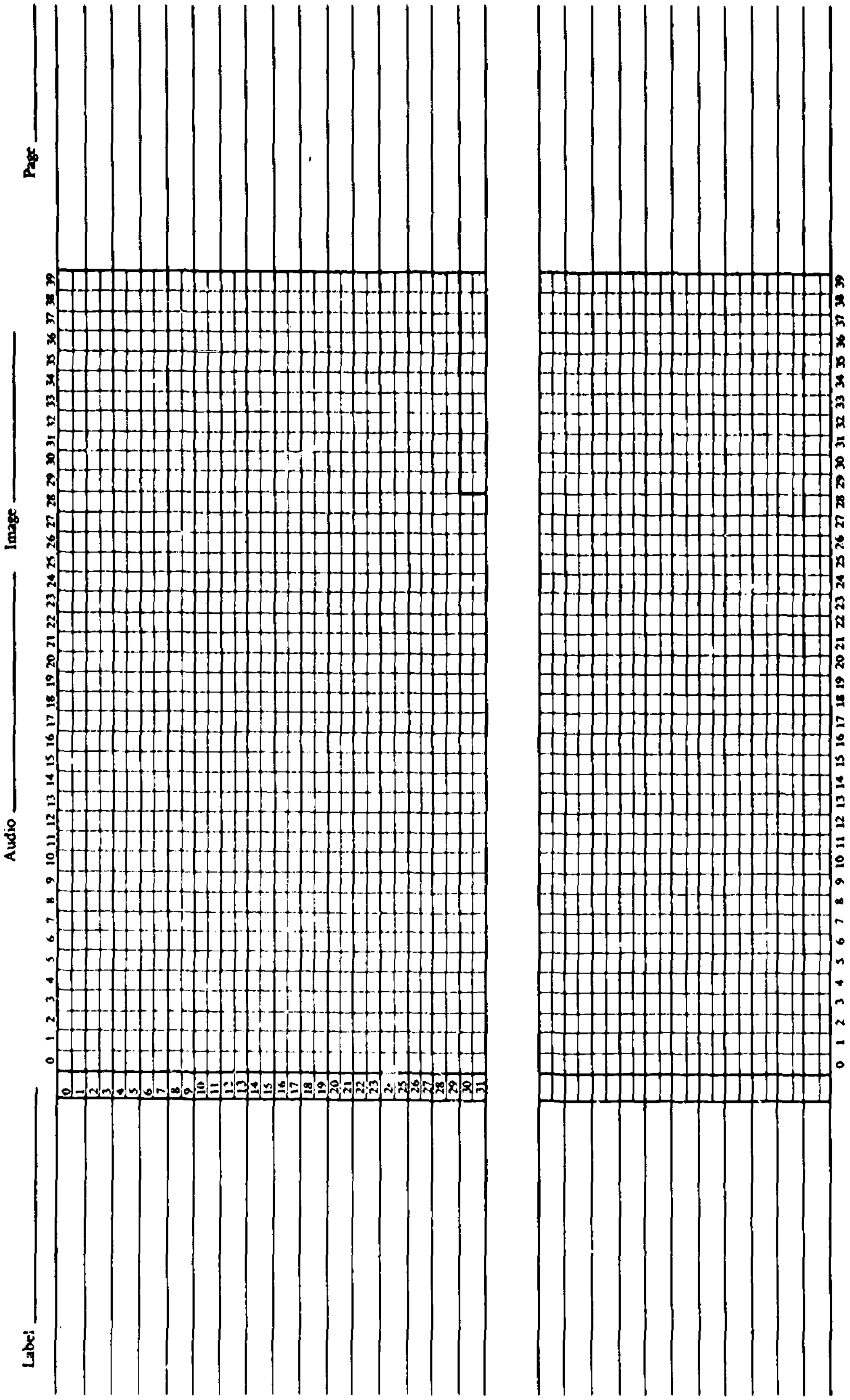

START
Design Course Informally
Fill Out CW II Coding Forms
Have Keypuncher Punch Up CW II Program On Cards
Submit Card Deck for Processing by 1130 Assembler
Sign On at CRT In Student Mode and Evaluate Course
Satisfactory
No
Yes
Make Revisions by Changing Input Deck
Allow Students to Sign On to Course and Receive CAI
STOP

Fig. 7. CW II Authoring Procedure Using Card Input.

The Pennsylvania State University
The Computer Assisted Instruction Laboratory



Label ____________ Audio ____________ Image ____________ Page ____________

Fig. 8. TACL Coding Form

changes and/or additions to a previously processed course segment, the EDIT software would be used. This software uses the raw TACL tape in conjunction with a previously created master TACL tape and performs the desired editing, creating a new master TACL tape at the same time.

Whether INIT or EDIT is used, the result is the same. That is, two important output forms are produced: the master TACL tape and a CW II program that is equivalent to the TACL program. This CW II program is in compressed form and is stored on disk. The master TACL tape is in lined form and contains a compressed version of the actual TACL program. The structure of this master TACL code is described later in this chapter.

The output that the author sees is a TACL source listing showing a picture of the CRT for each frame and the TACL commands which correspond to the given text. (See Appendix 2). This listing is evaluated by the author who decides to make more revisions or to assemble the CW II course. If revisions must be made, the author once again signs on to AUTHOR and, using TACL commands, makes the desired corrections using the EDIT software and the master TACL tape. If the listing is judged satisfactory, the author will request that the CW II program that was produced be assembled. This assembly (done on the 1130 system) will produce an executable course segment on disk.

Next, the author would sign on to his course. This is really the critical part of evaluating his work. That is, he is receiving the computer-assisted instruction which he wrote in TACL and which his students will receive. If he is satisfied, the course is ready for students to take. If not, he will sign off his own course, sign on to AUTHOR once again, and perform the necessary editing.

Figures 9 and 10 explain the TACL procedure and illustrate the various modes.

Software Divisions

TACL is broken down into several different modules.

AUTHOR. The software that allows a course author to enter a course in the TACL language is called AUTHOR. This module is used each time an author wants to create additional code or modify existing code. He signs on to the IBM 1500 system as a student taking a course called AUTHOR.

As the TACL commands are entered through the CRT using AUTHOR, they are written on a magnetic tape previously referred to as the raw TACL tape. The 1500 system has two magnetic tape drives. One tape is used to keep track of student performance. The other tape (alternate to the performance tape) is used by AUTHOR to save the TACL commands.

AUTHOR is written in the CW II language and is essentially another course. When the 1500 system is "up" students may sign on to any course which is on one of the available disks. Thus, a TACL author signs on to AUTHOR and proceeds to enter his TACL program. As explained earlier, since the coursewriter on-line assembler is not needed in this mode, system response time is better and over-all system performance is improved. AUTHOR itself is a short course (program). The coding is straight-forward and very efficient from a systems point of view.

Several authors may be signed on to AUTHOR at the same time. Each record that is written on the raw TACL tape is identified by a user identification code. Also, when the author initially signs on, he

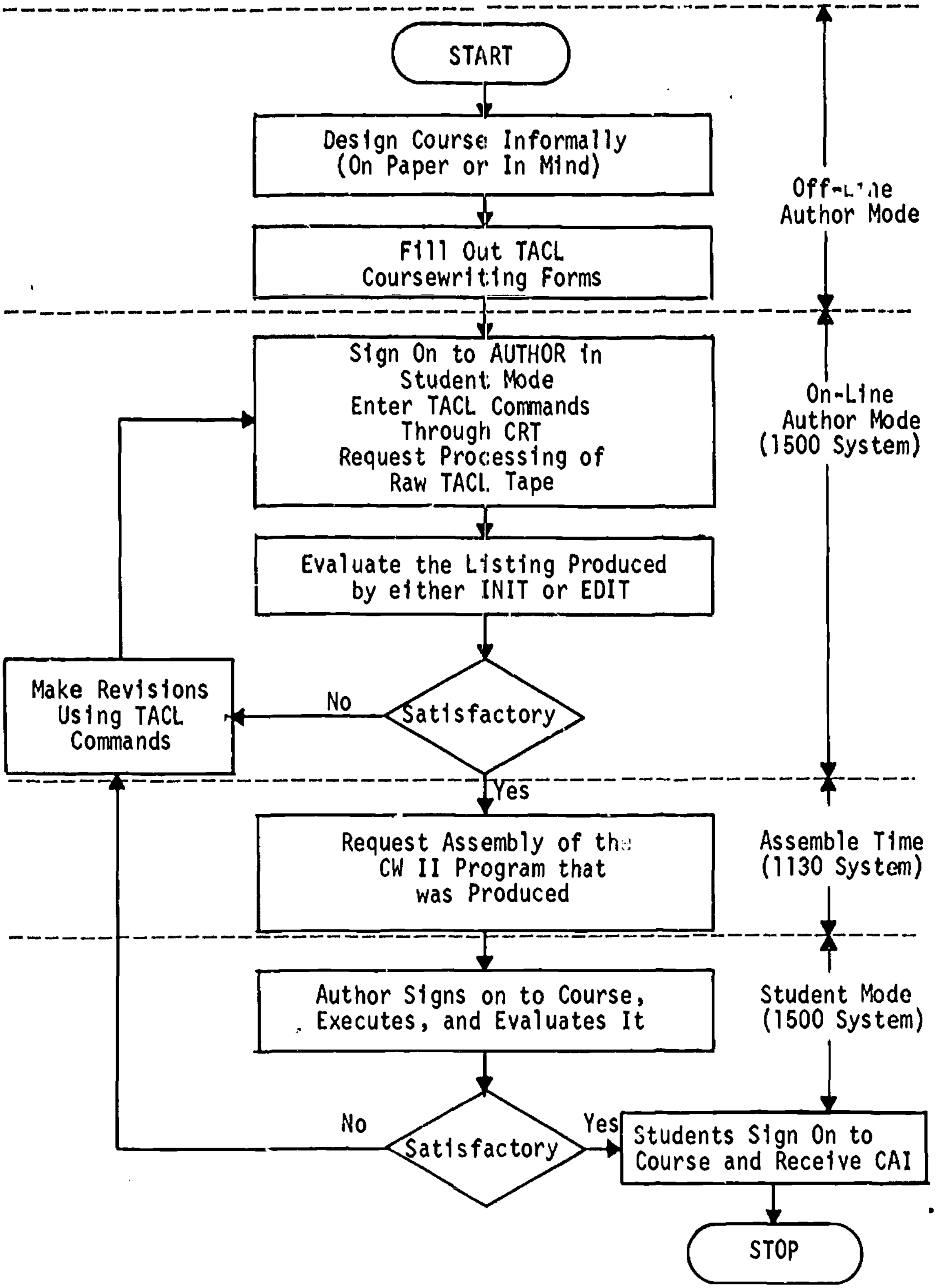

START
Design Course Informally (On Paper or In Mind)
Fill Out TACL Coursewriting Forms
Off-Line Author Mode
Sign On to AUTHOR in Student Mode
Enter TACL Commands Through CRT
Request Processing of Raw TACL Tape
On-Line Author Mode (1500 System)
Evaluate the Listing Produced by either INIT or EDIT
Satisfactory
No
Yes
Make Revisions Using TACL Commands
Request Assembly of the CW II Program that was Produced
Assemble Time (1130 System)
Author Signs on to Course, Executes, and Evaluates It
Student Mode (1500 System)
Satisfactory
No
Yes
Students Sign On to Course and Receive CAI
STOP

Fig. 9. TACL Authoring Procedure.

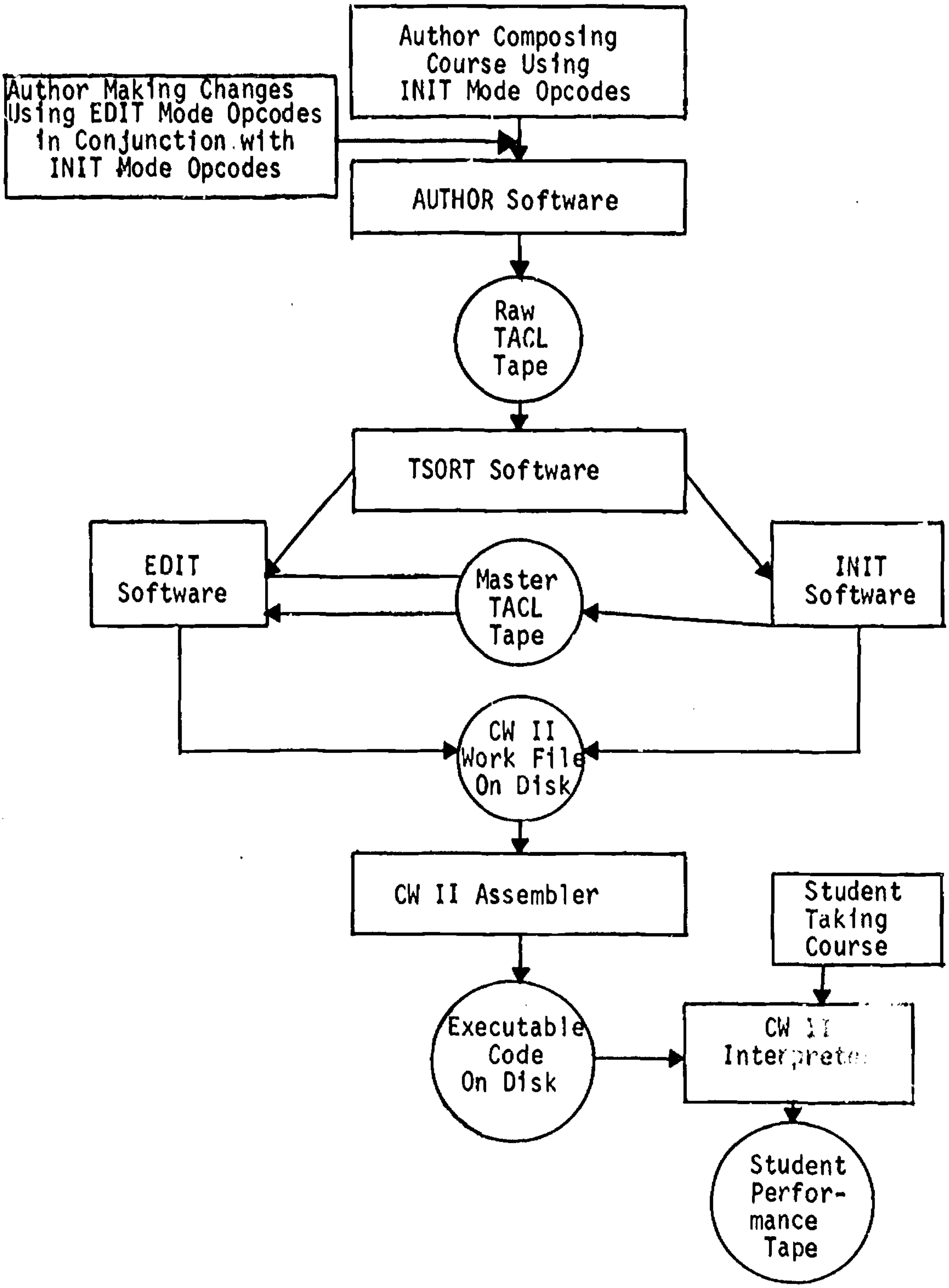


Fig. 10. Input/Output Diagram.

is asked to type the course name and segment code that he will be authoring and, to indicate whether he will be designing new material or editing a previously written course. Since both the INIT and EDIT modules process the same raw TACL tape, this information is necessary to make each record identifiable.

One of the strongest points for the authoring system being designed to use the CRT for entering commands is that the author can construct his text on a given line very much as it will appear when the course is executed. The relationship between two lines is sometimes distorted due to the presence of TACL opcodes. A feature is included which enables the author to double or triple space between lines without using the SKIP command; pushing the index key from one to four times creates spacing between lines. For the majority of cases double or triple spacing will be sufficient. This indexing feature cuts down the row counting necessary by reducing the use of the SKIP opcode.

Another function of AUTHOR deals with the FRAME opcode. Since coursewriting is frame (page) oriented, this command is used frequently. When AUTHOR senses this command at the beginning of a line of opcodes, it will ask the author if this specific location in the course is to be a restart point. After the author answers that question (yes or no), the entire CRT will be erased and the author will continue writing his course at row 0 of the new frame.

Raw Tape Records. As mentioned earlier, there may be several authors creating course material for the TACL system at the same time. When this occurs the raw tape records for a given course segment may be scattered throughout the tape. For example, a person may do a little authoring in the morning, sign off, and then return to enter some more code later in the day. In the meantime other authors may have been signed on. Another possibility is that the same author may enter course material for more than one segment on the same tape. These raw tape records are simply a copy of the TACL program as it was input through the CRT. Each record is marked with a user identification code, the segment name, and the segment number so that it may be identified by the TSORT software when processing of the raw TACL tape begins.

TSORT. Once an author has entered a sufficient amount of TACL course material he will want it to be translated into an executable course segment. The first phase in this step is to select the correct tape records from the raw TACL tape, and store them on disk for processing by INIT or EDIT. Tape records are selected from the raw tape according to the user identifier, course name and segment identifier. EDIT records are distinguished from initial records by a single word at the beginning of each record.

The author never explicitly requests TSORT. It is automatically run before either INIT or EDIT. TSORT, like all other software except AUTHOR, is written in 1130 assembler language. The CANCEL command results in the deletion of the TACL command immediately preceding CANCEL. This command is processed by TSORT; all other commands are processed later by INIT or EDIT. The resultant disk file may then be processed immediately or at a later time, thus allowing for time budgeting when necessary. TSORT, upon request, will give a rough listing (unformatted) that may be of help in finding obvious errors before the TACL code on the disk is processed. Thus, an author could do some editing before the original material is processed.

INIT - New Course Mode. This is the module that processes the initial TACL code for any given course segment. It will be executed only once for each segment. The input to INIT comes from the disk file that was created by TSORT. This disk file can contain over 1,400 sectors of TACL code with between 300 and 320 words per sector. Thus, a size limitation for TACL should not be a problem.

The INIT program itself consists primarily of a driver routine with calls to the various subroutines to generate the appropriate Coursewriter II code for a given TACL instruction. There are two main types of TACL instructions that must be processed by INIT. First, there is the text that will be read by the student. This is typed on the CRT as the author wants it to appear. Secondly, in order to specify the flow of this text, the author uses TACL opcodes. The opcodes determine, for example, how long to pause between the presentation of one paragraph and another, or to skip eight rows between two lines of text. A line of opcodes is distinguished from a line of text by preceding it with a cursor ([]).

Thus, the driver is continually checking for a cursor. If one is present, the TACL code is analyzed to determine what opcode is being used and the appropriate routine is called. In some cases an opcode may be determined from the first letter. At most, a scan of three is sufficient to isolate the opcode or to determine that the TACL code is in error. No cursor implies a line of text. If none is present the text routine then generates a Coursewriter II "dt" statement for the given text. This will cause the line to be displayed during student mode.

The author has the option to request that the TACL software skip the CW II code generation part of each TACL statement, only scanning for syntax errors. This will speed up execution.

If the CW II generation option is chosen, the coursewriter code that is generated is in workfile form. That is, it is linked together and compressed as it would have been if the input was Coursewriter II statements coming from punched cards. If the input were in punched card form, the CW II software would perform a pre-assembly which deletes trailing blanks and does some initial opcode scanning to catch misspelling and certain parameter errors. Thus, with TACL there is a saving of one complete pass by the IBM Coursewriter II assembler when the code is being assembled into executable 1500 system code. To speed up the assembly even more, the CW II code produced by INIT is in many cases given specifically. That is, instead of letting default conditions hold, certain parameters are filled in exactly as they should be. This increases the speed of the IBM CW II assembler, because of the way it was designed. Another point that increases efficiency in the CW II assembler is that CW II code produced by INIT is, with few exceptions, error free. Thus, few error traps are encountered in the CW II assembler.

As the CW II code is generated, a new master TACL tape for this course is also created. Each tape record is the same logical size as a disk sector. This tape contains a copy of the given TACL program in a shorthand form. Instead of saving the code in word form, a numeric code is assigned to each different opcode. Parameters, when present, are also abbreviated. As this master TACL tape is being created, it is linked together using a forward linked list structure. Processing thi$ master TACL code is much faster than processing raw TACL since much less analysis is required, the code is free of syntax errors, and each record is in compressed form.

INIT also saves a table that is associated with the linked master TACL tape. The table contains the statement number of the first TACL instruction on each tape record. That is, the fifth entry in the table would be the statement number of the first TACL instruction in the fifth record of the master tape. These tape records are 320 words long so that they correspond one-to-one with disk sectors. Ultimately, the processing is done from disk. This table is used by the EDIT software to quickly locate a TACL statement that is to be deleted or after which more TACL code is to be inserted. The input/output diagram in Figure 10 explains the relationship between the tapes and disks.

One of the most important parts of any system is the output format. The printer output from a TACL program (source listing) is designed to look like a CRT image (Appendix 2). The text is printed out corresponding to the row number in which it will appear during student mode. The opcodes are printed to the left of the screen image. Statement numbers are given for use in editing. Each opcode and line of text is assigned a different statement number.

In order to preserve the similarity of the listing to the CRT image, error messages are printed out at the end of each frame. For example, if there were any errors in frame 26 of a given course segment, they would be printed out below the picture of the CRT on the output for that frame. The error messages give appropriate statement numbers and description of the errors.

Since the printer is slow, the option of not obtaining a source listing is available. Error messages are still printed, but references would have to be made using the most recent listing.

Master TACL. The master tape that is created by both INIT and EDIT is a simplified form of the original TACL program and is called master TACL. Each instruction is linked to the next. The structure of some of the opcodes is changed, but the required information is saved. Let us look at an example.

0106	086D	000C	001A
word 1	word 2	word 3	word 4

In word 1 the first hexadecimal digit (left-most) is used in the editing phase. A zero implies that this is old master code to be used as is. The 106 in the right-hand 3 nibbles means that the next instruction is on disk sector 106. Word 2 contains the length of the present instruction in bits 0-6 and the displacement to the next instruction in bits 7-15. Thus, the length is 4_{16} and the displacement is $6D_{16}$. Word 3 contains the numeric opcode. In our example the $00C_{16}$ stands for the SKIP instruction. The $001A_{16}$ is the modifier of the SKIP. That is, SKIP to row $1A_{16}$ (26_{10}).

When EDIT is executed it will use this master tape to create a revised course and tape. The edit instructions appear on the master tape along with the master TACL code. Referring to the above example, let us assume that the SKIP 26 was statement number 56 in the course segment. To remove this instruction from the course a DELETE 56 would be used. This would change the left-most hex digit in word 1 from a 0 to an 8, making word 1 negative. Subsequently during processing, any master code for an instruction that has a word 1 with a negative value would be ignored.

To insert code into the segment in place of the SKIP 26, you would use an INSERT 56 instruction followed by the TACL code to be inserted. This would cause word 1 to be changed to the sector number where the inserted material is. The displacement field of word 2 would be changed to point to the beginning to the new inserted material in the new given sector. The length field would remain unchanged. Two new links must be created in front of the inserted code to point back to the next executable master instruction.

When the entire link building process is completed, the instructions are scanned from link to link leaving out deleted code and processing both master TACL and the new inserted material.

Each TACL statement, whether an opcode or a line of text, is given a statement number according to its sequential location within the TACL program. These statement numbers are not on the master tape. Consequently, one important rule in the editing phase of a course is to use the most recent source listing for referencing statement numbers. All editing instructions must be in relation to the statement numbers as they exist on the master tape (and, therefore, the latest listing). For example, if the author did a DELETE 16-20 and then an INSERT 20 followed by regular TACL code, the new code would be inserted after the twentieth instruction counting the deleted ones. This saves the author from having to keep track of where and how much is to be inserted and eliminates one of the problems of Coursewriter II that sometimes makes editing quite frustrating.

Since the master tape is not in CW II and the author is signed on to a CAI course, online, there is no way to immediately see the editing results (since they are done by the 1130 system). Immediate execution of TACL code would be a very desirable feature but would necessitate the writing of a completely new 1500 system background application.

EDIT. If a master TACL tape is available for a given course segment, editing may be performed on that segment. Existing material may be modified and/or additional code inserted. This is done by signing on to AUTHOR in the edit mode and using TACL edit commands to delete unwanted statements, to insert new statements, or to change TACL code. A raw TACL tape is created and TSORT is executed exactly as in the INIT mode. At this point, however, the EDIT software is used in order to update the old master TACL tape and create a new master TACL tape. EDIT transfers the master TACL program from tape to the same disk that the EDIT instructions are on.

First, the EDIT commands only are processed creating a linked TACL program consisting of both master TACL and raw TACL. Thus, we now have a program that has many instructions in the master TACL tape form (numeric opcodes, etc.) with new TACL code linked in various places throughout. Some of the master code may be marked for deletion during phase two of EDIT. Phase two is logically very similar to INIT with the difference being that two types of code must be handled. The new inserted TACL code is handled exactly as it was in INIT with many of the same routines being called. The old code, in most cases, takes much less analysis. Thus, there are a few separate routines used by EDIT that are not needed in INIT.

Generally, EDIT will execute faster than INIT in the actual processing of the entire TACL program largely because of the ease of processing the master code and the absence of syntax errors.

The source listing is primarily the same as in INIT. To distinguish between inserted opcodes and opcodes from the old master TACL tape, however, the statement numbers of the inserted TACL code are preceded by an '*'. This enables the author to quickly spot the code that he added.

EDIT produces a workfile Coursewriter II program (as does INIT) and a new master TACL tape (Figure 10). This master tape will be continually updated as the editing phase of creating a course segment continues. It should be emphasized again that after the initial processing of a TACL program (by INIT) the EDIT software will be used each time a change is made.

Coursewriter II Assembler

The purpose of the CW II assembler, which was written by IBM, is to transform (assemble) CW II source code into an executable module. Many of these modules may be stored on disk simultaneously, thus allowing many students to be signed on to many different courses all at the same time. The CW II assembler is written in 1130 assembler language, and like TSORT, INIT, and EDIT can only be executed when the 1130 system rather than the 1500 system is up.

The input to the CW II assembler when run under the 1130 system generally comes from cards. There is a 1500 system CW II assembler, but as previously mentioned, using it while students are on-line degrades the system (response time increases). The card format is quite rigid, specifying exactly where to punch the opcodes, parameters, and so forth. The other possible input may be from disk if the CW II code is in workfile form. This workfile form is what is produced by both INIT and EDIT. (See figure 10). The output, which is optional, of the CW II assembler is a printer listing of the CW II code. An author using CW II would generally want this; however, an author may use TACL without any knowledge of CW II code.

If the 1500 system assembler is used, the code is assembled and stored on disk immediately. This offers the advantage of being able to execute the code as soon as it is entered. To get a source listing, however, requires the use of a disassembler called LSTCSY. This would be executed at a later time by the 1130 system.

Error Recovery

As in most computer software, the processing of errors in TACL created some problems--many of them involve a decision as to how much to assume or how strictly the rules must be followed. The general approach was to separate mistakes into two categories: errors or warnings. If an opcode was undecodable, an error would result, the statement would be ignored, and the code would be flushed to the beginning of the next basic syntactic unit (i.e., a new opcode or a line of text). If, however, the error was logical rather than syntactic and processing could continue, then the appropriate assumptions are made, a warning message is printed, and processing continues. For example, an author might try to create more than 16 lines of print in a single frame. TACL would assume he wanted to proceed to the next frame starting at line zero. A subsequent edit would be needed to eliminate the warning.

Student Performance Tape

An important part of student and course evaluation is done through the use of student records that are kept automatically by the 1500 system during the student mode. Whenever a student makes a specified response, that information may be recorded on a student performance tape. Such information as the number of times a certain question was answered correctly or whether a certain response was ever given as well as many other statistical data may be obtained. This tape will have thousands of pieces of such information on it by the time several students have gone through the course.

Each response is identified by the student number and other information pertinent to the course. When the author looks over the student records he can easily locate where in the course the student or students were when they gave the response(s) he is analyzing. Code to record this information is programmed into the Coursewriter II course by TACL using identifiers such as frame numbers, multiple choice desired, statement number, and ordinal number of the IF statement within a given frame.

TACL Opcodes

In presenting a brief description of the opcodes used in TACL it is hoped that a better understanding of the entire system will result.

Frequently Used Terms. CRT stands for the cathode ray tube. This hardware device is used to present the majority of the course content to the student. It is divided up into 32 rows numbered 0-31 and 40 columns numbered 0-39. A single letter takes up two rows, so only 16 lines of text may appear on the CRT at the same time. There may be 40 characters per row.

CRT-ROW refers to the next row available for text on the CRT.

Image Projector is an addressable slide projector which is program controllable.

Audio Unit is an addressable tape player which is program controllable.

Segment is comparable to a chapter of a book. A course segment is a logical unit of information. An entire CAI course is composed of many segments.

Frame is comparable to a page of a book. A course segment is composed of many frames.

Alternate coded characters are distinct from regular characters. They appear on the CRT as block letters on a white background.

As stated earlier TACL commands are distinguished from text material in that they are preceeded by a cursor ([]).

The list of opcodes in Figure 11 are all those usable in TACL. The CANCEL opcode is handled by TSORT. The second column of opcodes when processed result in equivalent CW II opcodes which will be exe-

CANCEL

BEGIN CW
CLASS
CLOSE IMAGE
DROP
END
ERASE
FRAME
GO TO
IF
KEYBOARD
LABEL
LIGHT PEN
PAUSE
PLAY AUDIO
POSITION AUDIO
POSITION IMAGE
REPEAT
RESUME AUDIO
SHOW IMAGE
SKIP
TRANSFER
UN
replacement statements
text

INSERT
DELETE
REPLACE
MOVE
COPY

Fig. 11. TACL Opcodes.

cuted during student mode. The opcodes in the third column are used to define the editing to be done. They are used in conjunction with the non-edit opcodes.

CANCEL. This instruction is used during the author mode to cancel the previous instruction that was entered. It may not be used to cancel instructions entered before the latest one. For example:

```
[] SKIP 16;
[] CANCEL
```

The SKIP 16 would not be put on the disk for processing. It is the TSORT software that actually processes the CANCEL instruction.

Regular TACL Commands

The following descriptions are of the TACL opcodes which define the logic and text of the CAI course being designed. These commands will result in CW II code which is operationally equivalent to the TACL definition.

FRAME. This opcode is one of the most often used. It logically means that the code that follows will define a new "page" of information. The author would design his course by presenting the course material frame by frame with branching techniques built in which cause different students to see different frame sequences when the code is executed by the student. Some students might see all ten of ten consecutive frames, while others may skip frames six, seven and eight due to the content of their responses to questions asked by the author (in his course). The FRAME command causes several things to happen during the execution of both INIT and EDIT. CRT-ROW is reset to zero, a label which identifies the new frame is created, and many of the variables used in the software are reset or incremented.

The labels that are generatad correspond to the segment identifier given by the author. This information is shown in the lower right hand corner of every frame to aid in tracing students, debugging, and editing.

The FRAME opcode can also denote a restart point. This means that when a student signs off, the system will keep track of the last frame that he passed through that was declared a restart point. The next time that he signs on, he will start at the restart point that was saved. Restart frames are declared during the author mode while using AUTHOR.

LABEL:cccc. Use of this opcode enables the author to create his own frame labels within a course segment.

If a label is defined that is just a single letter (A-Z), it means that a sub-label for the current frame is to be created. Thus, LABEL: C will result in a label consisting of the present frame label concatenated with the letter C. For example:

[] LABEL: TOP results in TOP

[] LABEL: B results in FA13B assuming the user id is FA and the current frame number is 13

Lables defined as in the first example may consist of 2 to 4 letters.

GO T There are four forms of this opcode:

1. [] GO TO label

2. [] GO TO letter

3. [J GO TO NEXT

4. [] GO TO #

All forms cause an unconditional branch to some label defined within the current course segment.

Label is a label that was created through the use of the LABEL command.

Letter is a single letter (A-Z) that is used to define subsections of a frame.

NEXT means the next frame. Editing may change the frame number, but it will not change the "next" relationship between a frame and whatever comes next. That is, if a new frame is inserted between two frames, say frames 10 and 11, the new frame now becomes frame 11 when processed and is the "next" frame in relationship to frame 10.

# refers to a frame number.

The author initially knows what numbers are assigned to each frame by referring to his coding forms. After the first processing by INIT, he will use the source listing to determine this. For example:

```
[] GO TO TOP
[] GO TO A
[] GO TO NEXT
[] GO TO 27
```

IF. There are three different formats for the IF command.

1. Use with light pen responses:

```
[] IF (c................c) GO TO label
```

where: c...........c can be from 1 to 8 consecutive alternate coded characters that define the possible contents a lighted portion of the screen where the student is to point using the light pen. This lighted portion must have been defined (in a line of text since the occurence of the last light pen opcode) by the use of one or more consecutive alternate codes. Their position in a specific line of text defines exactly where they are. Each must be different from all others used in a given question so that TACL may distinguish one from the other; e.g., IF (TRUE) GO TO NEXT. If the student points to the correct area of the CRT (where TRUE is), the GO TO will be executed.

2. Use with the keyboard responses:

[] IF (CLASS # op CLASS # op.......CLASS #) GO TO label

or

[] IF (word) GO TO label

where: op is either | for a logical OR (disjunctive) or, & for a logical AND (conjunctive). Word = some single word answer. Both | and & have the same precedence. Parentheses may be used and have the same meaning as in an algebraic expression. The number of CLASSES that may be used within a single IF statement is limited only by an overall limit of 100 letters for the entire IF; e.g., IF (CLASS 1 | CLASS 2) TO TO D; IF (CLASS 3 & CLASS 4 | CLASS 1) GO TO BOT; IF (yes) GO TO NEXT. If the student types the correct response, the GO TO will be executed.

For more information on this format, see the description of the CLASS opcode.

3. Use with variables and constants:

[] IF (ID .op. ID) GO TO label

where: ID is a predefined variable or a numeric integer constant as defined in replacement statements. op is EQ,NE,GT,GE,LT, or LE that have their standard meanings; e.g., IF (N.EQ. 3) GO TO 13; IF (SUM.LT. 100) GO TO HEAD.

TRANSFER seg. When a student completes a segment (chapter) of a course, he may logically want to continue with another segment (chapter) of the same course. This requires the TRANSFER instruction. While GO TO is for local branching between instructions within a given course segment, TRANSFER allows for the branching from one segment to another. The course author would put a transfer instruction at the end of a given segment in order for execution to continue in logical order to the next segment.

ERASE. This command has three formats:

1. [] ERASE
2. [] ERASE R_1
3. [] ERASE R_1-R_2

ERASE causes the entire CRT to be erased or blanked out.

ERASE R_1 causes rows R_1 and R_1+1 to be erased. That is, one line of text is erased on the CRT starting at row R_1.

ERASE R_1-R_2 causes the CRT to be erased from row R_1 through row R_2 inclusive.

The ERASE instruction might be used to blank out the top portion of the screen in order to display more text information without using the FRAME opcode.

SKIP #. This opcode advances CRT-ROW to #. The next available row on the CRT becomes #. SKIP is used for spacing between lines of text. For example:

```
[] FRAME
   This is a line of text.
[] SKIP 10
   This text starts in row 10.
```

SKIP might also be used to skip up the screen. For example:

```
[] PAUSE 10
[] ERASE 20-32
[] SKIP 20
```

PAUSE. The two forms of this opcode are:

1. [] PAUSE
2. [] PAUSE #

PAUSE causes the flow of the course (in student mode) to pause until the student presses the space bar to continue.

PAUSE # results in a pause of # seconds.

This opcode might be used to allow the student to take whatever time is necessary for him to read a paragraph of text on the CRT or to study a graph on the image projector. It could also be used to stagger the presentation of the text.

LIGHT PEN. Similar to KEYBOARD, this opcode informs the system that a light pen response will be required of the student. Text lines are scanned for alternate coded areas defining the different choices. The coordinates of these areas must be saved in order to know if the student points to the right (or wrong) area of the CRT.

KEYBOARD. This command is used to inform the system that the author will be asking the student to construct a keyboard response within the next few instructions. The system action is to check all text lines up to the next IF statement for a cursor (not including the leading cursor). If one is found it implies that the response is to be constructed within a line of text. The absence of a cursor before the next keyboard IF opcode implies that the author desires to have the student construct his responses on the next available row on the CRT.

CLASS. CLASS #(____op____op____op. . .____)

where:
op is either | for a logical OR (disjunctive)
or, & for a logical AND (conjunctive)

| and & have the same precedence and are processed from left to right.

= 1, 2, . . . , 6

This opcode is used to define a class of answers. The author might think of a class of correct answers or a class of expected wrong answers that a student might make as a response to a question. For example:

[] CLASS 1 (Bat | Ball | Glove)

[] CLASS 2 (one | 1 | won)

[] CLASS 3 (Boy & Girl)

[] CLASS 4 (2 | 3 & Two | Three)

These classes may then be used in IF statements to control the flow of a course according to the student's response.

TACL also provides for defining approximate or partial answers. For example, within a class an acceptable answer may be given as only the first few letters of the exact answer. There are many modifications of this feature which are inherited from CW II and are still possible in TACL.

For example: [] CLASS 5 (CA*)

means to accept any response beginning with the letters CA.

REPEAT. This causes the last question to be re-asked. It is often used when the student was asked a question and failed to answer it in a way that would cause him to continue or to get remedial instruction. Consequently he is asked to make another response in an attempt to determine if he knows the answer. The system erases his incorrect response from the CRT, and the last student-system interaction is repeated. For more details see the description of the UN opcode.

UN. This opcode is used to define what to do if the student responds with an unrecognizable answer. That is, if his response is not included in an IF statement, any code following the UN and before the next REPEAT opcode will be executed (including TACL commands). For example:

```
[] N = 0
   Please type your answer here: []
[] IF (CLASS 4) GO TO BB
[] IF (CLASS 1 | CLASS 2) GO TO NEXT
[] UN
   Sorry, I do not understand your response.
   Please try again.
[] REPEAT
[] UN
   You still haven't got it.
   Think about it and try again.
[] REPEAT
[] UN
[] N = N+1
[] IF (N.GT.2) GO TO BB
   NO! NO! But don't give up.
[] REPEAT
[] LABEL: BB
   The answer is 96. You'll get it the next time.
[] GO TO NEXT
```

Explanation: If the student's first response is not contained in CLASS 4 or CLASS 1 or CLASS 2, the first UN-REPEAT combination will be executed. Thus, he will receive the feedback, "Sorry, I do not understand your response. Please try again." He then will get another chance to answer. If he is wrong again, the second UN-REPEAT combination will be executed and he will receive the feedback, "You still haven't got it. Think about it and try again." The student then gets another chance to answer correctly. If he is wrong a third time, the third UN-REPEAT combination will be executed. This will cause N to be increased by 1. If N is greater than 2, execution will continue at label BB. If N is not greater than 2, the feedback, "No! No! But don't give up." will appear on the CRT. He will then get another chance to respond. Thus, it would be the fifth incorrect answer that would result in a branch to label BB. At anytime, if his response was a member of CLASS 4 or CLASS 1 or CLASS 2, execution would continue at the "NEXT" frame.

Replacement Statements. Replacement or assignment statements are used to manipulate the numeric values assigned to variables. Variables are defined simply by using them in a replacement statement. A variable name may be from one to four characters long starting with a letter. An author may use up to 30 variables, but he will be warned after using 10. (The reasons for this are involved with coursewriter counters.) The variables may contain integer values from -2^{15} to $+2^{15-1}$. Subscripts are not allowed.

```
Format:  VAR = VC
         VAR = VC op VC

Where:   VAR denotes a variable and
         VC denotes a variable or an integer constant

Where:   op = +,_,*,or/

And:     VAR is defined by 1 to 4 characters, the first being
         alphabetic

E.G.:    [] N = 0
         [] NUM = NUM+1
         [] AVG = SUM/N
```

In course writing, variables are used for counters and keeping score. For example, if a student repeatedly responded with unrecognizable answers you would want to branch to some different coding. In that case you could put a counter (variable) within the UN-REPEAT code. If part of the course was a test, you would use a variable to keep track of the student's score.

DROP (var, var, . . var). This opcode is used to inform the system that the variables within the parentheses will not be used anymore. The author is then free to define more variables. Since only ten different variables may be saved over sign-off at any one time, this may sometimes be necessary. For example:

```
[] DROP(N,AVG,SCOR)
```

SHOW IMAGE #. If the correct image # is in position on the image projector, the shutter will simply be opened. If not, the film reel will rewind (or go forward) to image # and then the shutter will open.

POSITION IMAGE #. This will position image # so that it can be shown immediately by the use of a SHOW IMAGE # instruction.

CLOSE IMAGE. This simply means to blank out the screen on the image projector. If the author had been displaying a picture on the screen and was moving on to a new topic in the course, he might issue this command to remove the image from the screen.

PLAY AUDIO #. This command will cause audio message # to be played through the audio unit. (# denotes a number between 1 and 999.)

POSITION AUDIO #. This will result in the beginning of audio message # to be positioned at the playback head. When used wisely this can save a lot of waiting time for the student. If an audio message is not positioned correctly and a PLAY AUDIO # is executed it could take several minutes for the tape to rewind and be positioned correctly for playing.

RESUME AUDIO. This command is used in conjunction with what are called emphasis marks within an audio message. A single audio message may be divided into several parts and played a part at a time.

RESUME AUDIO means, then, to continue the audio message that is in position on the tape recorder.

BEGIN CW. This command signifies that one or more Coursewriter II commands will follow. These coursewriter instructions will not be preceded by a cursor ([]). (If it were not for the BEGIN CW command they would be taken as text.) The next line that begins with a cursor signifies the end of the coursewriter commands. For example:

```
[] BEGIN CW
   dt 12,0/2,0/40,0/How have you been?
   ep 14,4/2,4/26,4//99/GA12
   br GA12D
[] PAUSE
```

The Coursewriter II code will be put on the workfile exactly as they appear with no error checking. If they are in error, they will be caught during assembly by the IBM coursewriter assembler.

END. This signifies the end of a given course segment. As in most languages, it is not executable, but simply marks the physical end and causes termination of the software involved (INIT or EDIT).

Editing Opcodes.

During the editing phase of preparing a course segment, errors are corrected, new materials inserted, old material deleted, or material is moved from one place to another. This enables the author to continually update and revise the course. The opcodes used specifically for editing are:

1. DELETE
2. INSERT
3. REPLACE
4. MOVE
5. COPY

Keep in mind that these opcodes are used in conjunction with the regular TACL commands, in the EDIT mode. Also, the most recent source listing must be [illegible] to obtain correct statement numbers.

DELETE. This command deletes the specified statement or statements from a course. There are two formats:

1. [] DELETE n
2. [] DELETE n_1 - n_2

The first format causes only the single statement with statement number n to be deleted from the code. The second format causes all statements within and including the two specified statement numbers to be deleted.

[] DELETE 11

[] DELETE 25-29

INSERT n. This signifies that some new regular TACL instructions will be inserted after statement n.

e.g. [] INSERT 33. New TACL instructions (commands and/or text) will cause the new instructions to be inserted into the course starting with statement number 34. The amount of new material is restricted only by disk capacity and should not be a problem.

REPLACE. This is a combination of DELETE and INSERT. There are two formats:

1. [] REPLACE n

2. [] REPLACE n_1 - n_2

The first format causes the statement n to be deleted and new TACL to be inserted in its place. The second format causes all of the statements from n_1 to n_2 to be deleted and any number of new TACL instructions to be inserted starting a statement number n.

e.g. [] REPLACE 10-15. new TACL instructions

will result in the deletion of statements 10-15 and the new TACL starting with statement number 10 when EDIT is executed.

MOVE. This command moves one or more statements from one place to another within a course segment. (The effect is the same as taking several cards from one place in a deck and putting them at another place.)

e.g. [] MOVE 6-8,15

Statements 6-8 are inserted after statement 15. Once EDIT is run, statement 9 will be statement 6 and all other statements are readjusted accordingly.

COPY. This command copies one or more statements from one place to another. That is, one or more statements are copies at another location in the course segment; they are not deleted from their original location.

e.g. [] COPY 6-8, 15

Statements 6-8 are inserted after statement 15. That is, after an EDIT run, statements 6-8 in the course will be exactly the same as statements 16-18.

CHAPTER V

IMPLICATIONS

Statistical Summary

The TACL system has been implemented and is being used daily by the Computer Assisted Instruction Laboratory staff at The Pennsylvania State University. To date (April 1973) approximately 40 course segments containing over 100,000 TACL statements have been written completely in TACL and are summarized in Table 5. The largest segment is about 8,000 statements long. Student time has been increased by 4 1/2 hours per day since authors can write in TACL while regular students are signed on to their respective courses. This simultaneous use was not possible when authoring was done in CW II. The current estimated ratio of preparation time to on-line student time is 60 to 1 as compared to 100 to 1 when CW II exclusively was used for curriculum development. (36)

User Acceptance

To evaluate the acceptance and merit of the TACL system, a questionnaire was developed and distributed to those people in the CAI Laboratory at The Pennsylvania State University who were working in various capacities in relation to TACL. This questionnaire (see Appendix 4) listed various characteristics of the TACL system with a rating scale next to each. For analysis, the scale was divided into three parts (1, 2, or 3) with a rating of 1 meaning low or poor, and a rating of 3 meaning high or good.

Table 5

CAI Course Segments Written In TACL

Course Name and Segment ID		Number of TACL Statements	Number of CW II Statements
CARE 2	AA	662	864
CARE 2	AF	1,067	1,483
CARE 2	BA	1,729	2,349
CARE 2	CB	2,386	3,436
CARE 2	CJ	1,082	1,413
CARE 2	CL	1,979	2,588
CARE 2	CU	4,018	5,170
CARE 2	CP	1,606	2,126
CARE 2	DB	1,835	2,592
CARE 2	DM	2,236	3,067
CARE 2	DE	2,543	3,567
CARE 2	EA	4,642	5,586
CARE 2	ED	843	1,088
CARE 2	EM	5,194	6,713
CARE 2	FB	2,547	3,765
CARE 2	GA	1,958	2,372
CARE 2	GM	6,947	9,175
CARE 2	FO	1,475	2,077
CARE 2	FK	4,831	7,124
CARE 2	FM	2,731	3,619
CARE 2	FQ	4,748	6,562
CARE 2	JA	8,857	12,327
CARE 2	HF	5,447	7,588
CARE 2	HJ	2,861	4,009
CARE 2	JM	3,431	4,706
CARE 2	KA	5,032	5,452
CARE 2	KE	5,630	7,335
CARE 2	MA	655	951
CARE 2	ME	484	734
CARE 2	MM	7,302	9,836
CARE 2	NA	2,828	3,860
CARE 2	ND	4,332	5,604
	TOTALS	103,918	139,138

Tables 6, 7 and 8 give a summary of the information from the twelve questionnaires that were completed. In some cases a specific characteristic of TACL may not have been rated by a specific user since certain features pertained only to a certain user category. For example, course authors would not be able to rate how easy or difficult it would be to modify the existing features of TACL.

Table 6 lists the various user categories and their ratings of the characteristics of the TACL system. This table indicates that TACL has a much better acceptance from the author-programmer viewpoint than from the system viewpoint. This was to be expected since many of the operational features were purposely designed to be temporary until TACL was or was not judged to be worthy of full-scale adoption by the Lab.

Table 7 gives a complete summary of the questionnaire without regard to user category. In some cases users rated a characteristic that was actually not in their category. The figures in Table 7 include all the answers given.

Finally, Table 8 gives a summary from only the author, programmer, and input technician points of view. These were the main categories of users for whom TACL was designed. The characteristics of TACL chosen are those pertaining to those users. These characteristics closely parallel the features used to evaluate established languages and the ideal CAI language as given in Tables 2, 3 and 4 in Chapter 3.

A part of the questionnaire was reserved for comments to elaborate on any TACL characteristics or to make any pertinent observations. Some of the comments were:

Table 6

Abridged Mean Rating of TACL Characteristics by Different User Categories

Category	System Analyst	System Operator	System Manager	Input Technician	Instructional Programmer	Course Author	Project Leader
Number in Category	1	1	1	6	5	3	1
TACL Feature/Characteristic				Mean Rating[a]			
Entering TACL	3	3	3	3	3	3	3
Maintaining System	3	2	3	b	b	b	2
Understanding Listing	3	3	3	2.82	3	2.67	3
Communication with Authors	b	b	b	b	2.61	3	3
Debugging	b	3	b	2.82	2.79	2.67	2
Using TACL	3	3	3	3	3	3	3
Learning TACL	3	3	3	3	3	3	3
Editing TACL	3	3	b	2.82	2.79	3	3
Converting to CW II	2	1	2	b	b	b	b
Programming in Various Fields	b	3	b	b	3	3	3
Developing Courses in Various Fields	b	3	b	b	b	3	3
Over-All System Management	2	2	2	b	b	b	b
System Response for Students	b	2	3	b	b	b	b
Communication with Others	b	2	3	2.82	2.79	3	3
Graphics	b	2	b	b	1.59	2	1
Producing TACL Listings	2	1	3	b	b	b	b
Developmental Costs	b	b	3	b	b	b	3
Adding New Features	3	2	3	b	b	b	b
Teaching TACL to Others	b	3	3	2.82	3	3	3
Modifying Existing Features	2	2	3	b	b	b	b
Programming Non-Trivial Courses	b	3	b	b	3	3	3
Over-All Mean	2.64	2.37	2.86	2.89	2.80	2.89	2.73

[a] Obtained by dividing total points received by the number of people in the given category.

[b] This characteristic of TACL is not applicable to this category of user.

Table 7

Summary of TACL Questionnaire

Feature/Characteristic of TACL	Number of Different User Responding	Total Points Received	Mean Rating
Entering TACL Code On-Line	11	33	3.0
Maintaining the TACL System	10	26	2.60
Understanding the Listing	12	34	2.83
Communication With Course Authors	8	22	2.75
Debugging TACL	9	25	2.78
Using TACL	11	33	3.0
Learning TACL	12	36	3.0
Editing & Revising TACL Programs	9	26	2.89
Converting TACL to Executable Form	6	14	2.33
Programming in Various Fields	8	24	3.0
Developing Courses in Various Fields	9	26	2.89
Over-All System Management	5	12	2.40
System Response for Students	5	12	2.40
Communication with Other User Categories	11	31	2.82
Programming Graphics	8	15	1.88
Producing TACL Listings	8	18	2.25
Cost of Course Development	2	6	3.0
Adding New Features to TACL	9	23	2.56
Teaching TACL to Others	11	32	2.91
Modifying Existing TACL Features	7	18	2.57
Developing Non-Trivial CAI Courses	8	24	3.0
Totals	179	490	2.74

Table 8

Summary of Author, Programmer, and Input Technicians Evaluation of TACL Characteristics as Obtained from the Questionnaire

Feature/Characteristic of TACL	Number of Different User Responding	Total Points Received	Mean Rating
Entering TACL Code On-Line	11	33	3.0
Understanding the Listing	12	34	2.83
Debugging TACL Programs	9	25	2.78
Using TACL	11	33	3.0
Learning TACL	12	36	3.0
Editing and Revising TACL Programs	9	26	2.89
Programming Courses in Various Fields	8	24	3.0
Programming Graphics	8	15	1.88
Teaching TACL to Others	11	32	2.91
Developing Non-Trivial Courses	8	24	3.0
Totals	99	282	2.85

"TACL makes the author consider all possibilities in providing feedback for varying responses."

"TACL makes communication between the programmer and the author more precise without hindering the author."

"Constructing text material is very easy."

"The answer processing is much less complicated and easier to use than CW II."

"The source listing produced is very understandable and makes the logic easy to follow."

"It is easy to locate places causing the student difficulty since the frame number appears in the lower right-hand corner of the CRT."

"This language has been a boon to our course development."

"Turn-around time of one workday can be restrictive to programmers."

"Immediate execution would be an improvement."

"The system needs improvement operationally."

There were many other comments, but the ones given above are representative of the others. Like Tables 6, 7 and 8 they indicate that TACL is a success from the authoring viewpoint but could use some operational modifications. It should be pointed out that some modifications have already been made and others are being planned that will improve the operational (systems) portion of TACL. For example, the input to INIT and EDIT now comes from disk instead of tape. Also, if desired, the TACL listing can be produced on an IBM 360 system with high-speed printing capabilities.

Summary of the TACL Benefits

There are many aspects of CAI course design and development that have been improved by the development and use of the TACL system.

1. TACL code is input through a CRT when author is signed on in student mode.

2. As TACL code is being input it may be viewed similar to the way it will actually appear to the student.

3. Course logic and content are integrated into a single set of code.

4. The job of the course programmer is easier and yet more interesting.

5. All aspects of course development may be done using only the IBM 1500 system (with its 1130 capabilities).

6. TACL listings are easily understood and may be used in analizing the course.

7. Course implimentation time is substantially decreased.

8. The power of CW II has not been sacrificed.

Comparison to CW II

Comparing TACL with CW II is probably the most valid way of judging the TACL system. In terms of learning the language, TACL is much easier and faster. CW II, on the other hand, is necessary for using graphics and offers the author immediate execution. An informal study done by Nancy Enteen and Lynn Yeaton of the CAI Lab at Penn State showed that it takes approximately 1/3 the time to write a program in TACL as it would to write an operationally equivalent program in CW II. They also found that it takes, on the average, 1/3 more time to input a CW II program as compared to inputting the equivalent TACL program. Table 5 gives some of the course segments written in TACL and compares the number of TACL statements to the resultant number of CW II statements. The savings gained by using TACL accounts for a part of the reduction in the ratio of course preparation time to on-line student time when using the TACL system instead of CW II.

The Future

The ideal CAI system is still many years away. People must be convinced that CAI works and that it is financially feasible before its wide-spread use will be encouraged and supported. TACL was designed to be a step in this direction. It is non-trivial, easy to learn, easy to use, and open-ended. Initial use has shown that TACL can reduce costs of course development without restricting the course author. The design goals have been met.

There will be more software and hardware developments in the field of computer-assisted instruction. The plasma tube may very well lead to a large central processing unit with many terminals located in various places in a given state or even throughout the nation. Such hardware will necessitate improved software. There will be more CAI programming languages. Perhaps a translator will be written which inputs one of many existing CAI languages and outputs an equivalent program in any one of many languages which are used today.

Whatever developments occur, it is hoped that the knowledge gained through the development and use of the TACL system will be used to make the next CAI authoring language better. If so, computer-assisted instruction will continue to improve and, in turn, education will benefit. That, ultimately, is the real goal.

BIBLIOGRAPHY

BIBLIOGRAPHY

Abelson, P. H. Computer-assisted instruction. Science, Vol. 162, No. 3856, November 22, 1968.

Bitzer, D., & Skaperdas, D. The design of an economically viable large computer based educational system. CERL Report X-5, February 1969.

Block, K. Strategies in computer-assisted instruction: A selected overview. Learning Research and Development Center, University of Pittsburgh, 1970.

Brown, D. The machine of media. Proceedings of the Seventh Annual Conference on Higher Education in California, May 1968.

Cartwright, G. P. Issues in curriculum evaluation theme address. ADIS Meeting, Stony Brook, February 1, 1971.

Cartwright, G. P., & Ward, M. E. Some contemporary models for curriculum evaluation. Paper presented at Association for the Development of Instructional Systems, Quebec City, Quebec, August 1972.

Commission on instructional technology: To improve learning. Washington, District of Columbia, Government Printing Office, March 1970.

Dean, P. Preliminary report on the development of a simplified system for CAI author entry. IBM Corporation, 1969.

Dwyer, F. M. Computer-assisted instruction: Potential for college level instruction and review of research. The Pennsylvania State University, University Division of Instructional Services, 1970.

Galenter, E. H., ed. Automatic teaching: The state of the art. New York: John Wiley and Sons, 1959.

Hall, K. A. Computer-assisted instruction: Status in Pennsylvania. The Pennsylvania State University, College of Education, Report No. R-34, 1971.

Hall, K. A. Inservice mathematics education via CAI for elementary school teachers in appalachia. The Pennsylvania State University, College of Education, Report No. R-26, 1970.

Hall, K. A. Computer-assisted instruction: State of the art. Phi Delta Kappan, June 1971, pp. 628-631.

Hansen D., Dick, W., & Lippert, H. Research and implementation of collegiate instruction of physics via CAI-final project report. Project No. 7-0071, Grant No. OEG-2-7-000071-2024, Florida State University, 1968.

Hickey, A. CAI: A survey of the literature. Entelek, Inc., October 1968.

Hunka, S. CAI--the technical aspects -an educational innovator's viewpoint. From the Conference on Computers in Education, 1969.

International Business Machines. IBM 1500 Coursewriter II, author guide. From IBM Sales Representative.

International Business Machines. IBM 1500 instructional system introduction to CAI and system summary. Available from IBM Sales Representative.

Jerman, M. Characteristics of CAI configurations from an author's viewpoint. Unpublished report.

Jordon, R. R., Romaniuk, E. W., & Birtch, W. Vault manuals. The University of Alberta, Canada, 1969.

Kopstein, F. Humrro professional paper, computer administered versus traditionally administered instruction. The George Washington University, June 1967, pp. 31-67.

Lower, S. K. CAI at Simon Fraiser University. Unpublished paper, 1971.

Mitzel, H. E. The impending instructional revolution. Phi Delta Kappan, April 1970.

Mitzel, H. E. Computers and adaptive education. American Education, December 1970, pp. 23-26.

Newsom, C. V., Chairman. The computer in education. I.D.E.A., Occasional Paper, New York, 1970.

Pakin, S. APL/360 reference manual. Science Research Association, 1968.

Palmer, D., Shea, B., & Cartwright, G. P. Computer assisted remedial education: Early identification of handicapped children--course planning manual. The Pennsylvania State University, Report No. R-42, June 1971.

Report of the National Academy of Engineering. A study of technology assessment. Washington, District of Columbia, Government Printing Office, July 1969.

Report of the National Academy of Sciences. Technology: Processes of assessment and choice. Washington, District of Columbia, Government Printing Office, July 1969.

Rothman, S., & Mosmann, C. Computers and society. Scientific Research Association, 1972.

Silberman, H. F. Application of computers in education. Yale Scientific Magazine, November 1967, pp. 18-21.

Skinner, B. F. The technology of teaching. New York: Appleton-Century-Croft, 1968.

Stetten, K. J. Toward a market success for CAI--an overview of the TICCIT system. Mitre Corporation, Washington, District of Columbia, 1972.

Suppes, P., & Morningstar, M. Computer assisted instruction. Science, Vol. 166, 1969, pp. 343-350.

Watts, J. TACL: A teaching and coursewriting language. Paper presented at the Association for the Development of Instructional Systems, San Francisco, California, January 1973.

Woods, R., et. al. The production and evaluation of three computer-based economics games for the sixth grade. Report No. 3358/W769, Final Report. Board of Cooperative Educational Services (BOCES), Westchester County, New York, 1967.

APPENDIX 1

SAMPLE TACL PROGRAM ON TACL CODING FORMS

Label ______ Audio ______ Image ______

FRAME
SKIP 4

How does socialization bring About this change in the child's awareness of other points of view?

PAUSE 6
PLAY AUDIO 15E
SKIP 18

The ability to take the other persons point of view allows human beings to live together, which is the ultimate goal of socialization.

Label

FRAME

Audio Image

Okay, so societies set up rules for acceptable behavior and children learn the proper behavior from the people around them. But who specifically is responsible for socializing the child?

In our society there are two groups of people who accomplish most of a child's socialization: PARENTS and PEERS (other children of about the same age).

Label — Audio — Image

FRAME	Below is a list of behaviors. Type
KEYBOARD	whether you think the child learns all
	these behaviors __primarily__ from parents
	of peers: Language
	basic self-help skills
Position Audio 16E	SEX-Appropriate behaviors
CLASS 1 = (PAR)	
CLASS 2 = (PEER)	
IF(CLASS1) GoTo A	
IF(CLASS2) GoTo B	

UN	
	Type in one of the two words: __parents__ or
REPEAT	__peers.__

Label ______ Audio ______ Image ______

0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39

LABEL: A
SKIP 24
ERASE 24-31
PAUSE 1

Yes. Parents usually socialize the child in fundamental skill like these.

GO TO NEXT

LABEL: B
SKIP 24
ERASE 24-31
PAUSE 1

24 Not really. Peers may play some part in socializing these skills, but the primary burden is the parents

GO TO NEXT

0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39

Label ______ Audio ______ Image ______

FRAME; LIGHT PEN
PLAY AUDIO 16E
PAUSE 3
RESUME AUDIO
SKIP 5

SUM = 0
IF (A) GO TO B
IF (B) GO TO B
IF (C) GO TO A
IF (D) GO TO B
IF (E) GO TO B

PEERS teach each other:

A eating habits

B control of anger reactions

C cooperation skills

D avoidance of hazards

E how to dress oneself

UN
SKIP 26

PAUSE 3
ERASE 26-29
REPEAT

26 Touch one of the lighted boxes squarely.

Label ______ Audio ______ Image ______

LABEL:A
SKIP 24
ERASE 24-31
PAUSE 1

24 wonderful! The ins and outs of cooperaation are learned primarily from experience with peers.

GO TO CBBH

LABEL:B
SKIP 24
ERASE 24-31
PAUSE 1

24 No, parents usually teach that sort of behavior. Think about what skills would be especially important for getting along with peers.

IF(SUM.GE.1) GO TO NEXT
SUM=1
REPEAT

Label

Audio

Image

LABEL: CBB14
FRAME
BEGIN CW
LD 0/52
SKIP 10

PAUSE
ERASE
SKIP 0

You may remove your earphones.

Label Audio Image

SHOW IMAGE 223E

As the image shows, parents are responsible for teaching the child to help himself and control himself.

Peers, on the other hand, teach each other how to get along with people their own age. This requires that the child has already learned from his parents how to help and control himself to a certain degree.

Label ____________ Audio ____________ Image ____________

FRAME
CLOSE IMAGE

This section on socialization has
been long, and the concept itself is not
easy to grasp. We're going to give you
a short quiz with immediate feedback on
your answers as a review of the impor-
tant points.

APPENDIX 2

TACL SOURCE LISTING

```
STATEMENT  .   TACL                 EE&                         1         2         3
NUMBER         OPCODES                         0123456789012345678901234567890123456789   FRAME NO.   92


          .
 2C33    &FRAME
                                          0C
                                          01
                                          02
                                          03

          .
 2C34    &SKIP 04
          .
 2C35    &CLOSE. IMAGE                    04     -
 2C36                                     05     HCW CCES SCCIALIZATICN BRING ABOUT
                                          06
                                          07
 2C37                                     08  THIS CHANGE IN THE CHILC'S AWARENESS OF
                                          09
                                          1C
 2C38                                     11  OTHER PCINTS CF VIEWC
          .
 2C39    &PAUSE06
          .
 2C4C    &PLAY AUDIC 15E
                                          12
                                          13
                                          14
                                          15
                                          16
                                          17

          .
 2C41    &SKIP 18                         18     -
 2C42                                     19     THE ABILITY TC TAKE THE OTHER PER-
                                          20
                                          21
 2C43                                     22  SON'S PCINT CF VIEW ALLCWS HUMAN BEINGS
                                          23
```

```
                                                    24
  2044                                              25  TC LIVE TCGETHER, WHICH IS THE ULTIMATE
                                                    26
                                                    27
  2C45                                              28  GCAL CF SCCIALIZATICN.
                                                    29
                                                    3C
                                                    31

STATEMENT       TACL                  EE&                         1         2         3
 NUMBER         OPCCCES                                 C123456789012345678901234567890123456789   FRAME NC.    93

  2046     &FRAME                                   00     -
  2047                                              01     CKAY, SC SCCIETIES SET UP RULES
                                                    02
                                                    03
  2048                                              04  FOR ACCEPTABLE BEHAVICR AND CHILCREN
                                                    05
                                                    06
  2049                                              07  LEARN THE PRCPER BEHAVIOR FRCM THE
                                                    08
                                                    09                     -
  205C                                              10  PEOPLE ARCUNC THEM. BUT WHO SPECIFI-
                                                    11
                                                    12
  2C51                                              13  CALLY IS RESPONSIBLE FCR SCCIALIZING
                                                    14
                                                    15          .
  2C52                                              16  THE CHILCC
                                                    17
                                                    18     -
  2053                                              19     IN CUR SCCIETY THERE ARE TWC GROUPS
                                                    2C
                                                    21
  2C54                                              22  CF PECPLE WHC ACCCMPLISH MOST CF A
                                                    23
                                                    24                        . +++++++    +++++
  2C55                                              25  CHILC'S SCCIALIZATICN. PARENTS AND PEERS
                                                    26
                                                    27
  2056                                              28  (CTHER CHILCREN CF ABCUT THE SAME AGE).
                                                    29
                                                    3C
                                                    31
```

```
STATEMENT        TACL                    EE&                          1         2         3
NUMBER           OPCODES                                    0123456789012345678901234567890123456789   FRAME NO.   94

            .
 2057      &FRAME*

            .
 2058      &KEYBOARD                                         00     -                           -
 2059                                                        01      BELOW IS A LIST OF BEHAVIORS. TYPE
                                                             02
                                                             03
 2060                                                        04  WHETHER YOU THINK THE CHILD LEARNS ALL
                                                             05
                                                             06                              BBBBBBBBB.........
 2061                                                        07  THESE BEHAVIORS PRIMARILY          ---------- FROM PARENTS
                                                             08
                                                             09              .
 2062                                                        10  OR PEERS. LANGUAGE
                                                             11
                                                             12
 2063                                                        13              BASIC SELF-HELP SKILLS
                                                             14
                                                             15
 2064                                                        16              SEX-APPROPRIATE BEHAVIORS
            .
 2065      &POSITION AUDIO 16E

            . -----   ---
 2066      & CLASS1=(PAR)

            . -----   ----
 2067      & CLASS2=(PEER)

            .   -----  -----
 2068      &IF(CLASS1)GOTOA

            .   -----  -----
 2069      &IF(CLASS2)GOTOB

            .
 2070      &LN
                                                             17
                                                             18
                                                             19
                                                             20
```

```
                                                21
                                                22
                                                23

         •
2C71     &SKIP 24                               24  -                         •        BBBBBBB.......
2072                                            25  TYPE IN CNE CF THE TWC WORCS. PARENTS       ------- OR
                                                26      BBBBB.....
2C73                                            27  PEERS     -----.
         •
2C74     &REPEAT
         •
2C75     &LABEL  A
                                                28
                                                29
                                                30
                                                31
*****************************************************************************************

         •
2C76     &SKIP 24
         •
2C77     &ERASE24- 31
         •
2C78     &PAUSE01                               24  -   -
2079                                            25  YES. PARENTS USLALLY SCCIALIZE THE
                                                26
208C                                            27  CHILC IN FUNCAMENTAL SKILLS LIKE THESE.
         •     ----
2C81     &COTC NEXT
         •
2C82     &LABEL  B
                                                28
                                                29
                                                3C
                                                31
*****************************************************************************************
```

```
2083    &SKIP 24

        .
2084    &ERASE24- 31

        .
2085    &PAUSE01                          24  -  -        - - -
2086                                      25  NOT REALLY. PEERS MAY PLAY SOME PART IN
                                          26
2087                                      27  SOCIALIZING THESE SKILLS, BUT THE PRI-
                                          28                  +++++++
2088                                      29  MARY BURDEN IS THE PARENTS.                    .
        .    ----
2089    &GOTO NEXT

STATEMENT       TACL              EE&                     1         2         3
 NUMBER         OPCODES               0123456789012345678901234567890123456789   FRAME NO.   95


        .
2090    &FRAME

        .
2091    &LIGHT PEN

        .
2092    &PLAY AUDIO 16E
                                          00
                                          01


        .
2093    &SKIP 02                          02      -----           .
2094                                      03      PEERS TEACH EACH OTHER.
        .
2095    &PAUSE03

        .
2096    &RESUME AUDIO
                                          04
        .
2C97    &SKIP 05                          05    ++B
2098                                      06    /A A EATING HABITS
                                          07
                                          08
                                          09    ++B
2C99                                      10    /B B CCNTRCL CF ANGER REACTICNS
                                          11
                                          12
                                          13    ++B
210C                                      14    /C C CCCPERATICN SKILLS
                                          15
                                          16
                                          17    ++B
2101                                      18    /D D AVCIDANCE CF HAZARDS
                                          19
                                          2C
                                          21    ++B
2102                                      22    /E E HCW TC CRESS CNESELF
        . ---
2103    & SUM=0

        .   ++ -----
2104    &IF(/A)GOTOB

        .   ++ -----
2105    &IF(/B)GOTCB

        .   ++ -----
2106    &IF(/C)GOTOA

        .   ++ -----
2107    &IF(/C)GOTOB

        .   ++ -----


2108    &IF(/E)GOTCB

        .
21C9    &LN
                                          23
                                          24
                                          25
```

```
2110   &SKIP 26                          26  -                                          BBBBBBBBB.......
2111                                     27  TCUCH CNE CF THE LIGHTEC ECXES SCUARELY.           -------

2112   &PAUSEC3

2113   &ERASE26- 29

2114   &REPEAT

2115   &LABEL  A
                                         28
                                         29
                                         30
                                         31
**************************************************************************************************


2116   &SKIP 24

2117   &ERASE24- 31

2118   &PAUSE01                          24  -          . -
2119                                     25  WCNCERFLL1 THE INS ANC CUTS CF CCOPERA-
                                         26
212C                                     27  TION ARE LEARNEC PRIMARILY FRCM EXPER-
                                         28
2121                                     29  IENCE WITH PEERS.
              ----
2122   &GCTC CBBH

2123   &LABEL  B
                                         3C
                                         31
**************************************************************************************************
```

```
2124     &SKIP 24

         .
2125     &ERASE24- 31

         .
2126     &PAUSE01                                  24  -


2127                                               25  NO, PARENTS USUALLY TEACH THAT SORT OF
                                                   26
2128                                               27  BEHAVICR. THINK ABCUT WHAT SKILLS WOULD
                                                   28
2129                                               29  BE ESPECIALLY IPPCRTANT FOR GETTING
                                                   30
2130                                               31  ALONG WITH PEERS.
         .  ---  --   --------
2131     &IF(SUM.GE.1)GOTONEXT

         . ---
2132     & SUM=1

         .
2133     &REPEAT
         .     . ----
2134     &LABEL  CBBF
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
STATEMENT        TACL            EE&                         1         2         3
 NUMBER         OPCODES                    0123456789012345678901234567890123456789   FRAME NO.   96


         .
2135     &FRAME*
         .
2136     &BEGIN CW
                 +
2137     LC  0/S2
```

```
                                          0C
                                          01
                                          02
                                          03
                                          04
                                          05
                                          06
                                          07
                                          08
                                          09


 2138    &SKIP 1C                         10  -
 2139                                     11  YOU MAY REMCVE YCUR EARPHONES.

 2140    &PAUSE

 2141    &ERASE
                                          12
                                          13
                                          14
                                          15
                                          16
                                          17
                                          18
                                          19
                                          2C
                                          21
                                          22
                                          23
                                          24
                                          25
                                          26
                                          27
                                          28
                                          29
                                          30
                                          31
****************************************************************************************************
```

```
 2142    &SKIP OC

 2143    &SHOW IMAGE 223E                   0C    -
 2144                                       01    AS THE IMAGE SHOWS, PARENTS ARE

                                            02
                                            03
 2145                                       04  RESPCNSIBLE FCR TEACHING THE CHILC TO
                                            05
                                            06
 2146                                       07  HELP HIMSELF ANC CONTROL HIMSELF.
                                            08
                                            09
                                            10    -                    B
 2147                                       11    PEERS, ON THE OTHER HAND  , TEACH
                                            12
                                            13
 2148                                       14  EACH CTHER HOW TO GET ALONG WITH PEOPLE
                                            15
                                            16                B-
 2149                                       17  THEIR CWN AGE.  THIS REQUIRES THAT THE
                                            18
                                            19
 2150                                       20  CHILD HAS ALREACY LEARNEC FROM HIS PAR-
                                            21
                                            22
 2151                                       23  ENTS HOW TO HELP AND CONTROL HIMSELF
                                            24
                                            25
 2152                                       26  TC A CERTAIN CEGREE.

STATEMENT       TACL              EEE                  1         2         3
 NUMBER        OPCOCES                    0123456789012345678901234567890123456789   FRAME NO.    97

 2153    &FRAME.
```

```
2154   &CLOSE IMAGE                  00    -
2155                                 01     THIS SECTICN CN SCCIALIZATION HAS
                                     02
                                     03
2156                                 04  BEEN LCNG, ANC THE CONCEPT ITSELF IS NOT
                                     05
                                     06                B-
2157                                 07  EASY TC GRASP.   WE'RE GOING TC GIVE YCU
                                     08
                                     09
2158                                 1C  A SHCRT CUIZ WITH IMMEDIATE FEECBACK ON
                                     11
                                     12
2159                                 13  YOUR ANSWERS AS A REVIEW OF THE IMPOR-
                                     14
                                     15
2160                                 16  TANT PCINTS.
                                     17
                                     18
                                     19
                                     20
                                     21
                                     22
                                     23
                                     24
                                     25
                                     26
                                     27
                                     28
                                     29
                                     30
                                     31
```

APPENDIX 3

CW II PROGRAM GENERATED BY TACL

```
3210  CB92
3211        1        CTI 30,36+/2,30+/4,36+/+S+P*E
3212        2        EPI 30,38+/2,30+/1,39+/+/1+/*E
3213        3        AA  (8+/)ZZ*E
3214        4        PR  *E
3215        5        DE  0+/32*E
3216        6        DTI 30,29+/2,30+/6,29+/CB92       *E
3217        7        FPO *E
3218        8        CT  4,0+/2,4+/40,0+/    (H)OW DOES SCCIALIZATICN BRING ABCUT *E
3219        9        DT  7,0+/2,7+/40,0+/THIS CHANGE IN THE CHILC'S AWARENESS CF *E
3220       10        CT  10,0+/2,10+/40,0+/OTHER POINTS CF VIEW(/ *E
3221       11        PA  60*E
3222       12        FN  PSREMV+/  +/C(=)01,004L+,02+,T(=)492L+/C(=)01,0C5L+,C4+,T(=)541L*E
3223       13        AUP CB15*E5864,1+/234*E
3224       14        DT  18,0+/2,18+/40,0+/    (T)HE ABILITY TC TAKE THE CTHER PER- *E
3225       15        CT  21,0+/2,21+/40,0+/SON'S POINT OF VIEW ALLCWS HUMAN BEINGS *E
3226       16        DT  24,0+/2,24+/40,0+/TC LIVE TCGETHER, WHICH IS THE ULTIMATE *E
3227       17        CT  27,0+/2,27+/40,0+/GOAL CF SCCIALIZATICN.*E
3228  CB93
3229        1        DTI 30,36+/2,30+/4,36+/+S+P*E
3230        2        EPI 30,38+/2,30+/1,39+/+/1+/*E
3231        3        AA  (8+/)ZZ*E
3232        4        PR  *E
3233        5        DE  0+/32*E
3234        6        DTI 30,29+/2,30+/6,29+/CB93       *E
3235        7        DT  0,0+/2,0+/40,0+/    (O)KAY, SO SCCIETIES SET UP RULES*E
3236        8        DT  3,0+/2,3+/40,0+/FOR ACCEPTABLE BEHAVICR AND CHILCREN*E
3237        9        DT  6,0+/2,6+/40,0+/LEARN THE PROPER BEHAVICR FRCM THE*E
3238       10        DT  9,0+/2,9+/40,0+/PEOPLE AROUND THEM. (B)UT WHC SPECIFI-*E
3239       11        CT  12,0+/2,12+/40,0+/CALLY IS RESPCNSIBLE FCR SCCIALIZING*E
3240       12        DT  15,0+/2,15+/40,0+/THE CHILD(/*E
3241       13        DT  18,0+/2,18+/40,0+/    (I)N OUR SCCIETY THERE ARE TWC GRCUPS*E
3242       14        CT  21,0+/2,21+/40,0+/OF PECPLE WHO ACCCMPLISH MCST CF A*E
3243       15        DT  24,0+/2,24+/40,0+/CHILD'S SCCIALIZATICN(& +P+A+R+E+N+T+S )ANC +P+E+E+R+S*E
3244       16        DT  27,0+/2,27+/40,0+/(9)OTHER CHILCREN CF ABCUT THE SAME AGE(0). *E
3245  CB94
3246        1        DTI 30,36+/2,30+/4,36+/+S+P*E
3247        2        EPI 30,38+/2,30+/1,39+/+/1+/*E
3248        3        AA  (8+/)ZZ*E
3249        4        PRR *E
3250        5        BR  ON+/S31+/1*E
3251        6        DE  0+/32*E
3252        7        DTI 30,29+/2,30+/6,29+/CB94       *E
3253        8        DT  0,0+/2,0+/40,0+/    (B)ELOW IS A LIST CF BEHAVICRS. (T)YPE *E
```

```
3254      9     DT  3,0+/2,3+/40,0+/WHETHER YOU THINK THE CHILC LEARNS ALL*E
3255     10     DT  6,0+/2,6+/40,C+/THESE BEHAVIORS PRIMARILY*B*B*B*B*B*B*B*B(---------- )FRCM PARENTS*E
3256     11     CT  9,0+/2,9+/40,0+/OR PEERS(& )LANGUAGE*E
3257     12     DT  12,0+/2,12+/40,0+/           BASIC SELF-HELP SKILLS*E
3258     13     DT  15,0+/2,15+/40,0+/           SEX-APPRCPRIATE BEHAVIORS *E
3259     14     AU  CB16*E5979*E
3260     15     EP  17,0+/2,17+/40,0+/+/40+/ CB94      *E
3261     16     FN  ED+/+/,D+/+/ +/,+/.+/(6+/)-+/(&+/)&+/(/+/)/+/*R+/*I+/*B+/*C*E
3262     17     AA  (8+/)ZZ*E
3263     18     LC  PAR +/B1*E
3264     19     FN2 COD+/1,1*E
3265     20     FN2 SQ+/AA+/(99)1(00*E
3266     21     BR  CB94A*E
3267     22     AA  (8+/)ZZ*E
3268     23     LD  PEER  +/BI*E
3269     24     FN2 COD+/1,1*E
3270     25     FN2 SQ+/AB+/(99)1(00*E
3271     26     BR  CB94B*E
3272     27     UN  UA*E
3273     28     DT  24,0+/2,24+/40,0+/(T)YPE IN ONE OF THE TWO WCRDS(& )PARENTS*B*B*B*B*B*B*B(------- )CR*E
3274     29     DT  26,0+/2,26+/40,0+/PEERS*B*B*B*B*B(-----).*E
3275     30     BR  RE*E
3276     31     EA  *E
3277  CB94A
3278      1     DE  24+/6*E
3279      2     DTI 30,0+/2,30+/28,0+/*E
3280      3     PA  10*E
3281      4     DT  24,0+/2,24+/40,0+/(Y)ES. (P)ARENTS USUALLY SCCIALIZE THE*E
3282      5     DT  26,0+/2,26+/40,0+/CHILD IN FUNDAMENTAL SKILLS LIKE THESE. *E
3283      6     BR  CB95*E
3284  CB94B
3285      1     DE  24+/6*E
3286      2     CTI 30,0+/2,30+/28,0+/*E
3287      3     PA  10*E
3288      4     DT  24,0+/2,24+/40,0+/(N)OT REALLY. (P)EERS MAY PLAY SOME PART IN *E
3289      5     DT  26,0+/2,26+/40,0+/SOCIALIZING THESE SKILLS, BUT THE PRI-*E
3290      6     DT  28,0+/2,28+/40,0+/MARY BURDEN IS THE +P+A+R+E+N+T+S. *E
3291      7     BR  CB95*E
3292  CB95
3293      1     CTI 30,36+/2,30+/4,36+/+S+P*E
3294      2     EPI 30,38+/2,30+/1,39+/+/1+/*E
3295      3     AA  (8+/)ZZ*E
```

```
3296      4       PR  *E
3297      5       DE  0+/32*E
3298      6       CTI 30,29+/2,30+/6,29+/CB95      *E
3299      7       FN  PSREMY+/  +/C(=)01,004L+,02+,T(=)492L+/C(=)01,CC5L+,04+,T(=)541L*E
3300      8       AUP CB16*E5979,0+/234*E
3301      9       DT  2,0+42,2+/40,0+/          (PEERS )TEACH EACH CTHER(& *E
3302     10       PA  30*E
3303     11       FN  PSREMV+/  +/C(=)01,006L+,02+,T(=)543L*E
3304     12       AUP ,0+/28*E
3305     13       AC  C54+/C53*E
3306     14       DT  5,0+/2,5+/40,C+/ +/+A*BA EATING HABITS *E
3307     15       DT  9,0+/2,9+/40,0+/ +/+B*BB CONTROL OF ANGER REACTICNS*E
3308     16       DT  13,0+/2,13+/40,0+/ +/+C*BC COCPERATICN SKILLS*E
3309     17       DT  17,0+/2,17+/40,0+/ +/+D*BD AVCIDANCE CF HAZARDS*E
3310     18       DT  21,0+/2,21+/40,0+/ +/+E*BE HOW TO DRESS CNESELF*E
3311     19       LC  0+/C30*E
3312     20       EPP +/ CB95     *E


CARE2-4     03/27/73


3313     21       AAP 4,4,2,1+/AA*E
3314     22       BR  CB95B*E
3315     23       AAP 4,8,2,1+/AB*E
3316     24       BR  CB95B*E
3317     25       AAP 4,12,2,1+/AC*E
3318     26       BR  CB95A*E
3319     27       AAP 4,16,2,1+/AD*E
3320     28       BR  CB95B*E
3321     29       AAP 4,2C,2,1+/AE*E
3322     30       BR  CB95B*E
3323     31       UN  UA*E
3324     32       DT  26,0+/2,26+/40,0+/(T)OUCH ONE OF THE LIGHTED BCXES SCUARELY.*B*B*B*B*B*B*B*B*B(-------  *E
3325     33       PA  30*E
3326     34       DE  26+/4*E
3327     35       BR  RE*E
3328     36       EA  *E
3329  CB95A
3330      1       DE  24+/6*E
3331      2       CTI 30,0+/2,30+/28,0+/*E
3332      3       PA  IC*E
3333      4       DT  24,0+/2,24+/40,0+/(W)ONDERFCL(6 T)HE INS AND CUTS CF CCCPERA- *E
```

```
3334         5         DT   26,0+/2,26+/40,0+/TION ARE LEARNED PRIMARILY FRCM EXPER-*E
3335         6         D:   28,0+/2,28+/40,0+/IENCE WITH PEERS. *E
3336         7         BR   (CBBH*E
3337   CB95B
3338         1         DE   24+/6*E
3339         2         DTI  30,0+/2,30+/28,0+/*E
3340         3         PA   10*E
3341         4         DT   24,0+/2,24+/40,0+/(N)O, PARENTS USUALLY TEACH THAT SCRT OF*E
3342         5         DT   26,0+/2,26+/40,0+/BEHAVIOR. (T)HINK ABCUT WHAT SKILLS WCULD *E
3343         6         DT   28,0+/2,28+/40,0+/BE ESPECIALLY IMPCRTANT FCR GETTING *E
3344         7         DTI  30,0+/2,30+/28,0+/ALONG WITH PEERS. *E
3345         8         BR   CB96+/C30+/GE+/1*E
3346         9         LC   1+/C30*E
3347        10         BR   RE*E
3348   (CBBH
3349   CB96
3350         1         DTI  30,36+/2,30+/4,36+/+S+P*E
3351         2         EPI  30,38+/2,30+/1,39+/+/1+/*E
3352         3         AA   (8+/)ZZ*E
3353         4         PRR  *E
3354         5         BR   ON+/S31+/1*E
3355         6         DE   0+/32*E
3356         7         CTI  30,29+/2,30+/6,29+/CB96      *E
3357         8         LD   0+/S2*E
3358         9         DT   10,0+/2,10+/40,0+/(Y)OU MAY REMCVE YCUR EARPHONES.*E
3359        10         CTI  30,36+/2,30+/4,36+/+S+P*E
3360        11         EPI  30,38+/2,30+/1,39+/+/1+/*E
3361        12         AA   (8+/)ZZ*E
3362        13         DTI  30,36+/2,30+/4,36+/   *E
3363        14         DE   0+/30*E
3364        15         DTI  30,0+/2,30+/28,0+/*E
3365        16         FPI  123*E
3366        17         DT   0,0+/2,0+/40,C+/   (A)S THE IMAGE SHCWS, PARENTS ARE*E
3367        18         DT   3,0+/2,3+/40,0+/RESPONSIBLE FCR TEACHING THE CHILD TC *E
3368        19         DT   6,0+/2,6+/40,0+/HELP HIMSELF AND CONTRCL HIMSELF. *E
3369        20         DT   10,0+/2,10+/4C,0+/   (P)EERS, CN THE CTHER HANC *B, TEACH*E
3370        21         DT   13,0+/2,13+/40,0+/EACH CTHER HOW TO GET ALCNG WITH PECPLE *E
3371        22         DT   16,0+/2,16+/40,0+/THEIR OWN AGE.  *B(T)HIS REQUIRES THAT THE *E
```

```
3372     23     CT  19,0+/2,19+/40,0+/CHILD HAS ALREADY LEARNED FRCM HIS PAR- *E
3373     24     DT  22,0+/2,22+/40,0+/ENTS HOW TO HELP ANC CCNTRCL HIMSELF*E
3374     25     DT  25,C+/2,25+/40,0+/TO A CERTAIN DEGREE.*E
3375  CB97
3376      1     DTI 30,36+/2,30+/4,36+/+S+P*E
3377      2     EPI 30,38+/2,30+/1,39+/+/1+/*E
3378      3     AA  (8+/)ZZ*E
3379      4     PRR *E
3380      5     BR  ON+/S31+/1*E
3381      6     DE  0+/32*E
3382      7     DTI 30,29+/2,30+/6,29+/CB97      *E
3383      8     FPO *E
3384      9     CT  0,0+/2,0+/40,0+/    (T)HIS SECTICN CN SCCIALIZATICN HAS*E
3385     10     DT  3,0+/2,3+/40,0+/BEEN LONG, AND THE CCNCEPT ITSELF IS NCT*E
3386     11     DT  6,0+/2,6+/40,C+/EASY TC GRASP.  *B(W)E'RE GCING TO GIVE YCU*E
3387     12     DT  9,0+/2,9+/40,0+/A SHORT QUIZ WITH IMMECIATE FEECBACK CN *E
3388     13     DT  12,0+/2,12+/40,0+/YOUR ANSWERS AS A REVIEW CF THE IMPCR-*E
3389     14     CT  15,0+/2,15+/40,0+/TANT POINTS.*E
3390  CB98
3391      1     DTI 30,36+/2,30+/4,36+/+S+P*E
3392      2     EPI 30,38+/2,30+/1,39+/+/1+/*E
3393      3     AA  (8+/)ZZ*E
3394      4     PRR *E
3395      5     BR  ON+/S31+/1*E
3396      6     CE  0+/32*E
3397      7     DTI 30,29+/2,30+/6,29+/CB98      *E
3398      8     DT  0,0+/2,0+/40,0+/1. (W)E HAVE TALKEC ABCLT INCIVICUALITY *E
3399      9     DT  3,0+/2,3+/40,0+/AND HOW CHILDREN CAN BECCME INCREASINGLY*E
3400     10     DT  6,0+/2,6+/40,C+/DIFFERENT FROM EACH CTHER. (A)NC WE HAVE*E
3401     11     CT  9,0+/2,9+/40,0+/ALSO DISCUSSED A PRCCECLRE THAT WCRKS IN  *E
3402     12     DT  12,0+/2,12+/40,0+/THE OPPOSITE DIRECTICN TC MAKE PECPLE *E
3403     13     CT  15,0+/2,15+/40,0+/MORE LIKE EACH CTHER. (W)HAT CC WE CALL *E
3404     14     DT  18,C+/2,18+/40,0+/THAT PRCCEDURE(/ +& *E
3405     15     EPI 18,16+/2,18+/23,16+/+/23+/ CB98      *E
3406     16     FN  ED+/+/,D+/+/ +/,+/.+/(6+/)-+/(&+/)&+/(/+/)/+/*R+/*I+/*B+/*C*E
3407     17     AA  (8+/)ZZ*E
3408     18     LC  SOC +/B1*E
3409     19     FN2 COD+/1,1*E
3410     20     FN2 SQ+/AA+/(99)1(00*E
3411     21     BR  CB98A*E
3412     22     UN  UA*E
3413     23     CT  24,0+/2,24+/40,0+/(N)O. (T)HE CCRRECT ANSWER IS +S+C+C+I+A+L+I+Z+A+T+I+C+N.*E
```

```
3414      24      FN  ANS+/3+/T+/(8+/)   *E
3415      25      BR  CB99*E
3416  CB98A
3417       1      CT  24,0+/2,24+/40,0+/(R)IGHT.*E
3418       2      FN  ANS+/3+/T+/(8+/)   *E
3419       3      BR  CB99*E
3420  CB99
3421       1      DTI 30,36+/2,30+/4,36+/+S+P*E
3422       2      EPI 30,38+/2,30+/1,39+/+/1+/*E
3423       3      AA  (8+/)ZZ*E
3424       4      PRR *E
3425       5      BR  ON+/S31+/1*E
3426       6      DE  0+/32*E
3427       7      DTI 30,29+/2,30+/6,29+/CB99     *E
3428       8      DT  0,0+/2,0+/40,0+/      2. (W)HY DO SCCIETIES HAVE TO SET *E
3429       9      DT  3,0+/2,3+/40,0+/         UP RULES FCR ACCEPTABLE*E
3430      10      DT  6,0+/2,6+/40,0+/         BEHAVICR(/*E
```

APPENDIX 4

TACL EVALUATION QUESTIONNAIRE

Name ______________________

Please complete the following questionnaire in respect to your experience with the present TACL system. If some of the questions are not pertinent simply leave them blank.

EVALUATION OF TACL

Feature/Characteristics of TACL	Rank						Comments
	Low Poor					High Good	
Entering TACL code on-line	1	2	3	4	5	6	
Maintaining the TACL system	1	2	3	4	5	6	
Understanding the listing	1	2	3	4	5	6	
Communication with course authors	1	2	3	4	5	6	
Debugging TACL programs	1	2	3	4	5	6	
Using TACL	1	2	3	4	5	6	
Learning TACL	1	2	3	4	5	6	
Editing and revising TACL courses	1	2	3	4	5	6	
Converting TACL to executable for m	1	2	3	4	5	6	
Programming courses in varied fields of study	1	2	3	4	5	6	

EVALUATION OF TACL (Continued)

Feature/Characteristics of TACL	Rank						Comments
	Low Poor					High Good	
Developing courses in varied fields (i.e., math, education, psychology)	1	2	3	4	5	6	
Over-all system management	1	2	3	4	5	6	
Over-all system response for students	1	2	3	4	5	6	
Communication with others in Lab concerning TACL	1	2	3	4	5	6	
Programming difficult CAI techniques	1	2	3	4	5	6	
Producing TACL listings	1	2	3	4	5	6	
Cost of course development	1	2	3	4	5	6	
Adding new features to the TACL system	1	2	3	4	5	6	
Teaching TACL to others	1	2	3	4	5	6	
Modifying existing features of TACL	1	2	3	4	5	6	
Developing non-trival courses	1	2	3	4	5	6	

Please give your own general comments concerning TACL.