COMPUTER ASSISTED INSTRUCTION
VIA GRAPHICS TERMINALS

by

CATHERINE TROTTER KOSTETSKY, B.S.

August 1974 TR-42

This report constituted in part the author's thesis for the M.A. degree at The University of Texas at Austin, August 1974.

Technical Report 42

Department of Computer Sciences

The University of Texas at Austin

Austin, Texas 78712

ACKNOWLEDGEMENTS

I wish to express my appreciation to my supervising professor, Dr. Terrence W. Pratt for his guidance and assistance. I am also grateful to Dr. A. G. Pearson for his many helpful criticisms and suggestions.

C. T. K.

July, 1974

ABSTRACT

This paper examines how CAI/graphics systems can be developed for educational institutions which have specific graphics capabilities in mind. The diversity of graphics capabilities and the effect they have on the system design is emphasized. Some existing CAI/graphics systems are described in order to give insight into the problem. Next the variety of ways in which both hardware and software components of a CAI/graphics system can be realized is discussed. The hardware components are graphics terminals, central control computer, and communication links; the software components are operating systems, graphics subsystems, CAI/graphics applications, and CAI/graphics author languages. Finally some guidelines for development of a CAI/graphics system are given.

TABLE OF CONTENTS

Chapter		Page
I.	INTRODUCTION	1
	A. Purpose	1
	B. Graphics Capabilities in CAI	2
II.	DESCRIPTION OF EXISTING CAI/GRAPHICS SYSTEMS	6
	A. Brentwood Tutorial Mathematics Program	7
	B. University of Alberta Drill and Practice Mathematics Program	9
	C. University of Texas CAI Program in the Arabic Writing System	11
	D. The University of Illinois PLATO IV Biology Lesson	13
	E. The University of North Carolina Numerical Analysis CAI Program	15
	F. University of California at Irvine Physics Lesson	17
	G. The University of Illinois Geometry Program	19
	H. CATTS: Computer-Aided Training in Trouble-Shooting	21
	I. The University of Connecticut Music Program	23
	J. Massachusetts Institute of Technology Elementary Mathematics Program	25
	K. Point Park College Surgery Lesson	29
III.	THE COMPONENTS OF CAI/GRAPHICS SYSTEMS	32
	A. Hardware	32
	B. Software	52

LIST OF FIGURES

Figure	Page
1	27
2	32
3	47
4	48
5	49
6	50
7	51
8	59

CHAPTER I

INTRODUCTION

A. PURPOSE

The primary objectives of this paper are the following: (1) an analysis of some existing CAI/graphics systems, (2) a discussion of the varieties of and requirements for each component of a CAI/graphics systems, i.e. hardware, software, and courseware, and (3) the formulation of how to select the right components for a desired CAI/graphics system.

One of the main problems encountered in a discussion of CAI/graphics is the lack of agreement among educators and computer scientists as to what class of computer applications can be called "CAI/graphics." In this paper, a CAI/graphics system refers to a single or multi-terminal instructional system, where the terminals used have the capability of presenting "smooth curved pictures" on the screen rather than simply alpha-numeric characters, but do not provide a hard-copy of the interaction. An instructional system refers to the situation where a student sits at a terminal, interacts with the system, and learns, as defined by educators. So CAI as used in this paper the student response, the next appropriate picture is chosen for that student, so that the sequence of pictures and quantity of pictures could vary from student to student. However, at the end of the interaction with the CAI/graphics system, the students all would have achieved at least a minimum level of mastery of the information presented. The CAI/graphics system in this case controls the student's progress through the material based on previously defined rules.

Simulation of some real world situation using a CAI/graphics system can require a strikingly different graphics capability from that mentioned above. In this case, the student might be presented with a picture of a real world situation which can be dynamically changed based on the student's interactions with the system. In essence, the student "sees" the effects of his requested actions on the situation. Oftentimes, the results of his actions are shown to have an adverse effect on the situation. Since the student is still learning about the real-world situation, for the student to see the results of his actions immediately is an invaluable use of graphics.

In teaching quantitative courses using CAI/graphics, the desired graphics capabilities can range from showing a graph of a given equation to letting the student input geometric figures.

for each part of the system. Chapter IV gives some guidelines on the selection of the different parts of the system to suit a particular CAI/graphics requirement.

The types of systems to be described range from static or passive graphics applications to dynamic interactive graphics programs. The systems are presented in order from the simplest to the most complex.

A. BRENTWOOD TUTORIAL MATHEMATICS PROGRAM

This program was one of many implemented by the Institute for Mathematical Studies in the Social Sciences at Stanford University under the direction of Patrick Suppes. (Suppes, 1969) The goal of this program was to examine the feasibility of tutorial CAI in elementary (first and second grade) mathematics. Approximately 400 lessons covering counting, addition, subtraction, sets, and geometry were written.

The Stanford-Brentwood system uses an IBM 1500 computer, which was designed and built by IBM working with Stanford personnel. The 1500 was one of the first tailor-made CAI computers. The hardware included the central computer, IBM 1500, and several student response terminals which consisted of a Philco-Ford CRT unit, keyboard, light pen, film projector, and headphones with microphone. (Wilson, UD) The CRT unit was capable of presenting only alphanumeric characters and simple line drawings. If the student answered a question correctly, either by light pen is the light pen input capability. To use a light pen would be easier for elementary children than to type a response via a keyboard.

B. UNIVERSITY OF ALBERTA DRILL AND PRACTICE MATHEMATICS PROGRAM

This program is similar to the Stanford-Brentwood program, except for some interesting variations in evaluation of the system. (Hicks, 1972) The objective here was to drill each student in addition and subtraction of the numbers 0-9 until 90% accuracy was reached with a response time less than 15 seconds. The problems were generated randomly. The hardware, IBM 1500 with Philco-Ford CRT units, and author language, Coursewriter, were the same as at Stanford. The graphics output types were as follows: a picture of Donald Duck to draw the attention of the student, large numbers in the statement of the problem, stars for correct answers, big X's for incorrect answers, and percent right indicated graphically.

The student inputs his answer via the keyboard. If the student answered incorrectly, the light pen was used. For example, if the student were asked the sum of 3 and 2, and if he answered incorrectly, he would have to use the light pen to point to 3 items and then 2 items.

C. UNIVERSITY OF TEXAS CAI PROGRAM IN THE ARABIC WRITING SYSTEM

This program is slightly more sophisticated than either of the two previously described programs. (Abboud, 1971) Its main objectives were to teach adult students to write Arabic correctly in the shortest possible time, and to maintain interest and motivation in the study of the language. The ability to discriminate sounds from a computer-controlled tape recorder and the ability to read simple material are secondary objectives of this program.

The hardware used was the IBM 1500 system, including a light pen, which was used only to point to correct answers, and an IBM 1510 Instructional Display, which is a CRT for display of alphanumeric information. In addition to the standard alphanumeric characters, the user can define and load three new dictionaries of character sets or images, where each dictionary can contain 128 separate images. These images are defined in a dot matrix 12 dots high and 8 dots wide. These dictionaries of images are one form of "graphics" on this machine. "Graphics" sets can also be defined in the system, but only three sets can be used in any one course written in Coursewriter, the author language used here. Each set can contain 64 pictures which are encoded in a 36 x 16 dot available, such that the task of creating the Arabic character set would have been a trivial task for the author.

D. THE UNIVERSITY OF ILLINOIS PLATO IV BIOLOGY LESSON

This program is different from those previously described because it is a simulation CAI application rather than the more common drill and practice or tutorial CAI program. (Hyatt, 1972) The objective of this simulation is to teach the laws of biological inheritance. The students perform the traditional fruit fly matings experiment in the simulated laboratory environment as displayed on the graphics terminal screen. The graphics output is pictures of the genetically different types of fruit flies, i.e. wingless, red-eyed, etc. The student selects two fly types he would like to mate by pointing to them with the light pen. The next graphics output is a picture of their offspring. The students collect data and maintain logbooks based on their observations of the types of offspring, just as they would do in a normal laboratory. The students can request matings for successive generations since recessive traits do not appear in first generation offspring. This simulated biology laboratory takes only minutes, whereas the traditional biology department the cost of maintaining the actual laboratory equipment. However, the biology department probably incurred at least an equal cost in the computer time used by the students.

The lesson was only one of a CAI course in biology. The use of a graphics terminal in this lesson is justified because many different types of flies can be generated using random numbers and Mendel's Laws of Inheritance. A random-accessed computer-controlled slide projector could not do the job in this case.

E. THE UNIVERSITY OF NORTH CAROLINA NUMERICAL ANALYSIS CAI PROGRAM

This program was used in conjunction with traditional teaching to help develop insights and skills necessary in numerical analysis. (Oliver, 1969) It can be used either as a lecture aid or by individual students. The objective of this program was to provide meaningful illustrations in numerical analysis which would give students an intuitive understanding of the relationship between symbolic and graphic representation of equations.

The graphic outputs were plots of various curves. The students could dynamically change variables in a symbolic equation and have it plotted. The students could see how the change affects the curve by plotting both the original and modified equations simultaneously.

during the final statistical analysis of the post-test. For example, the results of the post-tests of the students in both groups that had similar scores on the pretest were compared. The results showed that the students that used graphics developed a better intuitive understanding of numerical analysis than the control group.

The evaluation technique used in this program is slightly questionable. The experimental group received more hours of instruction than the control group, so that variable alone would seem to justify their better performance on the post-test.

F. UNIVERSITY OF CALIFORNIA AT IRVINE PHYSICS LESSON

This CAI/graphics lesson was used in addition to regular lectures. (Bork, 1971a, 1971b, 1972) Its purpose was simply to demonstrate one concept, the laws of motion, in a college physics course. The program allowed the students to launch a lunar lander from the earth to the moon, where the initial velocity and angle of the launch were the variables chosen by the student. The graphics output was a picture of the earth, the moon, and the path of the currently active lunar lander. The students could see that his lunar lander missed the moon, crashed into the moon's surface, or landed successfully. Depending on author was that the physics talents needed in the non-graphics version of the program were different from those needed in the graphics version. In the non-graphics version, students took the printed output for an attempted lunar landing and calculated the requirements for a more successful launch. In the graphics version of the program, students just relied on the curves to get a more intuitive feeling for the laws of motion. The author "suspects" that the students learn more using the graphics version.

The terminals used in this lesson are limited in their capabilities. For example, selective erasure or animation is not possible. However, these terminals are the cheapest graphics terminals. It seems that insufficient funds was one of the constraints of this project. Therefore, since graphics input is not even used here, a small hard-copy terminal plotter with a teletype is cheaper than the graphics terminals, and would provide the same plots in just slightly longer time.

G. THE UNIVERSITY OF ILLINOIS GEOMETRY PROGRAM

This program was designed to give high school students experience with facts of geometry before introducing them to formal proofs. (Dennis, 1969) A student

The graphics input in this program could have been accomplished in an easier fashion by using an additional piece of hardware, the joystick. The student draws a line by pushing the joystick in the desired direction of the line. Another disadvantage of the graphics hardware here was that circles or curves could not be input, a serious deficiency in a geometry lesson.

H. CATTS: COMPUTER-AIDED TRAINING IN TROUBLE-SHOOTING

The objective of this CAI/graphics program is to teach the students how to trouble-shoot some electronic and mechanical systems. (Landa, 1972) This system did not teach any new concepts, but taught the students how to assimilate their bits and pieces of knowledge into an efficient and effective trouble-shooting process.

The most noteworthy feature of this system is the way in which the author encoded the graphics information to be output during the program. The author was relieved of the tedium of specifying in detail each point on each frame for the entire program. The author was allowed to simply draw on an "electronic tablet" the set of pictures which would be used somewhere in the program. Then the control program for the "electronic tablet" built a file in the central computer which contained a description of

I. THE UNIVERSITY OF CONNECTICUT MUSIC PROGRAM

The objective of this program was dependent on the abilities the student already had acquired. (Heller, 1971) If the student had not yet mastered a musical instrument on which to play his own composition, the student could still compose music, in a nontraditional manner, using the graphics terminal as his input device, and could hear his composition immediately played back. If the student was already familiar with basic music theory, this program could illustrate advanced concepts by graphical means.

The hardware for the system was an 8K PDP-9, an 18-bit machine, and a type 339 CRT with light pen. The student could draw a music graph on the CRT with the light pen. The high and low spots on the graph of the music would represent the high and low notes respectively. Longer plateaus in the graph would represent notes of longer value than short plateaus. The graph could be continuous or broken, and the music would be affected in a similar manner. The pitch of the tone depends on the relative vertical position in the graph. The graph of the music could be played back immediately by converting the graphics information into signals which can drive a device called a CASS (Computer Assisted Synthesizer System) which instruction was inferior, which the author attributed to lack of motivation on the part of the eighth grade vocal group. So CASS instruction was equal to instrumental instruction for the same length of time.

This CAI/graphics system certainly has some powerful features, but in this case, the objective for musically untrained students seemed inappropriate. Why should a fifth grade student learn to perform music on a CASS? In a college level composition class, this system would be of more value.

J. MASSACHUSETTS INSTITUTE OF TECHNOLOGY ELEMENTARY MATHEMATICS PROGRAM

This lesson is one of many which can be learned at MIT's CAI laboratory. (Papert, 1971b, 1971c) The objective here is that by giving students "tools" such as the CRT and drawing commands, he will discover the laws of geometry by drawing a lot of different geometric figures. The educational philosophy employed is that children learn by doing. So rather than having a "canned" CAI program to teach specific things, they have a basic system on which the students write and execute programs. The students learn by debugging their programs and by seeing the results of the program changes they make.

When this routine is called POLY 150 120, an equilateral triangle is drawn with sides 150 units long. When this routine is called POLY 75 60, an equilateral hexagon is drawn with sides 75 units long. The output of these two different calls and the student's desire for creating "interesting" pictures will certainly cause the student to think about why each figure is drawn.

An interesting variation of this routine which produces fascinating results is as follows:

```
TO POLY:  STEP:  ANGLE
1 FORWARD:  STEP
2 LEFT:  ANGLE
3 POLY:  STEP+5:  ANGLE
END
```

Then when POLY 5 90 is executed, the following figure results:

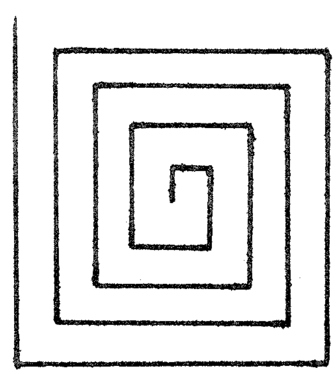

Figure 1

K. POINT PARK COLLEGE

SURGERY LESSON

The objective of this program is to teach medical students surgical techniques by giving them access to an elaborate surgical simulation environment. (Canter, 1971) The student surgeon can conduct a simulated operation and graphically view the progress of his activity.

There were two types of graphics terminals used in this project: 1) an Adage ACT/10 two-dimensional terminal and 2) a full-color 3-D CRT. The output for the 3-D terminal was assembled from thousands of digitized photo images stored on a peripheral device of the main computer. The main computer was time-shared, multi-user system, where each user had access to the two types of CRTs. A light pen and function buttons were used for the input devices.

The graphics interaction would begin by the student selecting an operation he wanted to perform. The picture of the prepared patient is displayed on the 3-D color CRT. On the two-dimensional CRT are animated-simulated graphs showing the patient's heart rate, pH levels, respiration, etc. The student can define the current use of the light pen by pushing a function key, such as scalpel, clamp, sponge, etc. Then when the student touches the light pen to the CRT it performs the

1. Students seem to be motivated more when graphics are used.

2. Lots of information cannot be easily or clearly represented in words, i.e. maps, mathematical graphs, cardiograms, etc.

3. Pictures more fully utilize the student's eyes, which are one of his input devices for learning.

4. "A picture is worth a thousand words."

In the next chapter, the hardware and software components are studied in detail. This includes author languages, courseware, graphics terminals, etc. The various ways in which each component can be realized are discussed.

The function of each part of the system can be performed in a variety of ways. Some of the possibilities for realizing the different parts of the system are described below.

Graphics Terminals

The basic elements of the graphics terminal are the display device, display control, the user interface, programming characteristics, and peripheral storage equipment.

DISPLAY DEVICES: The most commonly used display device is the cathode ray tube (CRT). The basic component of the CRT is an electrode gun which shoots a beam of electrons onto the inner surface of the CRT screen, which is coated with one of several compounds composed of rare-earth elements. These compounds are called phosphors. When electrons bombard the phosphors, they glow for a short time. The length of time the phosphors will glow and the color of the glow is dependent on the compound used for the phosphors. The deflection of the beam of electrons is controlled by two sets of electro-static or electro-magnetic deflectors, which affect the (X,Y) positioning of the beam. One disadvantage of the CRT is that the phosphors must be refreshed or re-bombarded at regular intervals in order to maintain the picture on the screen, otherwise the image is maintained on the wire grid. According to Newman (1973) the disadvantage of the direct view storage tube is that selective erasing is hard to program. Another disadvantage is that the "flooding electrons" gradually charge the entire wire mesh, and the picture that was there becomes invisible after about an hour.

A modification to the direct view storage tube that makes it highly desirable is the use of the raster scan to copy the image on the screen. Instead of using a continuous flood of electrons to copy the grid image to the screen, two beams of electrons scan alternating rows of the grid, continuously copying each point onto the screen in a top to bottom, left to right pattern (raster scan). This interlaced scan reduces flicker. The raster scan technique can also be employed to read the wire mesh, and then a signal can be generated to drive television receivers. The same pattern can be sent to any number of television receivers.

Another device for image storage and viewing is the silicon target tube. The silicon target tube uses a plate coated with silicon as a target instead of a wire mesh as in the conventional direct view storage tube. Generally, the tube is very small and used only for storage. The image is scanned, and the signal generated from the scanning can be used to drive a television

<u>DISPLAY CONTROL</u>: Many of the features provided by some graphics terminals manufacturers, such as character generation, line generation, function keys, etc., are hardwired logic, while other manufacturers provide these features via a display processor. The display processor decodes and executes instructions fetched from either ROM (read only memory) or its own core memory, into which the user has loaded his program. Oftentimes, the ROM has standard subroutines for pre-defined features, such as vector generation, rotations, scaling, etc. The display processors are usually 16 bit computers, sometimes with an elaborate interrupt structure. These display processors usually are coded in assembly language, but some manufacturers provide a FORTRAN compiler.

<u>USER INTERFACE</u>: The user generally interfaces with the graphics program via a <u>keyboard</u>, which provides an alphanumeric character set. The program in turn writes on the graphics terminal screen using the same character set to communicate with the user.

Besides character keys, <u>control keys</u> are often available. The control keys vary from one manufacturer to another. The commonly available control keys include: blank display, backspace one character, horizontal and

User interface devices which lessen the burden of moving the cursor about have been developed. One such device is the joystick, which is a mechanical device similar to a hand lever that moves back and forth and sideways. The lever outputs (X,Y) coordinates each time it is moved. These (X,Y) coordinates are fed back to the terminal and cause the cursor to move appropriately.

Another device that issues (X,Y) coordinates for the cursor position is the mouse, which is a small box on wheels. The mouse is moved about by hand, and each change in its position is translated into (X,Y) coordinates which are sent to the CRT. The cursor on the screen is then appropriately moved to reflect the movement of the mouse. The cursor movement may be done by the display processor or the central computer.

The tablet is another device that is used to control the cursor movement by the user. Similar to the mouse, the tablet transmits (X,Y) coordinates back to the CRT. The user draws on the tablet with a special stylus. The tablet is so constructed that it can detect where the stylus is touching the tablet, and translates that position into (X,Y) coordinates for cursor positioning.

An acoustic tablet uses a stylus that generates a sound that is detected by microphones. The position of enough (X,Y) coordinates to generate points in the pattern of a character. Some CRTs have hardwired dot character generators which use a seven by nine matrix of dots for each character.

Another method of character generation is the shaped beam. The character set available is etched into a thin metal sheet. (Chase, 1970) The program issues a code to the CRT for each character to be written. When a particular character code is received, the electron beam is directed to the corresponding etching of the character which is then copied onto the screen. This approach is limited in that once all of the character etchings have been mounted in the tube, they cannot be changed.

Characters are also generated in graphics terminals by the stroke generator. Short strokes are written to represent a character. The pattern of the strokes for each character is part of the control logic of the terminal. When a character code is received, the control logic directs the beam in the shape of the character.

The dot and stroke generators are the most commonly found in graphics terminals. A separate memory diode matrix or integrated circuit is required to store the and circles. If these features are not available in the hardware, these figures must be drawn by a series of short line segments which approximate the curved figure.

Three-Dimensional Line Generation: Three-dimensional representation of vectors is available in some CRTs. Shading or intensity modulation are available, which makes 3-D pictures more realistic. Manufacturers offer the means to scale pictures. The size of the image is changed without changing its shape.

Peripheral Storage: Many graphics terminals have various peripheral devices for data storage or printing of data. Disks and magnetic tapes are available for an auxiliary storage device which can be used by the program in the graphics terminal. Many systems provide the user with a means of saving data on tape cassettes or on removable disk packs for future use. When the user returns to the terminal at a later time, he can resume his work by mounting the disk pack or tape which has his data. In many instances the user would like to have the information on the graphics terminal printed for future reference. This printed material is referred to as a hard-copy of the graphics terminal image and is produced on various types of printers that can be available at the terminal. Hard-copies can also be made from the graphics terminal display to such devices as microfilm and plotters.

INTERRUPT SYSTEM: The interrupt system of the central control computer directly affects the response time that the student experiences at the graphics terminal. The time that is spent in servicing an interrupt includes the following: 1) latency -- the time it takes the CPU to recognize that an interrupt has occurred, 2) overhead -- the time it takes to find out what device caused the interrupt and to save the status of the computer, 3) the time it takes to execute the specific interrupt routine, and 4) the time it takes to restore the status of the machine.

The three common types of interrupt systems are polling, single level interrupts, and priority interrupts. In polling, the central computer checks each possible interrupt in turn to see if it needs servicing. Whenever an interrupt is active, it is serviced to completion, and then the polling continues. No priority system exists with polling. A single level interrupt system is one in which all interrupts trap to one pre-defined location. In the single level interrupt system, a greater amount of time must be spent in identifying the interrupting device than in polling. A priority interrupt system is one in which the interrupts trap to different locations. Each location has an assigned priority. A higher priority interrupt can interrupt the servicing of a lower priority interrupt.

will choose to have the central computer wired directly to a multiplexor which in turn will be directly wired to all terminals. This type of arrangement alleviates the need for a large number of wired interfaces coming from the computer to each terminal. Generally, only one wired interface is made between the central computer and the multiplexor. A variety of multiplexors are built. There are simple units that buffer one character at a time from one line to complex units with memory. Some of these complex multiplexors are programmable and act as stand-alone computers. See Figure 3.

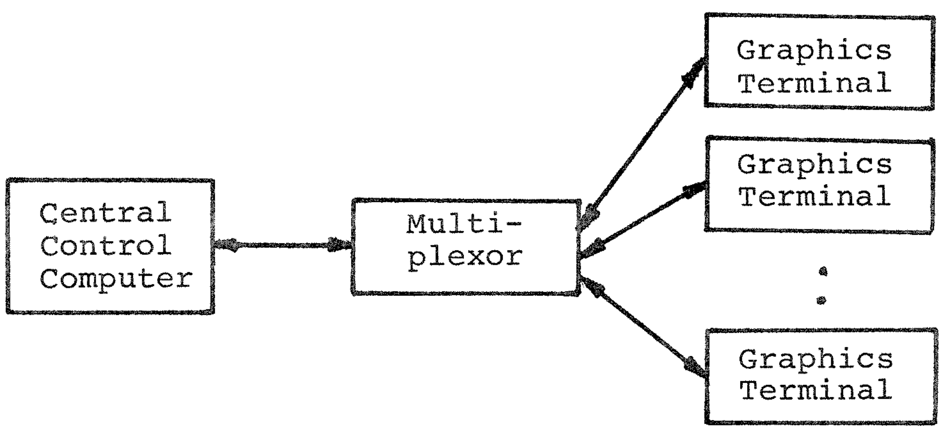

Figure 3

Many computer assisted instructional graphics systems require that the graphics terminals be remote from the central computer.

company. This type of line is limited in the transmission speed to about 300 bits per second because of the switching mechanism used by the telephone company. See Figure 5.

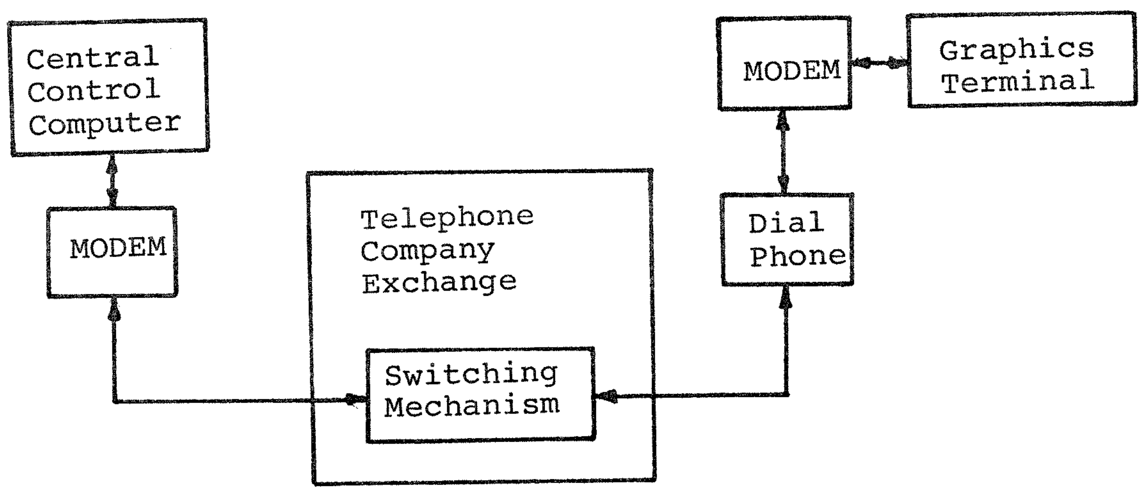

Figure 5

Many graphics systems require faster transmission rates, and therefore use private lines. A private line is an ordinary telephone line that is physically routed around the normal switching mechanisms of the telephone company exchange. This line need not be dialed up; it is always directly linked from the terminal to the central computer. The private line can have rates up to 1800 bits per second. See Figure 6.

As previously described, some graphics systems have multiplexors tied to the graphics terminals. With telephone communications, multiplexors are also used. See Figure 7.

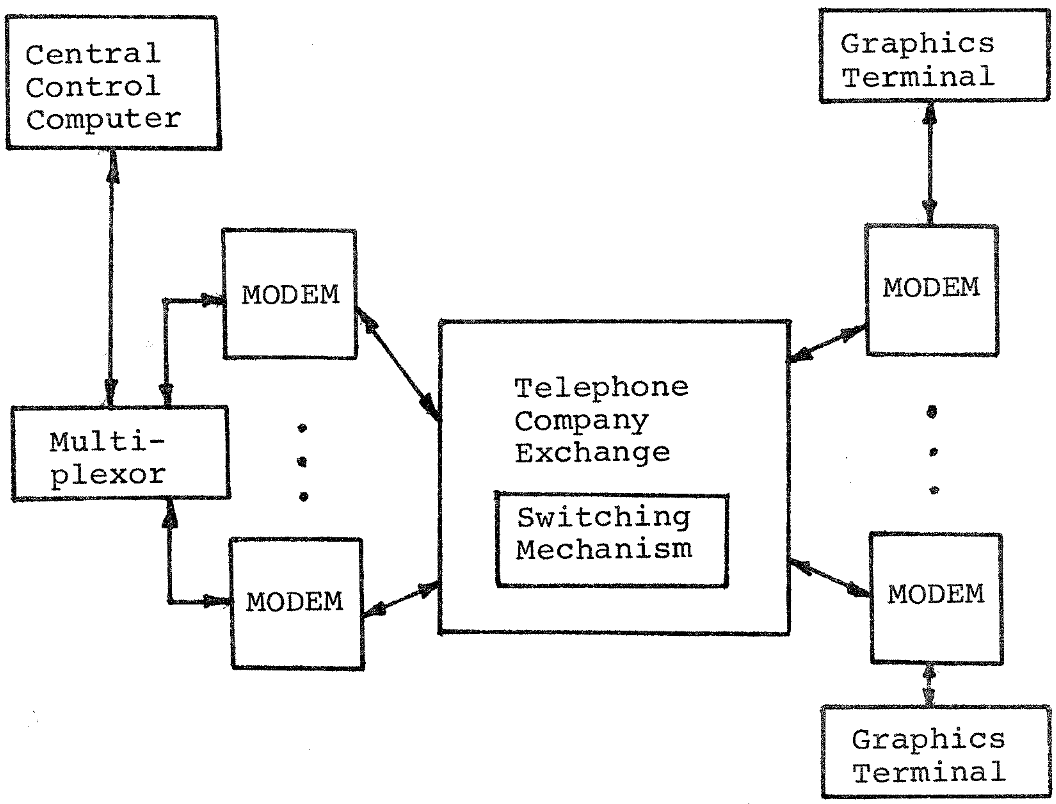

Figure 7

terminals, say two, a different operating system is generally used. This operating system is called a dedicated system and is designed solely for the CAI/graphics application. If one is dealing with only one or two terminals, generalizing the operating system is too costly in computer run-time and in programming effort. The dedicated operating system generally becomes interrelated with the application and is truly a CAI/graphics system. The disadvantage of designing and implementing a dedicated CAI/graphics system is that expansion of the system generally would involve a redesign, whereas in the general purpose system, expansion can be effected more easily.

Graphics Subsystem

The graphics subsystem generally consists of a set of graphics subroutines which control the graphics terminal. Graphics subroutines, such as scaling, zooming, selective erasing, etc. are a part of the graphics subsystem. A general purpose PLOT package is also included in the subsystem. The uniqueness of the interfaces of the graphics terminals requires that special I/O packages be added to the graphics subsystem in order to support each type of terminal in the system. These I/O routines would perform functions such as data formatting, where the data is put into a form acceptable to the graphics terminal.

the assembly language of the central computer is used to generate the courseware, but this approach is generally taken only for a small dedicated graphics system.

Some of the traditional instructional strategies for courseware include: drill and practice, tutorial, inquiry, simulation, and gaming.

Drill and practice CAI programs generally supplement some other type of instruction. New material is not presented in drill and practice programs, but they merely review the concepts presented elsewhere. Drill and practice programs are especially helpful if they are programmed to "learn" the student's weaknesses and then place emphasis on the types of problems with which he has difficulty.

Tutorial CAI programs are intended to be the only mode of instruction that a student receives on a subject. Tutorial programs present factual statements about a subject and then ask questions about the material just presented. Depending on the student's response, remedial branches can be taken. Sometimes the student is asked readiness questions before new material is presented, and if the student responds correctly, he can bypass sections of the lesson. The standard criticism of tutorial mode is that the student is too passive in his participation.

Inquiry CAI programs let the students become more active in the learning process. The student is presented courseware. This approach is taken in an effort to save time in the development of courseware.

CAI/graphics Author Languages

Many different author languages are used to develop courseware. The choice of author language depends on many factors which include the following: the level of sophistication of the author, the availability of the language, the versatility of the student terminal, the student record-keeping capability, and the answer processing features.

The author language can be a compiler-type of language, where the courseware is compiled either in background mode as a batch job, or on an entirely different computer system. The object code of the courseware then runs in the foreground of the central control computer. The language can be interpretive, and then the source code of the courseware is interpreted in the foreground as the terminal user executes the lesson.

The CAI/graphics languages that are used have generally been developed by taking an existing CAI language and attaching to it graphics subroutines which support the unique type of graphics terminal to be used. For example, Coursewriter II supports only the IBM 1510 display terminal; TUTOR IV supports the plasma display terminal. Some

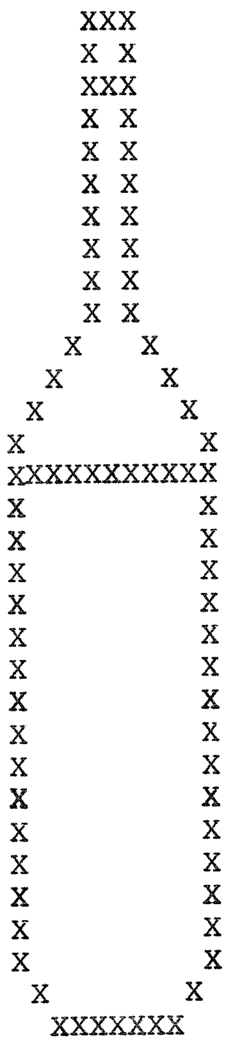

Figure 8

keypunched and stored on the file, where a unique number is assigned to each picture. Then, when the author wants to display a picture, he writes the Coursewriter command: dg rr,cc/xxx, where dg is a mnemonic which indicates display graphic, rr,cc defines the screen location by row and column, and xxx is the unique number of the graphic. The picture cannot be scaled.

Coursewriter does not allow graphic input. A light pen can only be used to point to a picture on the screen. The coordinates of the point at which the student is expected to point are stored by the author as the correct answer, and they are compared against the coordinates of the point at which the student pointed. Coursewriter II was certainly a worthwhile introduction into the CAI/graphics world.

TUTOR is a CAI/graphics language which was developed at the University of Illinois (Avner, 1970 and TUTOR User's Memo, 1973). It is designed to interface with the plasma display terminal. The language has been growing, with respect to its capabilities, since it was

A picture file capability, similar to that of Coursewriter, is available, however, in TUTOR, the author defines his pictures by drawing them at the graphics terminal. Then the picture definitions are saved on a file and displayed later (Pratt, 1974).

The author language, PLANIT, is one of the more recently developed CAI languages (Bannick, 1970). It was developed by Systems Development Corporation with the goal of creating a machine-independent CAI system. PLANIT has been implemented on several different machines. The student terminal normally used with PLANIT is the teletype, so the graphics are only tabular pictures. The pictures are drawn the same way as in Coursewriter. However, in Coursewriter, a picture which is 36 dots high by 16 dots wide occupies only a small portion of the CRT screen and the dots appear to be smooth lines. On the teletype in PLANIT, the dot matrix takes up a much larger section of the paper and the dots are quite distinct.

Development work in a more elaborate graphics capability is being done, where the student terminal consists of an electronic tablet and stylus (Williams, 1968). A novel feature of this developing system is complete graphics input. The student can answer the questions by writing (drawing) the solutions on the electronic tablet

CHAPTER IV

THE CAI/GRAPHICS SYSTEM

When considering the purchase or development of a CAI/graphics system, the educational institution must evaluate all aspects of the system before making a decision as to what type of system is really needed. A number of general questions must be asked and answered before considering the actual type of hardware and software to use in the system. The following questions should be considered:

- -What type of student will be taught?
- -What is the maximum number of students to be on-line at once?
- -What kind of courses will be offered?
- -What instructional strategies will be used?

A few general considerations in answering these questions are presented in the first section of this chapter.

What type of student? The age range and abilities or disabilities of the students who are to use the system must be catered to when selecting the hardware or when designing the software. System response time is more non-graphics CAI via the standard interface of a teletype. However, when CAI/graphics is considered, where the type of interface to the graphics terminal is not standard from vendor to vendor, the problem of tranferability of CAI/ graphics courseware is almost insurmountable unless two identical systems are involved. Until some sort of standards or guidelines are set up for courseware development, each ambitious CAI/graphics teacher has to write his own style of courseware.

What instructional strategies? Many course authors require extensive student records, so that the lesson can be dynamically tailored for the individual student. Some authors will keep a history of all student responses made during the current course. A complex file system may be needed to support the student files. A file system means more software to develop, and a larger CPU memory and a random access auxiliary storage device to purchase.

After the above questions are explicitly answered, the educational institution is ready to choose specific hardware and software to accomplish the desired goals. The next section of this chapter presents some guidelines in selecting the hardware and software.

the communications link does not have to be so fast and expensive because less information is transmitted per picture. A trade-off situation does exist here.

What kind of communication? The type of graphics terminal chosen and the physical lay-out of the system determine the type of communications. Some terminals must be very close to the main computer, while others have phone adapters and can be miles away. If phone communications are chosen, the cost of phone lines over long distances is not trivial.

What main computer? The central control computer should have a good interrupt system, flexible instruction set and reliable I/O system.

The interrupt hardware of the main computer is critical in time-sharing. An adequate priority system of interrupts is recommended, where the software can assign priorities to the interrupts and process them without too much overhead. External interrupts should be assigned specific locations in memory to which they interrupt since interrupt processing time is faster if this capability is available. If this process of interruption and servicing and returning to the interrupted program is fast enough to keep the graphics terminals and any other device running without any noticeable delay, then the interrupt system would prove to be adequate.

When analyzing the input/output system for speed to handle communications and processing, some basic parameters should be calculated (Tirrell, 1971). During one second of machine time, "X" number of machine cycles are available for either communications I/O or processing. The maximum number of machine cycles in one second required to handle communications can be calculated, which would yield some information as to how much time is left for processing per second. For example, if the speed of the communications line is 2400 bps and if the data character size and buffer width is 8 bits, then 300 transfers per second is the maximum necessary to drive one line, where each transfer takes one machine cycle. If 100 similar lines exist, then 30,000 machine cycles per second (maximum) are busy just with communications. An average computer with a machine cycle time of 2 microseconds has 500,000 machine cycles per second available. So in this case 470,000 cycles are left for processing, which includes execution of all courseware as well as servicing any interrupts. A mix of necessary "processing" instructions per second could also be determined. Then if the sum of the two requirements for machine cycles per second exceeds the number of available machine cycles per second, queuing of data must be considered or possibly an overloaded described in Chapter III that an adequate CAI/graphics language has not yet been written. According to Suppes (1970) the most satisfactory author language for non-graphics CAI has not even been developed. The additional requirement for an author language to support graphics input and output processing further emphasizes the lack of a satisfactory CAI/graphics author language, since the study of graphics input/output is also still in the developmental stages.

In this section of the paper, some of the necessary requirements in a CAI/graphics language are described and justified.

One of the basic features a CAI/graphics language should have is the ability to retrieve and add to a file of "picture" definitions, where these pictures can be displayed using different scale factors and at different locations on the screen. For example, if a flowcharting lesson were being presented, the decision box, when first introduced, could be the central figure on the screen with an explanation about it at the bottom of the screen. Later in the lesson, when the decision box is used in conjunction with other flowcharting symbols, it could be drawn smaller at the desired screen location. The decision box picture would simply be part of the picture could draw his desired file of pictures by using the light pen at the CRT, where the movement of the light pen would be followed by a tracking program, or by using an alternative device, such as a mouse or stylus (see Chapter III). In either case, the author could draw a picture, and then enough information to regenerate the picture would be saved on the file. The picture could be given a name, which could later be used when displaying it. After the author drew each picture, he could see it on the screen and change it if he were not pleased with it by interacting with a graphics editing program. If the author could not draw his picture file, he would have to generate it by writing a program, where each picture is described in terms of absolute vertices and lines, an extremely time-consuming task and quite prone to errors.

Some additional graphical output features of a CAI/graphics language that would be desirable are selective erasing of the screen, as well as blanking the screen completely, blinking certain pictures on the screen for emphasis, and a general PLOT package to dynamically plot various functions the author or student may want to see.

Graphical input, where the student draws a picture or perhaps points at something on the screen using a light pen or similar device, could be handled by the

V. CONCLUSIONS

CAI/graphics systems are still in their infancy. CAI systems and interactive graphics systems have existed independently for about 15 years. However, neither type of system has really been perfected, so when the two imperfect and undeveloped systems were merged into a CAI/graphics system, the resulting concept obviously needs a lot of research and development before a reasonable and inexpensive system can result.

This paper has explored some of the dimensions of CAI/graphics systems. Several existing CAI/graphics programs have been described in order to understand how far the CAI/graphics concept has been developed at different schools. The components in a CAI/graphics system have been discussed in detail so that some insight can be gained into how they all fit together. Finally, some guidelines have been given for developing a CAI/graphics system.

The future of CAI/graphics systems is clouded by the expense of the systems and by the traditional disagreements which educators have about CAI in general. However, as time passes and standards change, the author feels that

BIBLIOGRAPHY

Abboud, V. and V. Bunderson (1971) A Computer-Assisted Instruction Program in the Arabic Writing System. The University of Texas at Austin, Computer-Assisted Instruction Laboratory, Technical Report No. 4.

Atkinson, R. and H. Wilson (eds.) (1969) Computer-Assisted Instruction: a Book of Readings, Academic Press, New York.

Avner, R. and P. Tenczar (1970) The TUTOR Manual. Illinois University, Urbana, Illinois, Computer-Based Education Lab, R-CERL-X-4.

Bennick, F. and C. Frye (1970) PLANIT Author's Guide. System Development Corporation, Santa Monica, Calif.

Boehm, B., V. Lamb, R. Mobley, and J. Rieber (1969) POGO: Programmer-Oriented Graphics Operation. The RAND Corporation, Santa Monica, Calif., RM-5825-PR.

Bork, A. and R. Ballard (1972) Computer Graphics and Physics Teaching. California University at Irvine.

_____________ (1971a) Inexpensive Timeshared Graphics on the SIGMA 7. California University at Irvine.

_____________ (1971b) Teaching Conversations with the XDS SIGMA 7. California University at Irvine.

Bunderson, V. (1970) Justifying CAI in Mainline Instruction. Proc. of a Conference on Computers in the Undergraduate Curricula, University of Iowa, Iowa City, Iowa.

Canter, J. (1971) Blueprint for an Anatomical Surgical Computer. A Quarterly Report of SIGGRAPH-ACM Computer Graphics, Vol. 5, No. 2.

Chase, E. and D. Brick (1970) Interactive CRT Display Terminals. Modern Data, Vol. 3, No. 6.

Levien, R. (1972) The Emerging Technology, McGraw-Hill Book Company, New York.

Machover, C. (1972) Computer Graphics Terminals--a Backward Look. Proc. AFIPS 1972 SJCC, Vol. 40.

McWilliams, E. (1972) Large Scale CAI--The NSF Project. Proc. ACM 1972 Annual Conf., ACM, New York.

Newman, W. and R. Sproull (1973) Principles of Interactive Computer Graphics, McGraw Hill Book Company, New York.

Oliver, A. (1969) A Measurement of the Effectiveness of an Interactive Display System in Teaching Numerical Analysis, Ph.D. Dissertaion, University of North Carolina at Chapel Hill.

Papert, S. (1971a) A Computer Laboratory for Elementary Schools. Massachusetts Institute of Technology Artificial Intelligence Memo No. 246.

________ (1971b) Teaching Children Thinking. Massachusetts Institute of Technology Artificial Intelligence Memo No. 247.

Papert, S. and C. Solomon (1971c) Twenty Things to Do with a Computer. Massachusetts Institute of Technology Artificial Intelligence Memo No. 248.

Pratt, T. (1974) Conversation with C. Kostetsky, Austin, Texas.

Rapkin, M. and O. Abu-Gheida (1968) Stand-alone/Remote Graphics System. Proc. AFIPS 1968 FJCC, Vol. 33, Part 1.

Sleeman, D. and W. Phelps (Undated) A System for Graphical Communication in a CAI Context. The University of Leeds, United Kingdom.

Stifle, J. (1971a) The PLATO IV Architecture. Illinois University, Urbana, Illinois, Computer-Based Education Lab, CERL X-20.

________ (1971b) A Plasma Display Terminal. Illinois University, Urbana, Illinois, Computer-Based Education Lab, CERL X-15.