DOCUMENT RESUME

ED 125 662 IR 003 754

TITLE Hardware Developments; Microcomputers and Processors; Grade School/High School Instructional; and Computer-Aided Design. Papers Presented at the Association for Educational Data Systems Annual Convention (Phoenix, Arizona, May 3-7, 1976).

INSTITUTION Association for Educational Data Systems, Washington, D.C.

PUB DATE May 76

NOTE 51p.; For related documents, see IR 003 748-756; Some parts may be marginally legible due to print quality of original

AVAILABLE FROM Association for Educational Data Systems, 1201 Sixteenth Street, N.W., Washington, D.C. 20036 ($10.00 for entire proceedings)

EDRS PRICE MF-$0.83 HC-$3.50 Plus Postage.

DESCRIPTORS Computer Assisted Instruction; *Computer Graphics; Computer Oriented Programs; *Computers; Display Systems; Elementary Education; Instructional Innovation; *Instructional Media; *Instructional Technology; Man Machine Systems; Mathematics Instruction; Secondary Education; Teaching Techniques; Test Construction

IDENTIFIERS AEDS 76; *Association for Educational Data Systems; Interactive Computer Systems; Minicomputers

ABSTRACT

Compiled are ten papers describing computer hardware and computer use in elementary and secondary school instruction presented at the Association for Educational Data Systems (AEDS) 1976 convention. An oral/aural terminal is described followed by two papers about the use of minicomputers and microprocessors. Seven papers discuss various uses of the computer in elementary and high school instruction: a computer can be used to plot and display conic sections and environmental designs, to help teach reading skills, and to generate tests or homework exercises. One paper recommends the use of games in computerized drills, and another explains computerized demonstration of some mathematics principles. The importance of the school computer coordinator is outlined by the Minnesota Educational Computing Consortium. (CH)

ED125662

HARDWARE DEVELOPMENTS; MICROCOMPUTERS AND PROCESSORS; GRADE SCHOOL/HIGH SCHOOL INSTRUCTIONAL; and COMPUTER-AIDED DESIGN. Papers presented at the Association for Educational Data Systems Snnual Convention.

D. T. Clements
Robert Green
B. K. Funk
W. A. Stannard
James L. Altermatt
Muata Weusi-Puryear
Linda L. Johnson
Allan H. Paschke
Donald W. Moffat
Arthur Chapman
Phil Winninghoff

IR003754

ASSOCIATION FOR EDUCATIONAL DATA SYSTEMS

The 1976 AEDS International Convention

Proceedings

TODAY'S REVOLUTION:
COMPUTERS IN EDUCATION

May 3 - 7, 1976

Phoenix, Arizona

These Proceedings have been made possible by a grant from the IBM Corporation as a service to the educational community.

AN ORAL/AURAL TERMINAL FACILITY

D. T. CLEMENTS
COAST COMMUNITY COLLEGE DISTRICT, COSTA MESA, CA 92626

ABSTRACT: The Coast Community College district is currently in the developmental stage of configuring a telephone-to-computer "Oral/Aural" terminal facility which uses both Synthesized Voice Output and Voice Recognition technologies. The Voice Output side of the facility is now installed within the Administrative On-line System. It interfaces with touch-tone input to provide query capabilities for Counselors and other district officials. The plan, of course, is to eliminate the touch-tone constraint and allow the users to vocally specify their commands. This paper deals with the district's experience to date, as well as the current developmental concepts.

1. INTRODUCTION

As we trace through the evolution of data processing, it is interesting to consider the growth of man-machine communications, for it is a mutation in which man has been ever dominant. Instead of yielding to the machine's native tongue of endless ones and zeros, we have cultivated numerous translators, interpreters and decoding devices to bring the machine closer to our dialogues, thought processes and perception. The eventual ultimate in communication is unknown. However, a facility which allows man the comfort of "conversing" with the machine via his oral and aural (hearing) senses is now close to a reality.

Audio response or "talking" systems are not new. Voice Output became a practical reality in the mid 1960s and enjoyed a mild spurt of popularity. These early implementations (which for the sake of later delineation might be termed Traditional Audio Response) were composed of sophisticated storage systems which were "loaded" with pre-recorded syllables, words, phrases or perhaps full paragraphs. Each "sound unit" was addressable. A host program then would issue a request for a particular unit and it would be output or more properly, "played". In general, it was an analog procedure.

A more recent development is the Voice Synthesizer. This equipment differs from the Traditional Audio Response systems in that there are no pre-stored, pre-recorded sounds. It reacts only to digital excitement. The host program determines the phonetic structure of its output by means of a vocabulary lookup procedure or pronunciation algorithm and sends a binary string of appropriate pronunciation codes to the device for articulation. Figure one is presented to illustrate the difference between the two Voice Output techniques.

Voice Synthesis provides many advantages over the mentioned traditional systems, the most notable of which is that there is no limit on vocabulary size. Word/pronunciation tables can be easily stored on disk. The structure used at the Coast Community College district uses an ENVIRON/1 Byte-string file and accommodates the area of 8000-10,000 words with their phonetic codings per cylinder of 3330 disk. Another attractive feature is that an interactive procedure may be quickly developed to add or refine vocabulary as it is being accessed by other tasks.

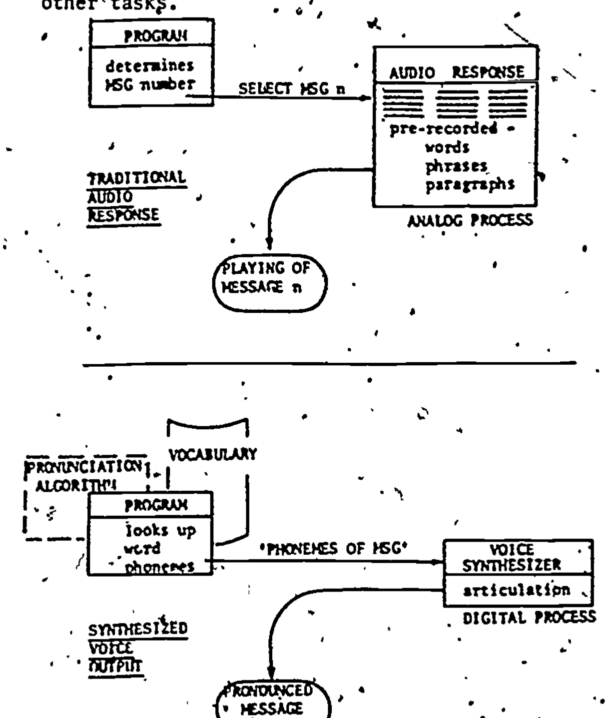


Figure 1: Audio Response vs Voice Synthesis

The ability to dynamically create sentences within a host program must be considered a valuable asset. That is, constant text may be spliced with variable data to create a meaningful output string.

Perhaps the most convincing argument in favor of Voice Synthesis is its economy. The district currently uses a synthesizer called VOTRAX. It is a product of the Vocal Division of the Federal Screw Works and may be purchased for $3500-$4000. It is well under the cost of the other Traditional Audio Response units comparable to IBM's 7770 which starts at a $60,000 figure and can go as high as $230,000.

The history of Voice Recognition is mainly all research and experimental systems. The objective seems to be a truly cognitive facility capable of distinguishing all words, their meaning and the context in which they have been used - to respond to unconstrained speech in a natural language. The research continues today. Advances in the art of pattern recognition and natural language processing have undoubtedly been helpful, but there are still many obstacles which need to be overcome. However, there are a few commercially available systems which are recent "spin-off" results from this research and within certain applications they are an attractive input device. In the early months of 1976, systems capable of recognizing anywhere from 16 to 144 words are in commercial service from a handful of vendors. (IBM, by the way, is rushing to join this market. They have had an experimental program since 1972. Supposedly, the goal of their effort is a voice-driven typewriter system capable of delivering rough, first-draft-type documents.)

Aside from the obvious communication advantages, computer voice-input offers some other assets which might not be thought of immediately. One of these is that as a clerk is busy inputting vocally, her hands are free to work with source material - whereas in the traditional on-line data entry situation her hands were needed by the teletypewriter. Consider also, that Voice entry minimizes personnel training and may eliminate the requirement for the clerk to have typing skills.

The district has recently procurred a VE 200 Voice Recognition System from the Perception Technology Corporation. In its current state, it is capable of distinguishing a vocabulary of 16 words. The standard vocabulary includes the digits zero thru nine and then six specified control words such as "enter", "cancel", "repeat", etc. The system is not limited to these words, only to the overall vocabulary size of 16 words/utterances. Larger vocabularies may be obtained my increasing memory size. It is capable of handling any defined user's speech pattern peculiarities with accuracy above 95%.

The rest of this paper shall concentrate on the operational characteristics of the speech processors mentioned previously, the resources which they require, and a method of coupling the two to provide an Oral/Aural terminal facility.

2. THE VOICE SYNTHESIZER

Phonetic coding for the VOTRAX Voice Synthesizer is shown in Figure 2. The device reacts directly to a coded string of phonemes. (A phoneme is the smallest unit of speech.) There are also pausing and inflection codes which may be included in the string for clarity. Note that the phonemes and pauses only use the low order six positions of an eight bit configuration. Inflection codes utilize the high order two bit structure and are properly OR'd to a phoneme for a desired sound. The example shown for pronouncing the word HELP indicates the necessary inflection/phoneme notation and the resulting binary codes. The H sound, of course, must be stressed; thus a level 3, high inflection. Note also that two phonemes are required to accomplish the diphthong vowel sound.

▶ THERE ARE PHONEMES,

PHONEME NAME	BINARY REPRESENTATION
AW (as in law)	00111101
AH (as in car)	00100100
AY (as in eight)	00100001
AE (as in cat)	00101110
B (as in book)	00001110
etc....	....
etc....	....

▶ PAUSE CODES,

PAUSE NAME	BINARY REPRESENTATION
PAØ (short pause)	00000011
PA1 (med. pause)	00111110
PA2 (long pause)	00110000

▶ AND INFLECTION CODES

INFLECTION	BINARY REPRESENTATION
IN1 (low)	10000000
IN2 (normal)	11000000
IN3 (high)	00000000
IN4 (higher)	01000000

EXAMPLE: The word HELP is constructed as follows:

$$\frac{\text{inflection}}{\text{phoneme}} \quad \frac{3}{H} \quad \frac{1}{EH1} \quad \frac{1}{UH3} \quad \frac{1}{L} \quad \frac{1}{P}$$

= binary 00011011 10000001 10100011 1001100 10100101

Figure 2: Phonetic coding and its binary representation

The device obviously belongs in a real-time environment, and there are many ways it can be configured. The district has initial voice-output configuration as shown in Figure 3. The synthesizer interfaces to a Bell 403 data set and subsequent acoustical touch-tone interaction.

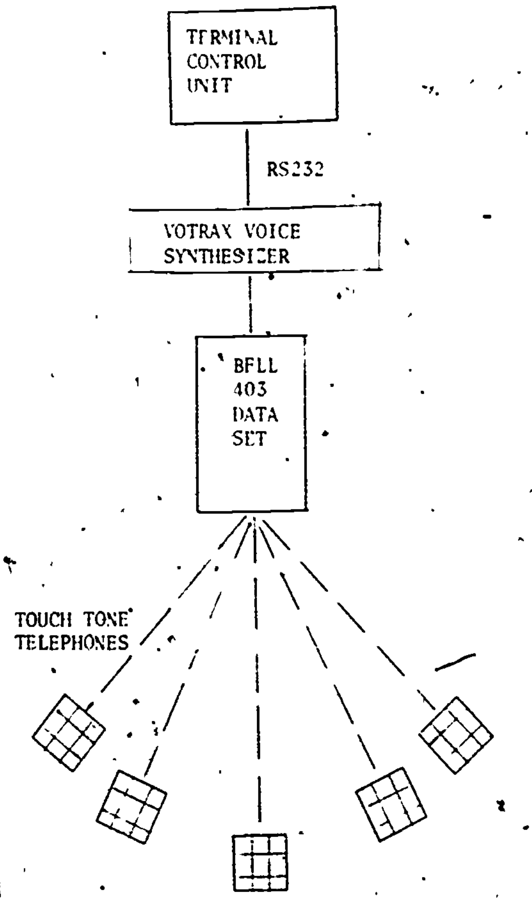


Figure 3: Voice Synthesizer and touch-tone interface.

Figure 4 expands the view of the voice output to include the host computer and the auxiliary storage needed to store vocabulary. ENVIRON/1 is a Data Communications and task management software facility obtained from CINCOM Systems, Inc. The district uses this package for its on-line administrative applications. The product also contains a superior disk accessing method called Byte-string files which has been used effectively to store vocabulary. The vocabulary table is structured with a prime hash algorithm. It has been given the acronymic name PHAST (Phonetic Hash Table) and has proven to be an expedient organization for the random access imposed by the Lookup procedure. The table is maintained at a load factor below 70% to enhance search performance. The average search is accommodated by 1.8 probes. And, because of the virtual nature of ENVIRON/1, searches for common words are often resolved in buffer store.

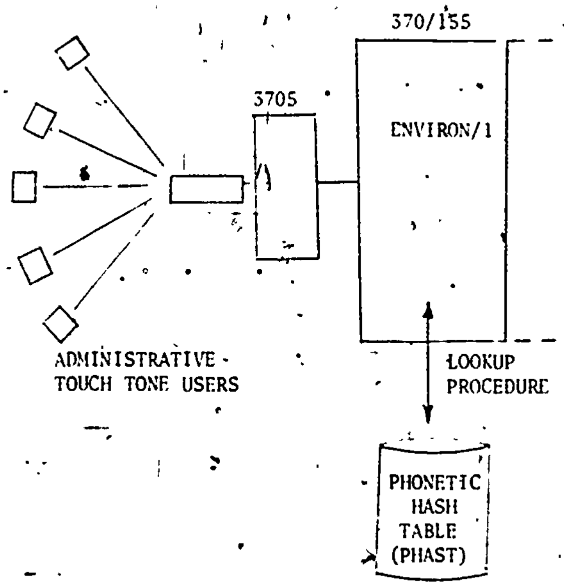


Figure 4: Host System with Voice Output Configuration.

A rather traditional Lookup procedure is used. The application program invokes this procedure with a text/sentence character string parameter. Scanning determines the text elements. If a numeric is detected, it is tested to see if it is evenly divisable by 10 (modulo-10). If this is the case, the leading digit's phonemes are concatenated with the phonemes of its place indication. For instance, the numeric text 700 would be pronounced "seven" "hundred". If an application program desires place indication verbage in its output, it purposely uses this logic. For example, if the value 1,234 is sent to the Lookup procedure as 1000 200 30 4, it would be pronounced as "one thousand" "two hundred" "thirty" and then "four". Numeric text which is not modulo-10 is pronounced digit by digit.

When a "word" or more properly a string of contiguous characters bound by separators, is detected, it is used as a search argument into the PHAST vocabulary. If the search is successful, the associated function of phonemes is output. Otherwise, the word is pronounced, or spelled, character by character.

Separators are used to issue natural pausing. The presence of a period effects a long pause, a comma obtains a medium pause, and all others such as hyphens, colon, blank, etc. result in a short pause.

Mention might be made at this point about an automatic pronunciation algorithm. Bell Laboritories has published accounts of a program which produces Synthetic English Speech by Rule. Although the speech produced is not inflected, it is intelligible on at least 97% of running text. It contains some 750 pronunciation rules and requires a lexicon structure for words which are determined to be exceptions to the rule. This is a promising effort.

3. THE VOICE RECOGNITION SYSTEM

The VE 200 Voice Recognition System is composed of three major units; a small "ear box" containing the speech processing circuits, a Digital Equipment PDP-8/E mini-computer, and an interface between the two. The system processes a word within 160 milli-seconds after vocal input of the word has ended. Its maximum rate is 120 words per minute, allowing at least 1/5 of a second between words. It is able to accept such input via microphone or telephone with equal accuracy. The output is a four bit TTL compatible register which presents the recognized word as a binary coded number from Ø to 17_8.

Word processing is accomplished by determining a spectral distribution of the speech signal and then passing it through a bank of six bandpass filters. The filtered outputs are rectified, smoothed, sampled every 10 milli-seconds and stored into the PDP8. The mini-computer then calculates linear combinations of the six channels and tabulates them to form the x-y data points of a two dimensional space.

Operationally, the system can be divided into two general modes; Training and Recognition. It must, of course, be informed (or trained) of the 16 application words and each particular user's unique pronunciations before it can ever be expected to recognize them.

Each user must go through a Training session with the equipment. Users having multiple applications need to undergo training for each vocabulary. Training sessions are quite straight forward - the system is placed into Training mode and then prompts the user to pronounce the 16 words of his application vocabulary, in sequence, one at a time. This process is then repeated two more times so that the system has had an opportunity to "hear" the pronunciation of each word three times. The purpose for the repetition is to recognize possible variances in the pronunciation. The final tabulation is called a User Voice Template.

When the system is placed in recognition mode, actual word identification is made by the computer comparing the pattern of the received speech to the patterns within the User Voice Template currently resident in its memory. The closest association is chosen, and its word sequence number (Ø-17_8) is output.

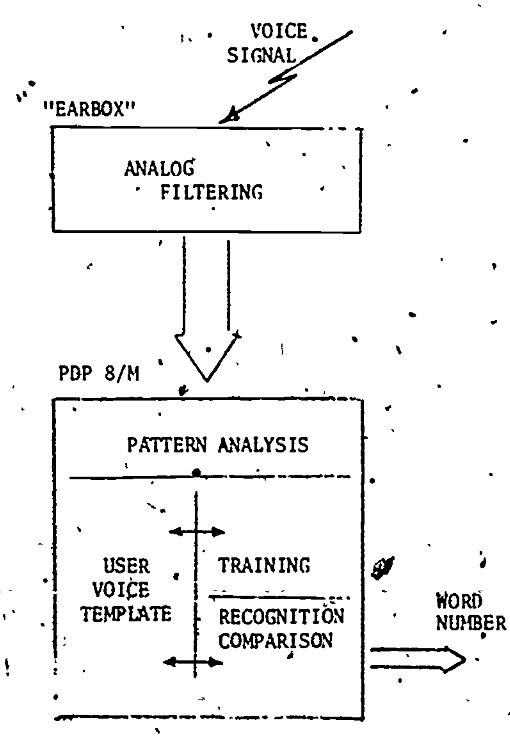


Figure 6: Voice Recognition System Outline

4. THE ORAL/AURAL FACILITY

Figure 7 is presented to show the district's configuration which has been labeled the "Oral/Aural" terminal; a combination of the two vocal processors mentioned previously. The user's single device is his telephone - and this is only for convenience, as he could also come to the computer facility and talk into the system with a hardwired microphone.

There are four different procedural/data paths within the configuration. The first one indicated is the Voice Output medium ("a") described earlier. The next path "b" TOUCH TONE INPUT is an option for those users with a touch tone telephone. Should this option be selected, only the voice response side of the configuration is used. That is - the tone input; the application program process; the tone input; the application program process; the word lookup procedure; and then the "a" path of digitized phonemes back out (ref. Figures 3 and 4).

The voice-in/voice-out option utilizes paths "a", "c" and "d". Once the user is identified, his template must initially be loaded into the PDP8 via path "c". This is accomplished by realizing the user's sign-on number (described later), accessing his template stored on disk, and then outputting it to the minicomputer's reception. This process, of course, enables the Voice Recognition path "d". Words may then be spoken, analyzed and numerically incoded ($0-17_8$) on to the host system. Voice response to the input is accomplished through the aforementioned lookup procedure and output path "a".

A note should be made that the "c" path is also used for device control from the host ENVIRON/1 system. The "d" path is also used as a result of Training sessions to store user voice template on disk for subsequent access.

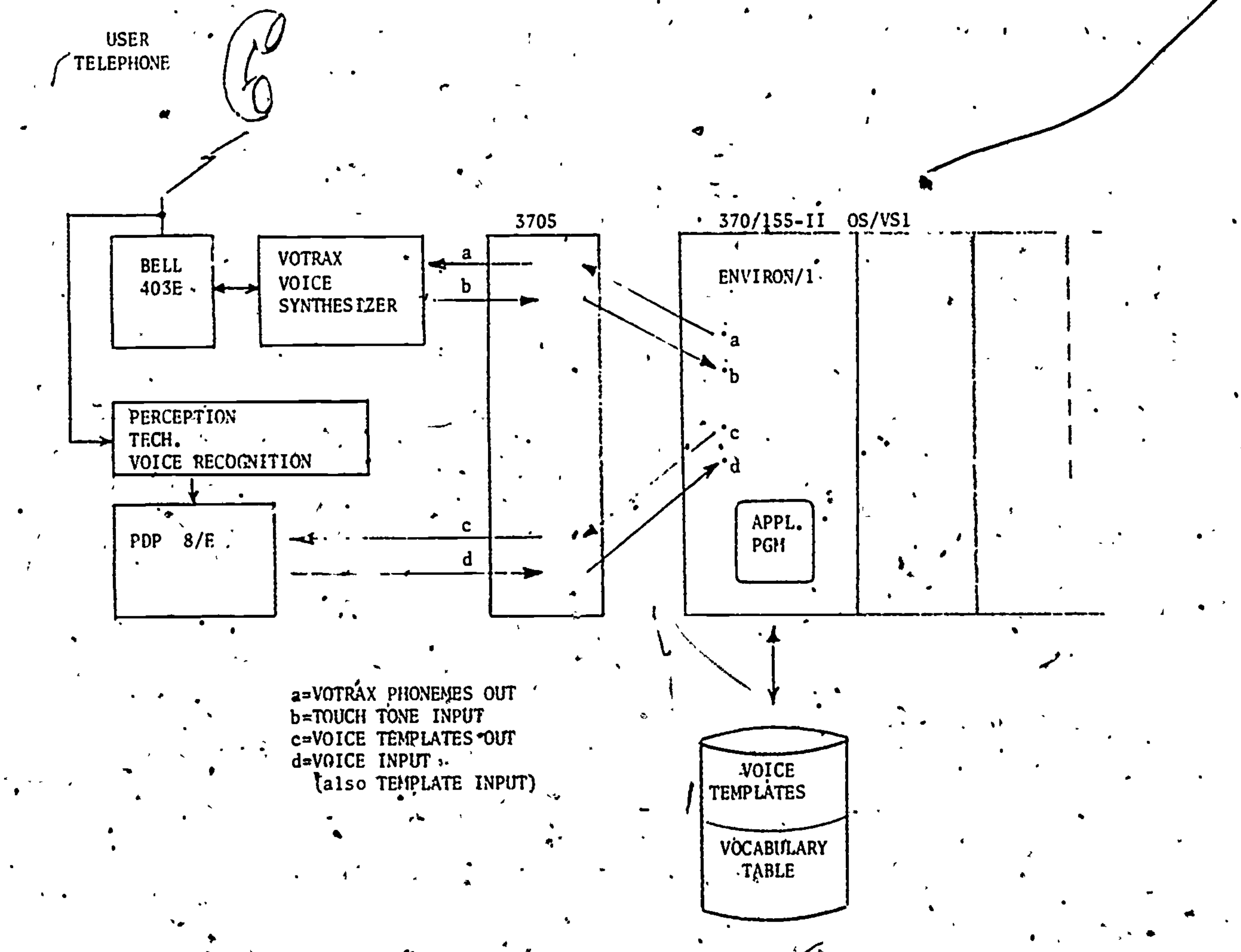


Figure 7: Configuration of the Oral/Aural Terminal Facility

Unique procedures have been created (by the vendor) to initially identify a User and his sign-on account number. A special "Joe Average" template is established. This template contains only two words. The words are digit names which have relatively opposite phonetic structure; for example, "two" and "five". The template is developed during a special training session in which as many users as can be accumulated are requested to speak (input) their pronunciations of the digit sounds.

User sign-on account numbers are constructed solely with the two selected digits and have a determined fixed length. Recalling the earlier example and assuming a sample length of seven (7), some typical account numbers might be: 2552552, 5525552, 2222225, etc. The number of possible account numbers within this scheme would be 2^n where n is the fixed length.

After system initialization and also after every user sign-off, the "Joe Average" template is stored into the minicomputer to wait for a subsequent sign-on. The reliability of this template is surprisingly good. However, should an erroneous number be received by the host system which maintains the user registry, the user is prompted by the voice output to speak more distinctly and try the sign-on again. Should the second try fail again, the occurrence is output to the Computer Operator's console log and instructions are sent to the Oral/Aural configuration to hang-up the line.

5. SUMMARY

What has been presented in a model: At this writing in early 1976, the configuration does not physically exist. The voice-output side is operational in a student information inquiry application. The voice recognition system is operational in a stand alone mode only and work still remains to be done on the host system handling procedures. Unfortunately, the configuration's development is often set aside because of other priorities. It is hoped, however, that a performing configuration will be demonstratable by the end of 1976.

PUBLISHED REFERENCES:

- All About Voice Response, Datapro 70, September 1974.
- Voice Recognition/Response From Perception Technology, Modern Data, November 1974.
- Voice Output For Student Information Inquiry, CAUSE Proceedings Volume II, 1974.
- Voice Input Systems for Computers For Computers Go On.Line, ADP Newsletter, Volume XX, Number 3, February 1976.
- Synthetic Speech by Rule, Bell Telephone Labs, March 1974.

VENDOR/EQUIPMENT REFERENCES

- VOTRAX Electronic Voice Systems
  Vocal Interface Division
  Federal Screw Works, Troy, Michigan 48084
- VE 200 Voice Entry System
  Perception Technology Corporation
  Winchester, Massachusetts 01890
- ENVIRON/1
  CINCOM Systems, Inc.
  Cincinnati, Ohio 45211

MICROCOMPUTERS IN THE EDUCATIONAL ENVIRONMENT:
AN APPLICATION IN FINE ARTS

ROBERT D. GREEN
CONSULTANT
2971 MAGLIOCCO # 10
SAN JOSE, CA 95128

ABSTRACT: The microprocessor- decendant of inexpensive calculator technology, is finding its way into a myriad of applications in our lives, from electronic games to grocery scales. Using it along with familiar types of support circuitry, inexpensive and yet relatively sophisticated computers: microcomputers result. Some of the history and impact of these microprocessors and microcomputers in the field of education is discussed along with a practical application. The application is in the area of textile and fabric arts, namely a weaving simulator.

Pick up virtually any computer or electronics publication and you will find a barrage of articles, columns and advertizements regarding the microcomputer. Many of you may be intimately familiar with the microcomputer and microprocessor but let me review the terms and describe the history of the beasties. That way those already familiar can understand me better and those who aren't can catch up.

As in much of what we do, there is an appreciable fuzzy area in the terminology surrounding the microprocessor. Therefore, let me define what I mean. I use the term microprocessor to describe a family of electronic components that provide the basic computer-like digital operations. That is performing an operation or series of operations in response to a coded instruction. I am certain that there are much more elegant definitions but I am purposely being vague so as to include a wide variety of components available. They range from little more than a four bit slice of an arithmetic-logic-unit (ALU) to a virtually complete computer device contained on a silicon chip roughly five mm. on a side. None of these devices can operate completely alone, all requiring some form of input and output devices. When combined with these devices, I refer to the result as a microcomputer.

Each of us probably has a different mental concept of a microcomputer bases on our preconceived ideas about computers. Most modern computing devices have four recognizable features that are found in the microcomputer:

1. An arithmetic-logic-unit (ALU) that actually performs the required operations on appropriate operands.
2. A stored program which executes the operation instructions.
3. A readable and writable storage area.
4. Some sort of input and output to the "outside world".

Let us look at the cheapest (that I know of) sophisticated microcomputer available to identify these features. Typical total price? About 7.95 dollars at any good discount store. This marvel of electronics is the pocket calculator. Looking at the topology of integrated circuit within the unit you almost always can identiy the logic area (ALU), the program area (Read-Only Memory), the storage area (RAM), and the I/O circuits. The integrated circuit has been purposely designed with these features in mind so that the manufactureer will have the lowest total costs as I will explain later. THINK ABOUT THIS: the calculator as a microcomputer complete with input and output devices represents probably the largest volume growth of computers in education that we will see for some time. I'll talk about other applications shortly.

It appears to me that there is primarily only one reason we will see the increasing use of the microcomputer in education: cost. Sure new cheap computers will allow all kinds of computer training to occur at all levels but it is still practical only because of low cost. In addition, the motivation behind the recent rapid development of the microprocessor has been the same desire to lower costs. In the late sixties (1968 and on) the semiconductor industry was perfecting the techniques of large scale integration (LSI) using metal oxide semiconductor technology (MOS). This allowed the implementation of many electronic functions on a single device, drastically reducing the number of components required. Promised reduced system costs brought scores of customers to the suppliers, each with his unique set of requirements. Major problems incurred at that time were typically minor logic errors or human errors in translating logic into circuitry. The net result was, however, substantial debugging and redesign costs along with associated delays. It became evident to the semiconductor manufacturer rather quickly that two things were happening:

1. He had many custormers whose applications were very similar, differing only in details. This was particularly true of calculator manufacturers.

2. He was having to maintain a large amount of his resources devoted to detecting minor errors which often did not relate directly to some custormer requirement. In other words he was often "reinventing the wheel."

Apparently the ideas of using a computer-like approach to the application of calculator circuits were develeoped by several companies about the same time. However it is apparent that Intel Corporation was the first to publically acknowledge the fact that the resulting circuit was useful in many types of applications besides a calculator and advertized as such. The circuit was announced not as a calculator circuit set but as a microprocessor.

What advantages resulted from this development? Well first the MOS company could have a family of circuits that did not have to be completely redesigned for each custorme. The ALU part was the same for all and the customer could fit this processor to his application. Many times the only changes required were on an external program storage ROM. Other more subtle cost savings occurred as well. For instance the task of mask generation and checkout was reduced, test generation was simplified, cost reductions that result from large production runs of a circuit resulted, documentation was simplified and on-and on.

Perhaps one of the finest advantages was that since the actual operation control was stored in a ROM, the whole system could be simulated completely using all of the components except the ROM. A ROM emulator which is basically a read-write memory was hooked up in place of the ROM and any program faults could be changed before building the actual ROM. Early attempts at simulation of custormer circuits before the microprocessor were often only approximate and sometimes took longer to check out than to build the actual device.

Disadvantages? Of course! Many potential users became dismayed because of low circuit speeds, primative instruction sets, word size and their unfamiliarity with such a device. However, often a slightly different approach to their problem allowed economical use of the microprocessor. Input and output to the early microprocessors was not always easy so that there was a large amount of support circuitry required. And yet the revolution and a revolution it is - was begun.

AND NOW...

Newer microprocessor designs with bit lengths to 16 bits and speeds ranging ever higher are available now. Popular eight bit processors in n-MOS technology have shortest instruction times of less than two microseconds. Some devices require only one power supply and most have a wide variety of support circuitry. For instance, 16 to 24 lines of programmable input and output are available in one package with provision for handshaking. Or another: a single device which handles all the requirements for interfacing with a remote terminal, including parity bits. Direct memory access controllers, reprogrammable ROMS, programmable timers, a mass of new "standard" circuits are appearing daily.

There are microprocessors that simulate the instruction set of computers like the PDP 8. There are microprocessors which have all different kinds of addressing modes: immediate, direct, extended, relative, indexed. Some have internal stacks, multiple accumulators, you name it. Some allow periphers to be treated as a memory address. The features are endless. It is, in fact, truely remarkable that the devices presently available are so powerful an we may be sure that even better devices are on their way. Because so much information is readily available, I will not go into specifics about these devices in this paper.

Ok, so the circuits are there, are they cheap? Almost!! In single quantites manufacturer reps are selling some microprocessors for $30 or less. They are selling kits which contain chipsets (usually enough to get them to do something at around $100 and up. But for the person who wants his computer relatively complete, the price tag starts about color TV rnage ($250 up). "Well," you say "thats pretty cheap" and it is. A cleaver person can accumulate enough other electronics to interface this computer to keyboards, TVs, typewriters and the like. If he has a teletype or a large checking account to buy one (2 to 5 times the cost of the computer kit!) he can hook right on. An entrapeneur's playground, there are already literally hundreds of companies supplying hardware, software, peripherals, kits, lessons, lectures and whatever else you will buy. There is a computer store next to a plant shop not far from me!

There are cheap computers available and this in itself is a revolution. The possibilities for the teaching of computer operation, of loical thought, and of games are staggering. But I believe there is more in the wings. Already the personal calculator has invaded the campus and school room...whats a slide rule? This is a number crunching educational tool. I am for a little creative thought to get the microcomputer out of the computer-think applications and ito more tools. People over the country are starting to do this in their hobby rooms and clubs in some areas of art: music, poetry and pattern generation. With minicomputer like power available, its coming. How about some other applications in education. How about tecahing machines, electronic tutors, ... What jobs cost money that would cost less done electronically? What tasks could be done both better and cheaper? I believe the microprocessor and microcomputer will provide many such inexpensive solutions.

I humbly present a for instance:

In the teaching of handweaving on a loom, one of the most difficult things to learn is how to generate patterns. As is explained below, only certain combinations of treads may be manipulated at any time so the design is said to be loom constrained. On even simple looms, however, the number of combinations of weaving patterns is virtually endless. There is a method of transforming a desired pattern into a weaving procedure called pattern drafting. It is important that the student learn this skill so that full use of the loom can be achieved. However, until considerable expertise is achieved, the only good way to check a pattern draft is to weave it. The student must go through the long and error prone tasks of threading the loom, tying up the controls and weaving. If only then an error is found it is more or less back to ground zero. In an educational environment, this means delays up to a weak and unnecessay use of the looms. Since looms are relatively expensive, the teacher is prone to ignor this important part of weaving or to limit the class size or to spend more capital.

I have attempted to develop a tool which will alleviate the above problem. With it the student or artist can simulate his threading, tieup and weaving sequence in a matter of minutes using a keyboard as an input and a standard television receiver as an output. The goal of the system was a "bare-bones" approach to the hardware.

To understand what the simulator must do, let us review the basic concepts of weaving. A weaving consists of a lengthwise set of threads (the warp) and a crosswise fill of threads (the weft). The weft is held in place and the patterns developed by passing individual warp threads either over or under the weft. This over/under result is accomplished by raising selected warp threads before the shuttle carrying the weft is thrown. The lifting of each warp thread is controlled by a small movable piece called a heddle. The warp is threaded through the heddle. In common hand looms used in classes the heddle may be connected to an actuating piece called a harness. There are usually 2,4 or 8 harnesses on a loom. Typical warp counts of up to 15 per inch are common. On a N harness loom, from one to N-1 harnesses may be lifted at each shuttle throw. To further aid the weaver, a series of footpedals is provided so that one ore more harnesses can be lifted with one action (Also so you don't need N-1 hands or feet). These pedals are called treadles.

The weaver must determine which warp threads must be lifted at each pass of the shuttle. If a pattern can be done on an N-harness loom, there must be N or less combinations of harness lifting required (ignoring tapestry techniques). With complex patterns it is easy to see how hard this is to determine. IBM has a large program devoted to this task.

Weaving as described above can be represented as a coordinate grid where the passing of the warp over the weft could be represented as a zero and the opposite case as a one. This grid or matrix of ones and zeros completely describes the pattern if given the parameters of density, color and thread size of the warp and weft. Computer oriented? of course and you may know that some computer concepts developed from the textile industry.

The microprocessor used (note microprocessor not microcomputer) loads a matrix in memory with the appropriate pattern as the weaving progresses. A keyboard is scanned during the vertical retrace time of the television set. If action is required, the processor attends the task via stored programs in ROM. Because some tasks last longer than the retrace time, the interrupt feature is used to store the program state in stack until the next one. There is no noticable lag however.

A typical session with the simulator goes like this:

1. Clear the machine (automatic if turn on).
2. Set up the warp shade, size and spacing.
3. Set up warp/harness pattern.
4. Set up the harness/treaddle pattern.
5. Set up the weft shade and size.
6. Lift a set of harnesses or treaddle.
7. Weave.

Steps 6 and 7 are then repeated, changing the weft specification when desired. Should the pattern not be suitable or have a mistake there are edditing sequences to make changes quite painless. The student has an opportunity to experiment and make errors without the frustration of doing it on the loom.

The microprocessor must keep track of the warp/harness threading, ORing together all of the warp threads which must be lifted and applying the pattern to the matrix. Also the current warp line must be stored to control the display, appropriate changes be made on demand as well as other book keeping tasks. A feature also allows the display of the back of the "fabric". Details of the kinds of instructions executed will be more fully described in the presentation.

One limitation of the MOS microprocessors available is that they cannot readily directly control the video signal. This is because there are too many internal and external instruction fetches associated with moving data for the 50-some microseconds of horizontal sweep. DMA is used to output eight 8 bit words for each display line. One word is used to indicate the harness combination, one for the warp information and control and the other six for the pattern to be displayed. MSI counters and registers are used to control the DMA and video signal generation. An LSI circuit is used to provide the horizontal and vertical synchronization.

WHERE NOW? Much can be done to extent this device (and increase its cost). For instance color may be added with little more complexity and a teletype output for hard copy of the pattern would be nice. There are large programs of interactive textile design from IBM and recently an interresting weaving program has been written by K. Huff at the University of Kansas. They both require large general purpose computers at this time. An advantage of my device is its size, cost and closeness to the actual weaving process. But I am most interested in expanding the microprocessor into design areas with capabilities similar to the EXPLOR and MINIEXPLOR programs of Ken Knolton (Bell Labs). The modular design of the weaving similátor will allow exploration (pardon the pun) of this area of the arts.

I believe the time is ripe for the creative minds in the electronics, computer and educational fields to apply this powerful component in providing new educational tools. I hope it is a challenge to you- 'it is to me.

I would like to aknowledge the aid and donation of microprocessor components by American Microsystems Incorporated, Santa Clara, CA. Their S6800 microprocessor forms the basis of the equipment described.

MICROPROCESSOR BASED EDUCATIONAL TERMINAL

B.K. Funk
UNIVAC Division of Sperry Rand

ABSTRACT: The advent of the inexpensive microprocessor can lead to the development of a terminal designed for rather than adapted to the educational field. A resident microprocessor can allow easy expansion from a basic display terminal to one that included present input/output devices such as random access visuals, touch panels, audio output, and graphics, while allowing for inclusion of systems to be developed in the future, such as video disc.

I. INTRODUCTION

I present this paper with the basic premise that all present would greatly benefit from the development of what I will call a complete educational terminal. To paraphrase a statement made in one of the excellent studies by the University of Texas, Project C-BE, "teachers are too often loaded down with the jobs of assigning, grading, and giving students feedback on homework and tests when a good combination of a computer system and as educational terminal can perform these tasks as well as or better than the instructor. Removing the mundane chores from the teachers role frees the teacher for work in the activities where they perform best: providing insight into difficult concepts, transmitting an understanding of abstract ideas, and just plain firing up or inspiring students." (1)

I propose to spend the next few minutes talking about three related subjects. First I will describe what will be, for the purpose of this talk, a "complete" computer terminal developed for the educational field. Due to the rapid pace of technology I may inadvertantly leave out some new device or another, but the system will be complete enough to allow the necessary points to be made. Secondly, I will try to show some of the problems encountered during the development and manufacture of as complex a system as this. As the prospective customer, you would normally not be involved in these types of problems, but perhaps this will help you to understand why there are no widely accepted educational terminals available presently. I am not forgetting that there are several vendors that have successful installations using general purpose computer terminals, such as the Chicago Board of Education's installation of Univac U-100 Display Terminals. I do feel that until an inexpensive terminal is developed with the education field in mind, the potential benefit from CAI will not be realized. Finally, I will try to show how the use of one of the newly developed microprocessors and it's associated circuitry can be used to solve the problems alluded to above and lead to development of the desired educational terminal.

II. EDUCATIONAL COMPUTER TERMINAL

The components in the basic configuration of our complete educational terminal should be a typewriter style keyboard and television style display device. This terminal could look like any number of plain vanilla CRT terminals now available. I propose a television style CRT device rather than the newer Plasma Display, because I feel that until there are some significant breakthroughs in Plasma Display Technology, the price of the Plasma Display will limit its use in the school room. This decision must be re-evaluated often but is valid now. In order to gain entry into most school systems, this terminal will have to be priced in the $2000 or less range. The terminal should be able to operate under control of a remote host computer, which may already be in the system handling payroll and other standard computer type jobs. It should also be able to operate under the control of a minicontroller in the systems where budgets prohibit large mainframe installations. The terminals communication protocol should be easy to change, to allow operation on whatever communications system the customer has in use. Finally, after all of these simple requirements are met, the terminal must have the capability to be enhanced with all the special peripherals that make the educational terminal unique. Any or all of the peripherals should be options, restricted only by the requirements of the school system.

Educational Peripherials
Input Devices

Most of the input devices defined here have the advantage that they are not only effective in saving time and simplifying a task, but they are also fun to operate. The first, a touch panel, is a device that is placed in front of a display surface and, while not obstructing the vision of the person looking at the display, senses the position of a finger or other pointing device as the operator touches the area on the screen corresponding to the desired input. There are several technologies that can be used to develop a touch panel, most of them involving placing light sources on a couple of sides of the display and detecting shadows on the other sides caused by reaching through the light beams to touch the display surface.

Another input device that is used to replace or supplement the terminal keyboard is the two axis control stick, or as it is more commonly known, the joystick. The joystick, identical to the ones on so many of the computer games, replaces the cursor positioning keys found on most display terminals. As the joystick is moved forward or backward or left or right, or combinations of these, the cursor (position indicator) on the CRT moves in a corresponding direction. The joystick can be moved to place the cursor in the box or area on the screen associated with the right answer or another desired input. A separate button is then pushed to indicate that an answer has been selected and the student is ready for the computer to check his or her choice.

Other exotic input devices that may find use in schools are light pens and graphic tablets. Light pens are used for the same purposes as the touch panel, but instead of using your finger to touch the display, a special wand must be used. An application and one of the methods of operation of the graphics tablet will be covered later in this paper.

As needs change the requirements on the keyboard may vary. Kodak, for their Kodak Interactive Learning Terminal, developed an overlay for their standard keyboard that could be used by very young children.[2] The complicated typewriter style keyboard was covered by the overlay and only had 5 or 6 large round keys, each about an inch in diameter, for the student to concern himself with. The student then only had to relate the correct answer to the number on the key or the color of the key. Keyboards on terminals that will be used for foreign languages, mathematics, and some advanced science classes, among others, must have keys on which the symbolic representation can be easily changed. There must also be a means whereby instructors can develop special symbols or sets of symbols and represent them on the keyboard with relative ease.

Educational Peripherals - Output Devices

The output devices required on the educational terminal are perhaps not as fun as the input devices,but they are certainly as unique. I know of no other marketplace, other than education, for a terminal that talks back and shows still or moving pictures. All of these are necessary. According to a presentation by D. Monson at the IEEE 1975 Region Six Conference, "the ability to remember instructions 72 hours after it is given is higher if the information is heard and seen rather than either heard or seen alone. Recall after 3 days in this particular study was about 10% is the material was heard, about 20% if seen, and 65% if both seen and heard."[3] This is reasonable evidence that educational terminals must have both visual and audio output.

After listening to several types of computer stored and computer generated audio output devices it is the author's opinion that only recorded voice is good enough to use for an educational terminal. College and secondary school students may be able to adapt to computer generated audio, but grade school children (especially in the first few grades) might be adversly effected. In these early years we should be careful to not confront them with a learning tool that cannot by the nature of the device follow all of the rules of diction. Once again, a device developed by Dr. Goddard at Kodak for their experimental terminal has a desireable set of parameters. The Kodak audio output device, under computer control, can generate any of over 100 audio messages, each up to 9 seconds in length. The time required to access each audio message is quick enough (under 1/4 second) that statements longer than 9 seconds can be developed by butting messages together. This still allows for very natural sound. The combination of a device with high speed random access and the tonal quality of recorded voice would seem to be necessary in an educational terminal being used by students of all ages.

There are many ways to enhance the video images normally seen on a CRT display. The most common of these is to offer some type of graphic capability. Again, there are several ways to accomplish this depending on the application. For primary and secondary schools the ability to reproduce pictures of limited complexity may suffice. Upper grades in high school and many college courses may require rather detailed graphical representations including color or at least gray scale graphics. Architecture and perhaps other sources require detailed graphics plus the ability to move and rotate parts of the display and easily change size of all or part of the display. Cost prohibits a vendor from offering one device that will fill all the requirements in the school graphics field. A complete educational terminal would have to have several options for graphics so that each system would only be charged for as sophisticated a system as they required.

Graphics at its best will not allow fine enough detail to produce visuals required at most levels of school. This problem can be solved with 35MM slides, microfilm, or microfiche. Due to the difficulty in developing a mechanism that will randomly access slides in the time required I will not include a slide projector in my complete terminal. Concerning time allowed to access either an audio or visual message, we would hope to not make the student wait more than one second for the device to react. Random access in this amount of time has been achieved in fiche and microfilm mechanisms. Either of these mediums displayed on a suitable surface should allow for black and white or color views of sufficient detail to fill our needs That is not to say that the microform devices spoken of above have been adapted to an educational terminal, but that these parameters have been met for other applications and thus could be built into our terminal.

Many school systems presently have another visual output device that has proved effective in certain subjects. The device is the television monitor that can be connected to an educational

TV station, to a closed circuit TV network, or to a video tape player. Since our terminal is built around a television style CRT display device it is easy to provide the extra control to allow connection as a television monitor. Of course, it is too expensive a device to use only as a monitor replacement, but if it can be used as a monitor during the hours that it is not in use as a terminal, a cost savings should result.

There are several input devices that may be on only terminals that are used to generate lessons. These items could be put on all the terminals sold but, due to cost, they will probably only be used on lesson building terminals. The first of these special input devices is a graphics tablet. As we envision the tablet being used, a copy of the picture to be represented would be laid on the tablet and traced with the graphics pen. As the picture is traced, the internal microcomputer would translate the output from the tablet into information that allows the picture to be stored in the host computer and painted on the CRT screen. Once the teacher or media specialist has traced the graphic and is satisfied with the representation, a number could be assigned that picture and it could be included in the lesson plan for the associated course. The other input device that would seem necessary for one preparing lessons, but not for students taking the class is a special keyboard. Lesson generation would require a keyboard with text editing capability. The ability to insert and delete characters or words both from the line and from the display, plus other editing features, would be essential for the teacher or person building the lesson but would be unnecessary on most student terminals.

The peripherals I consider to be in the future include the video disc and a voice input device. Although Teldec is presently marketing a video disc in Germany and several firms are selling voice input devices, in this country I feel that neither of these items are commerically available in a form that they could be used in the cost sensitive educational environment. Predictions are that the video disc players and the discs themselves will be commercially available to the instructional market with all serious problems solved by 1980. That fits into what I would consider not this years educational terminal but the next generation of terminals available in the early 1980's. I can't even guess when voice recognition will be advanced far enough that a useful educational device will be available. I hope to someday see a device tied to a terminal that any student could talk into and the spoken word would appear on the screen, along with accent symbols showing exactly how the word had been spoken. This sort of device would allow deaf students to receive immediate feedback to assist them in voice training. The video disc and the voice input device, among others, will be showing up in the near future and with these on our minds, we will go into the area of manufacturing problems that have plagued Univac and others until a short time ago.

MANUFACTURING PROBLEMS

This section will briefly cover some of the problems associated with a hardware terminal. I define hardware to mean that each separate terminal function has it's own area of control logic that is performed by a number of integrated circuits and other discrete components. There are three major areas in which problems are encountered. The problems occur in definition, in development, and in field support of the terminal after delivery to the customer's office or classroom.

Definition Problems

The first problem is encountered during initial definition of a new product. In response to a marketing request a development engineering team is assembled to define a product. Given the constraint that funds do not usually exist to allow development of many different products, the functions of the product that will be built must encompass as broad a range as possible. At this point, in order to offer a single terminal that will fill a variety of applications, several things must happen. First, some of the specific requirements of each of the individual terminals must be sacrificed to hold down the complexity of the general purpose device. Secondly, for those specific requirements that can't be sacrificed, the cost and complexity of the general device must be increased. This means that the product not only does not do what you would like it to, but costs more than it should have to cost. Some of the problems mentioned could be solved if a market was established (well defined) and large enough to support a special terminal designed just for that application. Unfortunately, the education market in the past has not seemed to fall into this catagory. Banking and point of sale terminals provide evidence that market sectors do exist that are large enough to support individualized terminals.

The type of peripherals to be attached to the basic terminal must also be determined during the definition phase. Since each interface to the terminal must be well defined, any new input or output device that can't be so defined must be excluded. Once the interfaces have been defined and the product is under development, any changes in the list of peripherals will require development to stop while all associated interfaces are changed to adapt to the new device. The result of these requirements are that terminals either are 1) released with peripherals that are behind the state-of-the-art in capability or are, 2) not ready on their proposed completion date. The educational terminal has many unique and rapidly developing peripherals. Perhaps it has been rejected as a development candidate because of the uniqueness of these peripherals and their limited application, or because the technology related to these peripherals is changing so fast that it is difficult to release a product with state-of-the-art devices.

Development Problems

When the new product is fully defined some members of the definition group, plus some design engineers began to build the product. The problem encountered by this group becomes one of time rather than function, since the functions, however painfully limited, have been defined for them. In the development effort there are, with some variation, the following steps. The mechanical, electrical, and electronic design must be actually built to prove the feasibility of the initial definitions. After the device has been tested to the satisfaction of the group, a major commitment must be made to pay for all the tooling necessary for mass production of the terminal. Also, the printed circuit boards that contain all the discrete components that are, in fact, the personality of the device must be built. The printed circuit cards are the means to produce a reliable device inexpensively and are the major stumbling blocks to a flexible system. Once the printed circuit (PC) cards are available, all of the tests must be rerun to find all of the areas where, because of the peculiarities of the PC card, the device no longer performs the function correctly. Once the modifications have been made to correct the device operation it is necessary to begin the development of printed circuit cards that will work without modifications. This step is required because modiciations must be made by hand and are, therefore, very expensive. Finally the PC cards that have all corrections incorporated without hand modification must be fully tested in the final version of the terminal and only after this time can the manufacturing facility, which has been preparing to build this device, assign area and people to the product. It will be several months after this assignment is made before the first products are in the field. The time involved for this process is considerable, and for terminals it will generally be from 18 months to two years. As mentioned before, any time during development that it is decided that a newly released peripheral is a mandatory addition, the schedule begins to lengthen beyond the alloted two years. Also, unless the product is in a well defined field, changes in the product will be suggested and at times dictated by Marketing, due to knowledge gained from custmer inputs or by analyzing competitors new products. Of course any changes incorporated will also cause delays in the release of the product. These delays in the scheduled release of the product and the long time required to bring a product to release are the problems referred to as development problems.

Field Support Problem

The field support problem that I feel, is relevant to this discussion, is not the one we normally think of when we hear of field support. Although problems of hardware failure and incorrect operations are serious they are not peculiar to hardware terminals. The problem I speak of is that of terminal operation that is exactly like the designer planned but different than the customer understood when he commited to the piece of hardware. It is of no importance whether the problem arises from over zealous sales personnel, poorly written documentation, or customers who aren't sure what they meant The problem does arise and unfortuately, the solution is beneficial to no one. Either the customer must change his system to accomodate the undesired terminal, the seller must remove his terminal which wastes much of the customers terminal selection times, or the customer must pay to have expensive hand modifications made to his device so it fills his requirements. None of the above are desireable and dealings like these tend to leave a bad taste in everybody's mouth. One alternative to hardware terminals may be the new intelligent terminals. From my point of view, terminals which can be programmed by the customer or the seller to look like anything the customer desires, have a pair of limiting faults. The first is that they are a bit more expensive than a comparable terminal that is not customer programmable. The second fault, also related to expense, is that in order to use their full capability a programmer or group of programmers must be hired to write all the code that controls ther terminal. The solution discussed would be the microcomputer controlled programmed terminal.

For those who aren't familiar with the terminology, I will define both microprocessor and microcomputer. The microprocessor is a single or in some cases several integrated circuits (IC). Each IC is, at most, about three inches long, one inch wide and about ¼ inch high. These devices can internally perform the same functions that can be performed by large computers. However, a microprocessor cannot operate without some additional circuitry and other ICs. When the necessary control circuits, storage and working memory, power supplies, and an interface that allows connection of peripheral devices have all been added to the microprocessor, you have a microcomputer. Although this may sound like the microcomputer has grown into a large device, it is easy to develop a microcomputer that fits on one printed circuit card about 7 by 11 inches (excluding power supply). Add a few more of the same size cards with additional memory and you have a system that is more powerful and operates considerably faster than a computer of the early 1960s (that fit in several cabinets each three feet wide by two feet deep by six feet tall). Fortunately, our application will not even require the extra PC cards for memory so it will be very compact.

Microcomputer Advantages and Applications

Microcomputers, like computers, can be used in a variety of applications and do many different types of jobs. Their capabilities allow them to do number crunching jobs on one end of the spectrum to controlling small peripherals on the other end. Fortunately, they are inexpensive enough to compete with the control circuitry normally used to control peripheral devices and do the job cheaper, better, or both. It is in this area that we will dwell while talking of solutions for the problems covered in the previous section. You may remember that we defined our hardware terminal as one in which each separate hardware function is performed by a number of integrated circuits and other discrete components. In the microcomputer based terminal, almost all of the functions that would have been performed by that hardware are done by the microprocessor. Instead of having a section of hardware to react to every possible combination of keys, switches, and other type of input or output stimuli, a part of the program (stored in program control memory) watches these stimuli and reacts. The advantages to this system are twofold. First, the control circuits, and code storage chips to perform most functions are one third or less in number and also offer considerable savings in physical space and power consumption. Secondly, and what this whole paper has been aiming at, making changes in the code stored in the code storage chips is easy. To change a function or completely redefine a series of operations can be done in the code. Physically this means that one or more ICs are removed from sockets and replaced with others. Considering this, let us go back over the list of problems. No longer is it necessary to restrict our definitions to generalized pieces of hardware. Using some common components such as a basic frame and power supply, and some common control cards (i.e., microcomputer card, control storage memory card, and communication interface cards) many different terminals can be defined, each to meet the specific requirements of one application. In each instance the peripheral control boards for the type of devices that are required for that specific application are developed. Since the terminal only meets the requirements of one application it should be less expensive to build. We were concerned with peripherals that were only partially defined as our development began. It is usually easy to get a list of the signals that define the interface and even know what pins these signals will appear on. The difficulty arises in finding exactly what each of the signals is and what the time relationship between them is. Our development can be well along and we can still change the code to control a device without much, if any, effect on our scheduled release, if we find that what we guessed the peripheral would do in a particular instance was incorrect. Since our device is physically less complex it takes time to develop. Although it is difficult to estimate the exact time it will take to develop any product, since there are so many variables, I will give some approximate times so we can get a feeling of the advantages of the microcomputer based terminal.

Time to Develop (in months)

Job to Perform	Hardware Terminal	Hdw Mod.	Micocmptr Term.	Frmwr Mod.
Define Terminal	3	1	3	1
Design Terminal	1-3	1	1-2	1
Build Engineering Model	1	.5	1	.5
Test Engineering Model	2	1	2	2
Develop 1st Artwork	3-4	2	2	0
Build and Test 1st Artwork	2	2	0	0
Develop Mfg. Artwork	1	1	0	0
Build and test Mfg. Artwork	1	1	0	0
Totals	15-18	9.5	11-12	4.5

Note! Please remember that these figures are rough approximations and don't include the time necessary for the manufacturing facility to produce a product once they have received all of engineerings documentation on how it should be built. Although it is common for the manufacturing people to be working with Engineering during the entire development, it still takes some time after Engineering says the product is ready before a unit is ready for customer delivery.

As we study the table and we see advantages that will help to solve problems in both development and field support of the terminal. The development time is shortened and, as was discussed before, the time to add a new peripheral or modify a specific operation is lessened. Concerning field support and the problem of what was wanted vs what was received, it is probable that the problem won't show up as often, since the terminal is not a general purpose device, but one developed for a specific market. In the cases where the problem does arise, the time and cost to modify the terminal to fit the exact requirements will be small enough that the customer, or even the manufacturer if the order is large enough, may choose ~~to pay for the modifications.~~

Summary

We began by defining what would be, for our purposes, a complete educational terminal. What made this terminal unique was the type of peripherals that must be attached. The peripherals ranged from input devices such as the touch panel and joystick to output devices such as microfilm projectors and audible message generators. The peripherals that are in the wings for the next generation of terminals, the video disc and audio input devices, should also be considered so that as they are made commercially available they will attach to our terminal with a minimum cost to the customer.

Manufacturing problems fall into catagories of definition, time loss, and adaptability. It has been difficult to define a terminal that covered a broad enough market segment to be profitable and still include the special peripherals required in an educational terminal. The choices available now are to buy a simple keyboard-display terminal with no peripherals or purchase a very expensive basic unit. This choice is forced on us by the hardware implementations of terminals. The development effort of a hardware terminal is so time consuming, inflexable, and expensive that limited use terminals are not being built. As a continuation of the same problems, after a terminal is built the inflexibility of the hardware prohibits changing the operation to solve a customer problem or to add a new and useful peripheral.

The proposed solution to the problems as stated is to build a terminal around a microcomputer. Use a minimum of hardware and make what is there as general as possible, so that it can be used in a broad range of products. Put all of the peculiarities of the device in the programs that control the terminal and put the programs in replaceable chips in the terminal. To make a change the terminal must be reprogrammed and some chips changed but the hardware does not have to change. To add a peripheral a new board must be designed but it will fit in a slot already available in the terminal. The basic terminal need not be redesigned to adapt to the new device as the device will work from the computer control lines.

All the changes necessary to the code to add a peripheral would be furnished by the terminal vendor as he attached new devices to the terminal. There would still be times when the operation of the terminal is not what the customer desires but the time, effort, and expense to change will not be prohibitive as they are today. I may sound like I have an interest in a microcomputer company, but a careful study of the facts will lead to our same conclusion. The microprocessor will allow vendors to begin to fill the needs in different application oriented areas. This is because the microprocessor allows low development costs through common hardware and maximum flexibility through changable programs.

References:

1. G.H. Culp, J.C. Gilbert, J.J. Lagowski, P.L. Stotter, "Computer-Based Instructional Techniques in Undergraduate Introductory Organic Chemistry: Rationale, Development Techniques, Programming Strategies, and Evaluation." Project C-BE EP-12

2. Murry C. Goddard "Kodak's Computer-Based Training System" Eastman Kodak Company Document.

3. D.J. Monson "Present Trends in Instrumental Technology" IEEE 1975. Region 6 (Western USA) Conference pp 161-163

USING THE PLOTTER TO INVESTIGATE SKETCHING IN UNUSUAL METRICS

W.A. Stannard

Eastern Montana College, Billings, Montana 59101

ABSTRACT: By accepting various definitions for "distance" (which are consistent with accepted properties for measure) one can obtain unusual outcomes for the conic sections when they are sketched by the plotter. The paper discusses the "new" conics, displays plotter results, and raises questions which the reader may pursue.

The criteria for "good" ways to measure distance have been clearly established by the following definition of a metric space:

A pair of objects $[X,d]$ where $X \neq \emptyset$ and $d: X \times X \to$ Reals is a metric space provided for all $x, y, z \in X$:

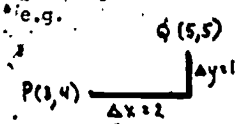
1. $d(x,y) \geq 0$;
2. $d(x,y) = 0$ iff $x = y$;
3. $d(x,y) = d(y,x)$;
4. $d(x,z) \leq d(x,y) + d(y,z)$.

Here d is of course the distance function.

Once this characterization of "good distance" criteria has been established, it is possible to investigate some "unusual measures of distance" which meet these criteria. Three such distances are:

1. $d_1(P,Q) = |\Delta x| + |\Delta y|$ where Δx is merely the horizontal displacement between P and Q and Δy is the vertical displacement.

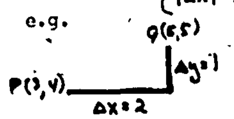
e.g.



$\therefore d_1(P,Q) = 2 + 1 = 3$

2. $d_2(P,Q) = \begin{cases} \max(|\Delta x|, |\Delta y|) & \text{if } |\Delta x| \neq |\Delta y| \\ |\Delta x| & \text{if } |\Delta x| = |\Delta y|. \end{cases}$

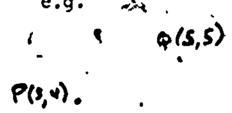
e.g.



$\therefore d_2(P,Q) = \max(1,2) = 2$

3. $d_3(P,Q) = \begin{cases} 1 & \text{if } P \neq Q \\ 0 & \text{if } P = Q \end{cases}$

e.g.



$\therefore d_3(P,Q) = 1$ since $(3,4) \neq (5,5)$

The fact that each of these three distances meet the criteria of being "good" is left to the reader.

Now consider the four conic curves by definition:

A. Circle - the location of all points in the plane a given distance from a fixed point (center).

B. Parabola - the location of all points in the plane equidistant from a fixed point (focus) and a fixed line (directrix).

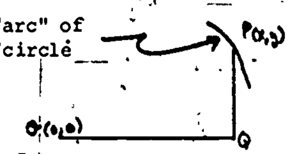
C. Ellipse - the location of all points in the plane such that each point P of the curve is located with respect to the foci F_1 and F_2 such that $PF_1 + PF_2$ is constant.

As a selected sample of these conics, let us examine those resulting from d_1.

First, the circle of radius 4 with center at the origin would by equation be:

$|x-0| + |y-0| = 4$ or simply $|x| + |y| = 4$.

"arc" of circle



Note: $d_1(O,P)$ is measured along OQ and then QP. Since $|OQ| + |QP| = 4$, $|x| + |y| = 4$ is the condition of the curve being a circle.

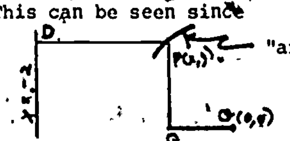
When plotted on a plotter, $|x| + |y| = 4$ yields the circle as:

(See Figure I)

Second, consider a parabola with focus at O (0,0) and directrix $x = -2$ under the constraint of d_1. The definition of the parabola yields:

$|x| + |y| = 2 + x$.

This can be seen since

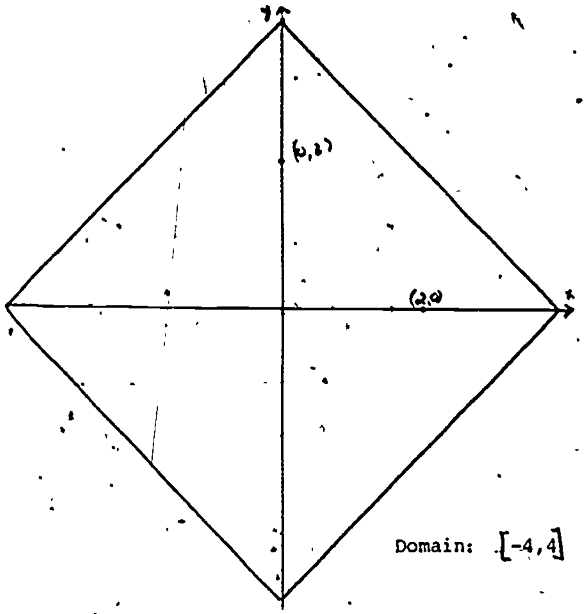


$d_1(D,P) = d_1(P,O)$ or $|DP| = |PQ| + |QO|$ or $(2 + x) = |y| + |x|$.

Using the plotter results in the curve:

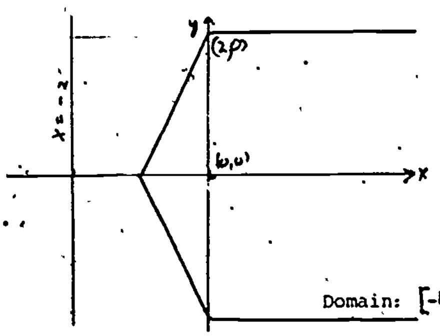
(See Figure II)

Figure I



Domain: $[-4,4]$

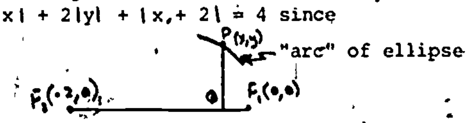
Figure II



Domain: $[-1,\infty)$

Third, consider the ellipse with foci F_1 ~~(0,0)~~ and F_2 $(-2,0)$ and constant sum 4. The equation of the ellipse must be:

$|x| + 2|y| + |x + 2| = 4$ since



where $d_1\ (F_2,P) + d_1\ (F_1,P) = 4$

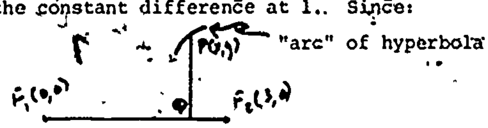
or $|F_2Q| + |PQ| + |PQ| + |QF_1| = 4$

or $|x + 2| + |y| + |y| + |x| = 4$.

Again, using the plotter results in an interesting curve. Can you predict it? It is:

(See Figure III)

To complete the conic study for d_1 now consider the hyperbola with foci F_1 $(0,0)$ and F_2 $(3,0)$. Set the constant difference at 1. Since:



$|d_1\ (P,F_2) - d_1\ (P,F_1)| = 1$

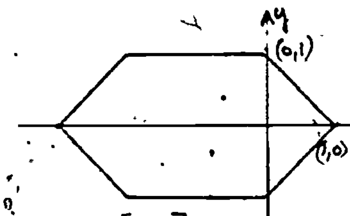
or $\left| (|F_2Q| + |PQ|) - (|PQ| + |QF_1|) \right| = 1$

or $|(3 - x) - (x)| = 1$ or $|3 - 2x| = 1$.

Can you predict that the hyperbola would thus plot as:

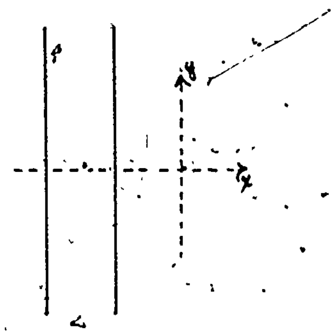
(See Figure IV)

Figure III



Domain: $[-3,1]$

Figure IV



The conic under d_2 are similarly "unusual" in that a circle of radius 2 with center at the origin is a square of side 4 centered in a horizontal position about the origin. The ellipse with foci at $(-1,0)$ and $(1,0)$ and constant sum 4 is centered about the origin and shaped like a stop sign. The parabola with focus $(0,0)$ and directrix $x = -2$ is composed of two "wings" whose slopes are +1 and -1 and a "blunt nose" along $x = -1$. And, the hyperbola with foci at $(0,0)$ and $(3,0)$ and a common difference of 1 is shaped like two opposing parabolas (for d_2).

Lastly, the conics for d_3 are interesting. Suppose, for example, the circle of radius 1 is to be plotted with center at the origin O.

• P (x,y)

O (0,0) •

Since $P \neq O$, $d_3\ (O,P) = 1$ and thus P is on the circle. But this would be true of any point X where $X \neq O$. Thus the entire plane, except the origin, is on the circle. On the other hand, if the radius were 2, there would be no points of the plane on the circle.

The reader is left to examine the fate of the parabola, ellipse, and hyperbola under d_3.

In conclusion, it is an interesting problem to define the parametric equations of the conics under each metric such that a simpler application of the plotter is enjoyed. As a representative sample solution, consider the circle of figure I. Parametric equations for this circle are:

$$x = \frac{4 \cos \Theta}{|\sin \Theta| + |\cos \Theta|} \text{ and } y = \frac{4 \sin \Theta}{|\sin \Theta| + |\cos \Theta|}.$$

THE EVOLUTION OF THE SCHOOL INSTRUCTIONAL COORDINATOR

James L. Altermatt
Regional Instructional Coordinator
Minnesota Educational Computing Consortium

ABSTRACT: The computer has proved to be an effective instructional tool in Minnesota. The Minnesota Educational Computing Consortium has facilitated an increasingly rapid growth rate of instructional computing. Now that the quantity of computing is significant, attention is quietly turning to improvement of the quality of use. Individual school concern is the responsibility of the coordinator for instructional computing. How the job of school computer coordinator is evolving in Minnesota is the subject of this paper.

Introduction

Computer services in Minnesota at all educational levels are coordinated by MECC through its three divisions: Instructional Services, Management Information Services, and Special Projects. This paper has particular reference to the Instructional Services Division and the instructional timesharing network as it serves elementary, secondary, and vocational schools (E/S/V) throughout the state.

Growth of Instructional Timesharing in Minnesota

Prior to the establishment of MECC by the 1972 Minnesota legislature, instructional computing in elementary/secondary schools was confined largely to the Minneapolis-St. Paul metropolitan area, as there were only a few small timesharing projects elsewhere in the state. The number of terminals in use in the past and present are as follows:

Table 1

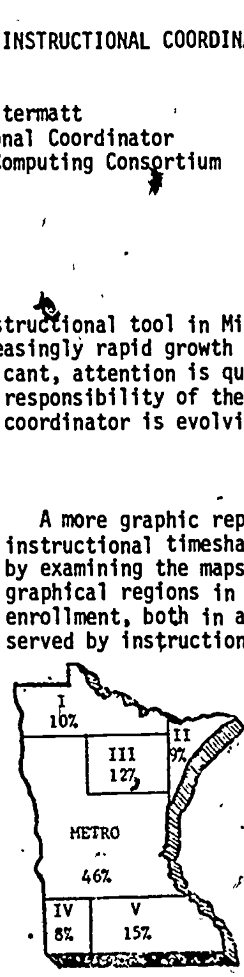
	E/S/V Terminals in Use
1973 - 1974	400
1974 - 1975	700
1975 - 1976	800
1976 - 1977 est.	900

Table 2 shows the numbers of students and school districts in the state and the proportions of those that have access to instructional timesharing:

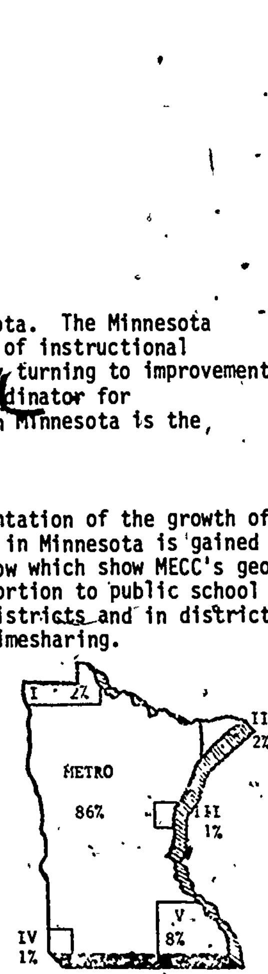
	Total State E/S/V enrollment (K-12)	Enrollment in Districts w/Timesharing	% of Total	No. of School Districts w/ Timesharing*	% of Total
73-74	990,000	484,000	49%	78	18%
74-75	950,000	775,000	82%	243	56%
75-76	911,000	830,500	91%	307	70%

* Total number of school districts in the state: 436.

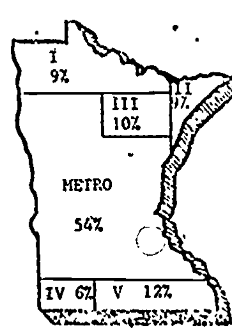
A more graphic representation of the growth of instructional timesharing in Minnesota is gained by examining the maps below which show MECC's geographical regions in proportion to public school enrollment, both in all districts and in districts served by instructional timesharing.



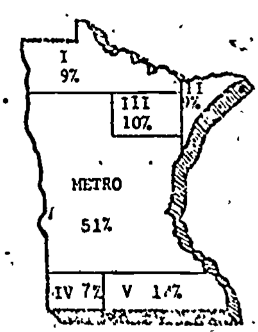
Map A...MECC E/S/V Regions shown in proportion to the average public school enrollment (K-12) in each region (73-76).



Map B...MECC E/S/V Regions shown in proportion to the public school enrollment served by instructional timesharing in 1973-74.



Map C...MECC E/S/V Regions shown in proportion to the public school enrollment served by MECC in 1974-75.



Map D...MECC E/S/V Regions shown in proportion to the public school enrollment served by MECC in 1975-76.

Need for Quality Usage

The substantial growth of computer usage in the schools having been demonstrated, it now remains to be shown how the quality of that usage has been and continues to be developed. Obviously on a statewide project of this size, quality usage cannot be developed by a few and serve the needs of all. To attain quality usage, an effort must be made within each school building that has access to the computer by a terminal device. The superintendent of each school district has been asked to designate a district coordinator for instructional computing, and for the larger districts to designate a coordinator for computing in each school building as well. The school coordinator is the key to quality usage of the computer within the school. Success or failure of each school's computing program, which affects the total district computing curriculum, in large part depends on the coordinator. The work of the coordinator is thus reflected in the total statewide instructional computing effort, which ultimately must provide answers to the question of whether or not instructional computing has educational value. The following discussion of various dimensions of the coordinator position may be helpful as a model for others trying to implement a similar concept.

Qualifications of School Coordinator

The job of school coordinator calls for many talents ranging from journalistic ability for writing newspaper and communication articles for school and local media, to library techniques, programmer skills, and accounting knowhow for keeping track of usage, time, and storage. At times the school coordinator is expected to act as a terminal technician and be able to explain the terminal operation to students, teachers and community people. The school coordinator needs to be a good evaluator of the computing program and be able to recommend and implement changes.

Educational backgrounds of school coordinators vary, but one certain prerequisite is enthusiasm for, and knowledge of the vital role of computers in the educational process. Some coordinators come to the job without special prior training and learn by doing. Others may be pursuing, or may have attained a degree in computer science. Two Minnesota universities, Mankato State and Moorhead State, currently offer four-year degree programs in computer science teaching. A specialist degree for coordinators is now being offered in some colleges.

Support for the School Coordinator

MECC strongly encourages school districts to support the school computer coordinator by providing time to attend and sponsor workshops. Holding individual and small group instruction seminars, acting as advisor to computer clubs, and supporting other required computing activities are also strongly encouraged. School coordinators in a given region of the state are assisted by the MECC Regional Instructional Coordinator living in that region. Regional Coordinators meet monthly to share ideas and materials contributed by educators throughout the state. In turn, school coordinators meet with regional coordinators to exchange ideas and information.

What Does the School Coordinator Do?

The tasks of the school computer coordinator are many:

Computer responsibility Users are taught to respect the limitations of the computer system and are made responsible for their use. They are taught to respect the privacy of each individual program and to understand the timesharing concept and use the system accordingly.

Duties Duties assumed by an effective coordinator vary according to the needs of the school year:

Beginning of School Year

a) Attending computer in-service workshops

b) Determining a set of objectives with his administrator

c) Assisting in finding suitable place to house the terminal

d) Ordering supplies for the terminal

Throughout the School Year

a) Getting teachers interested and involved

b) Keeping faculty members notified on new programs and ideas

c) Scheduling student and teacher use of the terminal

d) Representing his school district at computer meetings and workshops for school coordinators

e) Keeping administrators updated on computer related information such as the latest in equipment and applications

f) Corresponding with MECC coordinator in his area

g) Looking for funding possibilities; federal, state, and local agencies and industries

h) Keeping equipment in good working order

i) Maintaining equipment in good working order

j) Acting as the contact person between MECC and his community concerning special requests

Duties (continued)

End of School Year

a) Evaluating the achievement of the objectives set at the beginning of the year

b) Preparing an annual report for administration and school board

c) Recommending computer courses that should be added to the curriculum

d) Reordering computer books, files, and periodicals that should be added to the school library

e) Having equipment serviced for preventive maintenance

f) Recommending any enhancements that should be made to the computer program

g) Evaluating the need for additional time assistance, and compensation to do the job and maintain the program most desirable for his school

Involving Others Some methods used for getting teachers interested and involved in the use of computers are:

a) Working on a one-to-one basis as much as possible during a first exposure to the terminal

b) Showing how easily copies can be made for the whole class by having output printed on duplicating paper

c) Demonstrating the use of simulations

d) Assisting in the preparation of the use of computers in the classroom

e) Volunteering to teach a class for the teacher who is hesitant to try it

f) Assisting in arranging for the hardware for a video display so that all the students can see without difficulty

g) If programs need to be written, finding someone who can write the program

h) Having knowledgeable students available to assist teachers and other students as they use the terminal

i) Encouraging students to find computer applications in multiple areas and introducing these to teachers

Public Relations The effective coordinator tries to build awareness of the educational value of computers not only through contact with teachers and students, but also through contacting the community at-large.

a) Writing articles for school, and local and professional organizations

b) Putting on workshops and speaking at meetings

c) Entering or sponsoring computer-related contests

d) Inviting local media in for special reports

e) Arranging for local legislators to visit and see how instructional computing is used in the school

Summary

The school coordinator is the key to whether or not more instructional computing doors are opened for the total school staff. A school's commitment to instructional computing should not stop at dollars for computer time and equipment. A total commitment must be complete and will be successful, only if it is recognized that the school must have a coordinator who is given the opportunity to show the possibilities of instructional computing.

References

1. Minnesota Educational Directory, 1973-74, State of Minnesota, Department of Education.

2. Minnesota Educational Directory, 1974-75, State of Minnesota, Department of Education.

3. Role of the School Coordinator, 1975, MECC Minnesota Educational Computing Consortium 2520 Broadway Drive, Lauderdale, MN 55113.

AN ARGUMENT FOR GAMES IN COMPUTERIZED DRILLS AND TUTORIALS

Muata Weusi-Puryear

SCRDT, Stanford University, Stanford, California 94305

ABSTRACT: This paper is presented as an argument for the incorporation of games in computerized drills and tutorials. It is based on experiments conducted at the Lawrence Hall of Science, University of California at Berkeley. The experiments sought an answer to the question: "Can the game elements of a computerized tutorial/game motivate student involvement in the tutorial elements to a degree high enough to produce significantly greater achievement than the tutorial elements alone can produce?" The study supports an affirmative response to the question. Students, who had anopportunity to continue their participation in a game if they correctly responded to randomized exercises, achieved significantly more than students who did not have the opportunity to play, even though the game-playing students did fewer exercises.

Introduction

In the teaching of mathematics in elementary schools it is becoming increasingly popular to use games to aid instruction. The pages of professional magazines often contain advertisements for specially-designed games to teach arithmetic to elementary school children. The shelves of school supply stores are replete with games of varied descriptions. Schools with instructional computer systems create computerized games and trade them with other schools. In 1970 Gordon[1] attributed the following quotation to Boocock and Schild -- it articulated a major theoretical problem concerning the pedagogical value of games.

> "... the game designers are in the peculiar position of having a technology, or applied science, before the theoretical science has been developed. While most observers would agree that games do teach, what they teach and why are yet to be precisely measured."

In 1975 Weusi-Puryear[2] partially answered Boocock and Schild's questions --

> "Games can have a positive pedagogical value to mathematics instruction. The motivative effect of games can be great enough to significantly increase students' achievement levels. In particular, games can significantly improve the learning that takes place during a computerized drill-and-practice lesson."

The study that led to the above conclusion is presented in summary form in this paper, it constitutes an argument for the incorporation of games in computerized drills and tutorials.

Statement of Problem

There are a few educational research papers that show that students in experimental groups using games have done better (and significantly better) than students in non-game control groups. However, it might be successfully argued that these differences were due to Hawthorne effects, school differences, teacher differences, or teacher biases. By the very nature of most educational research that is done in the classroom, it is difficult (if not impossible) to positively assert that a particular variable produced a particular effect.

In the development of a theory of the pedagogical value of games to mathematics instruction it will be necessary to determine under "laboratory--like conditions" if the motivative effect of a game in teaching arithmetic can produce significantly greater learning. (Here "laboratory-like conditions" means a setting and design that closely approximates the type of experimentation done in the physical sciences -- where a quantity is systematically varied and the effects of this manipulation on other dependent variables are measured.)

This was a study designed to add to our knowledge of the pedagogical value of games to mathematics. It sought an answer to the question: "Can the game elements of a computerized tutorial/game motivate student involvement in the tutorial elements to a degree high enough to produce significantly greater achievement than the tutorial elements alone can produce?"

Procedure

The study used a computerized tutorial/game called GAMBO. There are two players in a GAMBO tutorial/game. One player is "the student". The other player, who shall be called "Jody", is a computer simulated player. The computer program also exhibits a third personality called "Gambo" who is the umpire and scorekeeper for the game. The teletype output from the GAMBO tutorial/game looks like a script of a play with three characters -- Gambo, the Student, and Jody. Each player in turn is given an arithmetical problem to solve: after he has committed himself to an answer, his opponent is given an opportunity to evaluate the player's answer: the umpire presents the correct answer: if the player's answer was correct, he gets an opportunity to make a move on a three-by-three Tic-Tac-Toe board: and if the player's answer was incorrect he would forfeit his turn on the Tic-Tac-Toe board.

The object of the game is to gain as much score as possible in a limited amount of time. In the GAMBO game a player can receive ten points for correctly answering the problem, five points for correctly evaluating his opponent's answer, and fifteen points for getting three marks in a row on any Tic-Tac-Toe board. A new Tic-Tac-Toe board is presented after either player receives three marks in a row.

Two experiments were designed to be part of one-day-field-trip activities for summer school classes of San Francisco Bay Area school districts. Each field trip started with students arriving at the Lawrence Hall of Science by bus from their school districts in mid-morning. Approximately 30 students (depending upon the number of terminals available that day) were randomly selected from among the students in the 8- to 11-year-old age range. Those students not involved in the experiments were taken to the activity halls where various scientific activities were available for them. The students in the experimental group were placed in every other seat of a small auditorium where they received a battery of tests. These pre-experimental measures were intended to measure the students' skill in the basic arithmetical operations and to inventory their attitudes towards arithmetic and arithmetic stressful situations (such as tests). While they were seated they were, without their knowledge, randomly divided into three sub-groups. The selection was based upon that day's random assignment of the auditorium seats. After the pre-experimental testing, two of the sub-groups were escorted from the auditorium to computer terminal rooms containing enough teletype computer terminals for each member of the group. During the next forty minutes: one group received a computerized tutorial; one group received the same computerized tutorial interwoven with a simulated Tic-Tac-Toe game; and the group remaining in the auditorium viewed two films of scientific interest but having no relevance to the tutorial. The computerized tutorial was the GAMBO system's. Students under 10-years-of-age received addition problems and multiplication problems were given to the older students.

The first two groups then returned to the auditorium where a post-test was given to all students on the material covered in the tutorials. Students then left the the auditorium and together with the non-experimental group of students had a bag lunch. After lunch, all students had a couple of hours to explore the Lawrence Hall of Science. All students had an opportunity to play games on a computer without regard to their previous groupings. In addition, other scientific activities were available for the students -- physics, biology, chemistry, astronomy, and mathematics. The experimental situation carried no risk to the students -- the entire field trip activity was designed to give all students a positive experience with science.

In each experimental situation there were three groups of participants: The full-treatment group was exposed to the GAMBO system as described above. The partial-treatment group was exposed to the same GAMBO system but they did not see any reference to a Tic-Tac-Toe game and all score-referenced output was surpressed. The no-treatment group watched two National Geographic films -- <u>The Chick Embryo</u> and <u>Succession from a Sand Dune to a Forest</u>.

The partial-treatment group corresponds to what is commonly called a "control group". As closely as possible we tried to make the difference between the full treatment and partial treatment the presence or absence of the game elements during the experimental phase. The full- and partial-treatments were conducted simultaneously in separate rooms. Through scheduling we controled for possible room effects, sex of laboratory assistants effects, and their possible interactional effects. A teacher from the participants' school was present in each experimental room to aid with disciplinary problems, but they were discouraged from actual participation in the experiments.

The purpose of the simulated opponent, Jody, was to provide a human-like opponent who was "equal" to the student in ability -- the probability that Jody would give a correct answer was equal to the proportion of correct to total answers previously given by the student. When we were in the checkout phase of the GAMBO system we ran sessions with elementary school children similar to those run later during the experimental phase. We observed that students did not give the impression that they were competing with a super-human machine. On several occasions the students remarked that "Jody is stupid," or "Jody is trying to cheat," or attributing other human characteristics to Jody. In addition to the errors Jody made, there was Jody's child-like method of typing. Whereas Gambo's presentations were done at a speed of 10 characters per second, Jody's presentations were made at a speed of (at best) 1 character per second. When Gambo had a message to type the message was sent to the teletype in its totality, hence when the teletype received a message it was typed at the maximum speed of 10 characters per second. On the other hand when Jody had a message to type it was sent to the teletype 1 character at a time and this allowed the presentation of the message to be interrupted by other users. Hence, when it was ready to send out the next character, a time lapse that may have varied between a few micro seconds and two seconds would have occurred. This difference in time was measurable and the program checked to make sure that the delay was greater than 1 second before it sent out another character. Hence, it appeared to someone watching Jody's presentation, that Jody was doing exactly what the students were doing when they typed -- that is, searching the keyboard for the next letter or digit.

The function of a no-treatment group is to statistically smooth out the effects of the highly motivating field trip atmosphere. By regressing pre-experimental measures on post-test scores, using the no-treatment group, we are able to obtain formulas for predicting post-test scores of the two experimental groups. These predicted scores used as covariants in an analysis of covariance on the post-test measure provide additional statistical control. Since the no-treatment group is a randomly selected sub-group of the experimental participants, and we can safely assume that they received no training in the concerned algorithm during the experimental period, their performance on the post-test items can serve as a statistical basis for measuring the difficulty of the items.

Summary of Analysis Results

There were 258 students who participated in the two experiments -- 84 in the "addition" experiment, and 174 in the "multiplication" experiment.

The students' attitudes toward arithmetic and their levels of debilitating anxiety did not differ significantly from those of a large national sample of fourth-graders[3] taken in 1962.

Grade, age, and attitude were much less able predictors of post-test score than pre-experimental computational ability.

Most of the errors made by the students were either very unusual misapplications of the addition or multiplication algorithms or they were miscalculations of simple (one digit) sums or products.

The students' Tic-Tac-Toe strategies were normally distributed, but their correlations with other measures were not significantly different from zero.

Students, who had an opportunity to continue their participation in the game if they correctly responded to randomized exercises, achieved significantly more than students who did not have the opportunity to play -- even though the game-playing students did fewer exercises.

Conclusions

Games can have a positive pedagogical value to mathematics instruction. The motivative effect of games can be great enough to significantly increase students' achievement levels. In particular, games can significantly improve the learning that takes place during a computerized drill-and-practice lesson.

Footnotes:

1. Gordon, Alice Kaplan. Games for Growth: Edcational Games in the Classroom. Palo Alto: Science Research Associates, Inc., 1970, Library of Congress Catalog Card No. 72-120693.

2. Weusi-Puryear, Muata. An Experiment To Examine The Pedagogical Value Of A Computer Simulated Game Designed To Correct Errors In Arithmetical Computations. Ph.D. Dissertation, Stanford University, 1975. Dissertation Abstracts International, Volume XXXVI, Number 4, 1975. Xerox University Microfilms, Ann Arbor, Michigan, Order No. 75-21.906, 157 pages.

3. The National Longitudinal Study of Mathematical Abilities

THE COMPUTER AND THE READING TEACHER

Linda L. Johnson

South Mountain High School, Phoenix, Arizona 85040

ABSTRACT: Reading teachers at South Mountain High School in Phoenix, Arizona, have been rescued from tedium and provided a far more comprehensive remediation program for students. The district computer permits teachers there to enter a student's name, reading level and needed skills. Thereupon it prints all materials that teach these skills at the student's level in the form of an assignment sheet. Eventually, easy-to-understand objectives and sample post-test items will be included. Teachers also use local computers to run computerized readability formulas and to check a lengthy diagnostic test. Students use the computer to learn various reading skills.

The computer is not yet the indispensable tool of reading teachers--for years they have survived quite well without it--and still do. However, as soon as they learn what the computer can do for them, and as soon as they obtain easier access to this miracle machine, it is doubtful they will ever allow themselves to be without one again.

At South Mountain High School in Phoenix, Arizona, the impact of the computer on reading teachers is already being felt. There the computer has influenced almost every area of the reading teachers' responsibilities.

First introduced to South Mountain reading teachers in the summer of 1975 when two teletype terminals were installed in the reading laboratory at the school, the computer was almost immediately put to use for storing data in a "Materials Retrieval Program," a project that revolutionized the placement of students into suitable reading materials. Other programs, helping with instruction, drill and determining readability, were added within seven months. All programs were written in BASIC and run on the Hewlett-Packard 2000F Time-sharing System, located at the Phoenix Union High School District offices, to which the two terminals were connected. Since no money was available for purchasing computer programs, all programs were written either by the teachers themselves, by the District Educational Computer Specialist, or by other programmers donating their efforts.

Materials Retrieval Program

Ready by September of that year was the most far-reaching undertaking: The Materials Retrieval Program.

This use of the computer was originally conceived as a way of assuring that benefits of the computer would reach each student and teacher in the reading program.

Prior to the arrival of the computer, the reading program operated with teachers spending hours diagnosing students and then "programming" them into materials that would give instruction and practice in the individual's weakest reading skills. The materials and skills were so numerous that is was difficult for teachers to remember, for example, in which workbook there was a particular page giving instruction and practice in distinguishing facts from opinions. As a result, many good materials were often neglected in favor of a small number of familiar materials. Furthermore, the process of programming five classes of 25 students into appropriate materials suited for each individual was a long one. And once programmed, students were not often reprogrammed unless absolutely necessary.

The computer has solved these problems. Now when students are assigned to a reading class, they can be programmed into appropriate materials rapidly. The teacher makes the usual diagnosis of reading difficulties and grade level of each individual. This information is typed into the computer. The computer then prints out an assignment sheet for each student with just those materials that teach the requested skills at the desired grade level.

The student chooses from usually lengthy lists which method of instruction and practice he wants, be it from a familiar workbook, a ditto, a game, or even from a computer terminal. As soon as a student and teacher feel that the student is ready, a post-test is given and the student may go on to other skills.

A high-speed printer at the district computer center enables assignments for entire classes to be stored and then printed rapidly for distribution to individuals by the following day.

In order to compile the data involved in the Materials Retrieval Program, teachers scanned page after page of workbook and textbook materials for skills and their page numbers. Other materials such as tape-recorded or film-stripped lessons were also examined. Distinguishing especially clear instructional material from that which gave merely drill and practice was begun. Over 300 books and workbooks were entered and over 100 skills were isolated.

Computer programs, written by the Educational Computer Specialist, entered, stored, edited, and printed the data in a form comprehensible to the student. These core programs, now in operation, are being augmented to include a number of innovations that in the near future will result in a teacher-developed, computer-managed instructional system.

For many of the 100-plus skills now listed, instructional objectives are being written in the student's own language. When the student sees his print-out next Fall, he will be able to determine exactly what about the skill he will be expected to learn. At the end of the print-out, sample post-test questions will be listed. When the student and teacher feel he has attained the objectives and can answer these types of questions, the student will be post-tested.

This computer-managed system could be expanded even further in the future. A division of the materials could be made so that each lesson is listed under the appropriate instructional objective within the skill area.

Because of the quantity, some materials listed on the present print-out are hard to locate in the classroom. Adding symbols to the print-out titles, putting corresponding symbols on the materials, and labeling classroom locations will alleviate that problem.

Comprehensive Achievement Monitoring

Developing sufficient pre and post test questions and writing objectives would seem to be extremely time-consuming for the teachers of one school to undertake, but these problems are rapidly being solved by another project, called the Comprehensive Achievement Monitoring System (1). This plan involves many reading teachers in the school district who are developing hundreds of test questions and writing objectives in a large reading project of their own.

The questions they write are printed on cards, one question to a card. By overlapping the cards on a carrier, any number of Zeroxed tests can be made. The computer's role is to score answer cards filled out by the student. The computer compiles summary data on each student's test results and a summary of an entire class's progress, also preparing at year's end an item analysis of the test questions.

CAMS is useful to any reading teacher in teaching a wide variety of reading skills and competencies. Eventually, it may be combined with the Materials Retrieval Program to further develop a computer-managed instructional system. When more computer storage room becomes available, CAMS questions could be entered, themselves, into the computer for even faster test construction. When this becomes a reality, an individual test suited to only one student could be rapidly constructed and scored.

Computer-Assisted Instruction

One way a student can learn a reading skill at South Mountain High School is through computer-assisted instruction. Although very few studies have been made of the effectiveness of this method of instruction, it does seem strongly motivational. Most of the students appear to love to use either of the two terminals located in the reading center; moreover, they seem to pay rapt attention to the instruction.

At the moment there are approximately twenty-two teacher-made instructional programs in reading available for student use. These teach or drill in several phonics skills, structural analysis, context clues, classification, and fact or opinion.

Several of the eleven reading teachers at the school have been involved in two workshops to learn BASIC computer programming and the Hewlett-Packard Instructional Dialogue Facility (2), offered by the school district for system credit. In addition, teachers may create simple instructional drills by filling out dittoed formats and typing them into the computer. These can be quickly done and require no knowledge of computer programming.

Future plans include the use of the tape recorder, slides or filmstrips which will be incorporated into additional instructional computer programs.

Diagnostic Testing

One of the reading teacher's most important tasks is accurage diagnosis of the student's skill deficiencies and reading levels. A widely-used device for testing students who read below the seventh grade level is the Silent Reading Diagnostic Tests (Bond, Balow and Hoyt) (3). A computer program developed at Arizona State University not only scores the eight subtests in the booklet but prints out errors made by the student so that a thorough error pattern analysis can be made. Many errors are classified by the computer so that they can be analyzed by the teacher. The computer program accelerates grading and enhances the diagnostic features of the tests.

Also being developed at Arizona State University is a reading test retrieval program. Soon, a teacher may be able to request a listing of just those tests from a 600-test bank that meet the teacher's specifications. If he has a ninth grade student reading on a very low level, he will be able to request tests that would help diagnose that type of student's needs.

Readability Formulas

The reading teacher is often called upon to determine the readability, or level of difficulty, of textbooks or other reading materials. The best readability formulas they use are often extremely time-consuming and complicated.

The Dale-Chall Readability Formula (4), for example, involves five steps the teacher must make. First the teacher must count 100 words from a typical passage. Second, he counts the number of sentences. Third, he looks up most of the words on a 3000-word list. Fourth, he must remember some 40 rules as this list is being consulted. Fifth, he makes seven computations involving large numbers. These procedures must be repeated for the several samples of the book that is being analyzed.

By computerizing the Dale-Chall, all of these steps are eliminated. A teacher merely types in each sample; the computer does the rest. The Dale-Chall computerized formula can be accessed on the UNIVAC computers located at Maricopa County Community Colleges and at Arizona State University. This formula determines readability of samples above the fourth grade level.

On the Hewlett-Packard computer as well as on the UNIVACs, is a formula for determining readability below the fourth grade level--the Spache Formula (5). With this formula, South Mountain reading teachers can perform readabilities on some materials without leaving the school. Soon, the Flesch Formula (6), for fifth grade and above, will be available in the school.

Other Possibilities

The above-mentioned uses of the computer in reading education are only a beginning. As more storage and memory becomes available, other schools in the eleven-school district will be able to use all of the programs. As more and more reading teachers come to understand the capabilities of the machine, new ideas will be developed and tried.

One more idea that will be on its way in the next few months will be a retrieval system for pleasure-reading books by level. A student will be presented with a list of books on his level divided into such categories as romance, sports, adventure and science fiction. This program will encourage the creation of a collection of a wide variety of books in each classroom and facilitate the choosing of appropriate books to read by the students.

In an individualized setting of twenty or more students, often reading well below grade level, the teacher has been moved from instructor to manager. With the computer, hopefully, the teacher can spend less time managing and more time instructing. The Materials Retrieval Program will include eventually not just materials that teach, but instructional strategies for teachers and teacher-aides, ideas for teaching a skill in small groups or one-to-one.

A teacher supplied with computer terminals in the classroom will find that through their use, diagnosis, instruction, evaluation and services to other teachers will greatly improve. The computer will never replace the reading teacher. It will create a much more efficient and skillful teacher instead.

Sources

1. Comprehensive Achievement Monitoring
   Copyright, 1973 by
   National Evaluation Systems, Inc.
   Post Office Box 226
   Amherst, Maine 01002

2. Instructional Dialogue Facility
   Copyright, 1973 by
   Hewlett-Packard Company
   11000 Wolfe Rd.
   Cupertino, California

3. Silent Reading Diagnostic Tests
   Copyright, 1970 by
   Meredith Corporation
   Lyons and Carnahan, Inc., Educational Division
   407 E. 25th St.
   Chicago, Illinois 60616

4. Dale, Edgar and Jeanne Chall. "A Formula for Predicting Readability." Educational Research Bulletin, vol. 27 (January 1948), pp. 11-20, 28.

5. Spache, George. "A New Readability Formula for Primary Grade Reading Materials." Elementary School Journal, vol. 53 (March 1953) pp. 410-13.

6. Flesch, Rudolf. <u>How to Test Readability</u>. New York, New York: Harper & Brothers, 1951.

GENIE - A PROBLEM GENERATOR FOR HIGH SCHOOLS

Allan H. Paschke

Niles Township High Schools
Skokie, Illinois

GENIE is a total software package which provides a full-scale computer-assisted test construction (CATC) capability on a mini-computer. GENIE was designed and implemented by the author for the Niles Township High Schools, a Chicagoland suburban district of 7500 students. The GENIE system has been in use since September, 1974, and is running of a DEC PDP/8 computer. It has become an accepted and very widely-used facility in these schools, with applications in four major subject areas.

In this paper, a brief overview of the GENIE system will first be given, followed by a more in-depth look at the features of GENIE.

Overview

With GENIE, a teacher or group of teachers can create a library of subject area questions that can be subsequently reproduced for use as tests, reviews, or problem and assignment sheets, or can be directly used for individual evaluation or remedial drill. Although originally designed for Mathematics, GENIE is now used by the English, Social Studies, and Science Departments, and is steadily growing in number and variety of applications.

An important feature of GENIE is the ease with which the final product, a sheet of problems, can be produced, and flexibility permitted in the appearance of this sheet. The designer of the sheet has a great deal of control over the format of the sheet, including titling, inclusion of special instructions to the student, number and variety of copies produced, interrelation of problems, spacing of problems, and much more. All this is accomplished through a concise set of commands that are either punched or pencil-marked on cards.

A second major point is the organization allowed in the library of questions. As questions are authored and entered into a GENIE library, each is assigned a unique four-character identification code. Judicious use of these codes not only lends organization and structure to the contents of the library, but also facillitates the manner in which questions are later retrieved for a problem sheet.

The third important aspect of GENIE is its author language. This is a hybrid language designed especially for the actual needs of a problem-generator setting, with many features included in response to the requests of authoring teachers. A typical question can usually be written in under 15 minutes, and then entered into the library either directly by typing it at a terminal keyboard, or via punched tape. A full range of editting and testing functions is provided in the GENIE system so that the authoring teacher will have full and, if desired, personal control over the progress of his question from inception to final logging into a GENIE library.

A Problem Sheet

Although GENIE supports multi-disciplinary libraries, at present the most fully developed library is in high school Algebra. And so, this will be used to illustrate the operation and uses of GENIE.

A somewhat typical problem sheet (reduced from 14 7/8 x 11 inch size) is shown in Figure 1 on the next page. The questions are printed on the left half of the sheet, and the corresponding answers opposite them on the right. The spacing of questions and answers is such that, in the original full size, an 8 1/2 x 11 inch sheet will accomodate all of the question side, but will omit the answers (see dashed line). This is done so that a teacher may use the computer-printed sheets as masters in a reproduction process, fitting the questions onto standard size paper, but without disclosing the correct answers.

Notice that titles and page numbers are given above both the question and answer sides of the

PAGE 1-2-1 NAME: PER

'GENIE' ALGEBRA SAMPLER

SOLVE THE FOLLOWING OPEN SENTENCES

1 $17 - 20(-11X - 10) = -2$

2 $0 = 13(-5M + 10) + 11$

3 $2X^2 + 11X + 3 = 0$

4 $9/10(3 + 7B) = -2/3$

WRITE THE FOLLOWING EQUATIONS IN SLOPE-INTERCEPT FORM

5 WRITE THE EQUATION OF THE LINE THAT CONTAINS THE POINTS (-12,-8) AND (2,-18).

6 WRITE THE EQUATION OF THE PERPENDICULAR BISECTOR OF THE SEGMENT JOINING (4,-6) AND (-3,0)

SIMPLIFY THESE EXPRESSIONS

7 $-(76C + 45)$

8 $(5D + -2)(D + 1)(2D + 5)$

9 $\sum_{J=0}^{6} 2J + 3 =$

SET UP EQUATIONS AND SOLVE

10 FIND THE DISTANCE BETWEEN THE POINTS (0 , -4) AND (-6 , -12)

11 A LAUNDRY WORKER WAS NORMALLY ABLE TO FINISH HER DAILY QUOTA IN 7 HOURS ONE DAY SHE BROUGHT HER TEENAGE DAUGHTER ALONG TO HELP TOGETHER THEY WERE ABLE TO FINISH THE QUOTA IN 5 HOURS AT THAT RATE HOW LONG WOULD IT TAKE THE DAUGHTER TO DO THE JOB BY HERSELF ?

PAGE 1-2-1

'GENIE' ALGEBRA SAMPLER

1 S1AN

$X = -219/220$

2 S1AN

$M = 141/65$

3 S2IC

$\frac{-11 +- \sqrt{97}}{4}$

4 S1MN

$B = -101/189$

5 R1JE

$Y = (-5/7)X - 116/7$

6 R1JI

$Y = (7/6)X + (-43/12)$

7 P1AB

$-76C - 45$

8 P1WC

$10D^3 + 31D^2 + 11D + 10$

9 P3HB

63

10 A5HG

10

11 A5YB

35/2 HOURS

Figure 1

Sample GENIE problem sheet

sheet so that a teacher can re-match sheets that may have been cut in the reproduction process. (The wide-print header above the questions is a luxury permitted us by the model printer we have.) Notice also the directions appearing just before questions 1, 5, 7, and 10; these are independent of the questions themselves and can be worded to suit a teacher's individual preferences. Finally, the four-character sequences following the numbers on the answer side are the library reference codes that identify each question. Note that questions 1 and 2 have the same reference code, but are different problems. This is because of internal variations built into this question at the time it was authored. Nearly all of the questions in the Math library have a similar kind of built-in variety, so that even a small number of questions can produce a vast number of specific problems. This also means that if the same choice of reference codes were used to produce another sheet, the chances of seeing the same coefficients and answers on even one problem would be remotely small. (It is not unusual for heavily numeric questions to contain internal variations that run into the millions.) The result of this potential for variety is that a teacher can produce equivalent copies of exams for class use or can create drill sheets of unlimited length for student review.

Uses of a Problem Sheet

When GENIE was first made operational, in the summer of 1974, its first goal was to provide a means of producing testing materials only for the various Algebra courses taught in the Niles high schools. Its strong points were seen as (1) the ability to produce, on demand, valid questions from a wide spectrum of topics; (2) to produce similar, equivalent tests for make-up or individual work; and (3) to do this with a minimum of teacher effort. GENIE was totally successful in meeting this initial goal. A teacher's preparation for a test could now be directed more toward content and less toward worrying over problems that would properly 'work out' or about the actual writing of a neat and organized master copy. The equivalent copies of a test that were possible because of internal question variation guaranteed much improved in-class and between-class test security. And, perhaps most important of all, the teacher's use of the same source materials greatly improved the standardization of course content and evaluation throughout the Mathematics Department and encouraged better articulation with feeder schools.

But this was only the start of GENIE usage. During the 1974-75 school year, several teachers found GENIE so convenient and easy to use that they began to integrate GENIE-produced materials directly into their courses as homework and supplementary assignment materials. This prompted a project in the summer of 1975 in which GENIE Math questions were carefully organized and documented in a Math Catalog, containing nearly 1000 coded questions. A sample page from this Catalog is shown in Figure 2, on the next page. With this Catalog, teachers began to correlate assignments, reviews, and remedial work so that course textbooks were completely supplemented by GENIE-produced problem sheets.

When several teachers started to produce GENIE sheets as remedial work for individual students, it was only a small step to the next plateau of GENIE usage. This was the duplication of certain key pages of the Math Catalog and distribution of these pages to entire Math classes. Students in these classes were given a brief introduction to GENIE commands, and were then encouraged to design and request their own remedial work as they felt the need. This process quickly spread through all the Mathematics courses, to the extent that student requests alone now average over 150 per week.

Requesting a Problem Sheet

The fact that large numbers of students can successfully design and request their own problem sheets may suggest that GENIE commands are quite easy to use. This is, in fact, true. Once the library reference codes for a group of questions are known, there is very little else to learn before designing a problem sheet. There are basically only two kinds of GENIE problem sheet commands: problem retrieval commands, and sheet formatting commands. The sample list of commands shown in Figure 3, which would produce the first portion of the sheet shown in Figure 1, illustrates both of these types.

The problem retrieval commands (lines 3, 4, and 5 of Figure 3) are simply the library reference codes for the desired questions. A prefixed number, as the 2 in line 3, indicates the multiplicity of a particular question. With this, an entire drill sheet can be produced with a single command, like 25 S1AN. A shortened question code, like the S2I in line 4, permits GENIE to randomly select a question with a code matching these first three characters. Reference codes shortened to two or even one character allow greater range in the selection of library questions.

Although not shown in Figure 3, retrieval commands can limit the selection of the range of reference codes in various other ways and can also guard against the random selection of the same question code more than once.

The formatting commands used in Figure 3 (lines 1 and 2) illustrate a header command (line 1) which supplies the title to be printed at the top of each page, and an instruction command (line 2) which gives a phrase to be inserted at that point on the problem sheet. Only one

Objective Code	Statement of Objective
S1A	Solve a linear sentence with variables on one side only

Questions available: 17

Question Code	Example	Description
S1AA	Variety of forms, all equivalent to X + A = 0	A is chosen from the integers -200 thru 200; X is a random letter
S1AB	AX = 0 or 0 = AX	A is chosen from the integers -200 thru 200; X is a random letter
S1AC	AX = 1 or 1 = AX	A is chosen from the integers -200 thru 200; X is a random letter
S1AD	Variety of forms, all equivalent to A - X = 0	A is chosen from the integers -200 thru 200 (nonzero); X is a random letter
S1AE	Variety of forms, all equivalent to - (X ± A) = 0	A is chosen from the integers -200 thru 200 (nonzero); X is a random letter; +- is chosen randomly
S1AF	Variety of forms, all equivalent to - (A ± BX) = 0	A and B are chosen from the integers -200 thru 200 (B nonzero); X is a random letter; +- is chosen randomly
S1AG	Variety of forms, all equivalent to B (A + X) = 0	A and B are chosen from the integers -20 thru 20; X is a random letter

Figure 2

Sample page from Mathematics Catalog

```
1)   $H "'GENIE' ALGEBRA SAMPLER".
2)   $I "SOLVE THE FOLLOWING OPEN SENTENCES".
3)     2 S1AN.
4)        S2I.
5)        S1MN.
6)   $$.
```

Figure 3

Sample list of GENIE generator commands

header command can be given for any problem sheet, but instruction commands are usually inserted before each new type of question, as is done in the example in Figure 1.

Line 6 of Figure 3 is an end-of-list indicator that must be present at the end of each command list.

These commands are marked or punched, one to a card, and are then submitted for processing in a batch mode. (GENIE is not and does not pretend to be an interactive tool for computer-assisted instruction in the classical sense.) At Niles, student requests are handled through an open computer lab, so that teachers do not have to collect card decks or distribute the problem sheets produced, and teacher requests are processed via a special drop-box.

There are many other formatting commands and several additional options to the retrieval command, but that shown above is all that is needed by students in the design of a problem sheet for drill or review. Teachers who wish to exercise the greater control desired in the creation of a test or assignment sheet, will often use formatting commands to control question spacing, paging, and number or variety of copies. They may also modify the retrieval command so that certain problems are worth more than others or so that problems are interrelated, even though they may contain internal variations. While these flexibilities are important and demonstarte the full power of GENIE, it is the fundamental simplicity of design and use that makes GENIE the successful tool that it is.

<u>Authoring a Question</u>

As mentioned earlier, the GENIE author language is a hybrid combining the forms of several existing languages and features that are probably unique to GENIE. The author language is BASIC-like in the appearance of its computational statements, but adds several non-BASIC functions like LPF (least prime factor), RPN (random prime number), and GCD (greatest command divisor; 2 arguments). There is even SRF, which takes two arguments, considers them as numerator and denominator of a fraction, and prints the fraction in reduced form. These and other functions, plus a randomizing operator, ?, make internal variation very simple to implement.

The sample question listing shown on this page in Figure 4 is short and easy to follow for anyone with a background in BASIC or FORTRAN. Line 1 of this listing illustrates one of the unique features of the GENIE author language, the choice assignment statement. In this statement, the expression between # and OF is evaluated to determine which of the succeeding expressions will be assigned to the left-hand variable. Strings and numeric values can be assigned interchangeably to any variable.

The output statement of GENIE, illustrated by lines 3, 4, 5, and 6 of Figure 4, allows full control over the appearance of the printed question. Values to be printed, enclosed in the [and] symbols, can be formatted themselves or shown in a compressed format by default. With the exception of the use of special characters and half-line advances, the question can be made to appear eaxactly as it would if typed on

```
1)   C = # INT(1?5) OF 'INCHES' , 'CENTIMETERS' , 'METERS' , 'FEET'
2)   A = INT(1?23)
3)   'THE BISECTOR OF ANGLE A OF A PARALLELOGRAM ABCD'
4)   'BISECTS THE SIDE BC.  IF AB = [A] [C], FIND THE'
5)   'LENGTH OF BC.'
6)   A: 'BC = [2*A] [C]'
```

Figure 4

Sample listing of GENIE Mathematics library question a standard typewriter. The A: of line 6 is an indicator for the automatic positioning of the answer portion of the question.

Figure 5 below gives just two of the many ways in which the question listed in Figure 4 may come out on a problem sheet.

Finally, it should be noted that GENIE permits questions of various types to be included in its libraries. Thus, while the questions shown in these illustrations have all been computational in nature with the correct answers given, they could just as well have been multiple choice, fill-in, or essay. The answers may be supplied or omitted at the authoring teacher's discretion.

Experience has shown that teachers unfamiliar with GENIE and, in fact, new to programming, can become competent and comfortable in authoring questions in a couple of days.

A1GF SAMPLE

*****

THE BISECTOR OF ANGLE A OF A PARALLELOGRAM ABCD BISECTS THE SIDE BC IF AB = 10 CENTIMETERS, FIND THE LENGTH OF BC

BC = 20 CENTIMETERS

*****

A1GF SAMPLE

*****

THE BISECTOR OF ANGLE A OF A PARALLELOGRAM ABCD BISECTS THE SIDE BC IF AB = 11 INCHES, FIND THE LENGTH OF BC

BC = 22 INCHES

Figure 5

Sample problems produced by the question shown in Figure 4

Summary

As has been shown, GENIE is a flexible, yet simple-to-use, system for producing computer-generated problem sheets. It is useful in a wide variety of manners and in a wide variety of subject areas. While the illustrations in this article have been drawn from a Mathematical setting, teachers in the Niles high schools have developed libraries in Chemistry, Physics, English, and Social Studies as well.

The GENIE system offers this range of applications on a very small computer (the original design runs on a Digital Equipment Corp. PDP/8 with 24K of core). This makes it economical enough for individual school districts to develop and support their own question libraries, while having others to share from. This ability to exist in small computers, with all the advantages this suggests, means GENIE should find a comfortable home in many educational communities.

Note: A large portion of the Mathematics library for the Niles schools was developed with the aid of funds from the Illinois Office of Education, Gifted Children Section.

COMPUTER CLUB COMPLEMENTS MATHEMATICS CLASSES

Donald W. Moffat

THE BISHOP'S SCHOOLS
Post Office Box 1948
La Jolla, California
92038

ABSTRACT: The art of teaching has advanced with modern techniques but there will always be some students who have great difficulty learning certain mathematics concepts. It has been found that, just like some children will accept love from a puppy but not from a human, some students will accept certain principles only when demonstrated by a computer. Examples are given to show that a computer can get weak students to do their workshop drills, can impress certain rules and procedures on students, and can provide quantities of data when needed. Background conditions are given to assist readers in relating the problems described to their own situations, and comments for further discussion are offered.

You have all fought the problem of trying to explain a complex relationship, using nothing but a chalkboard. If the relationship is essentially a dynamic one the only way to show it on a static chalkboard is to make several drawings -- a time-consuming procedure.

And what about those students who cannot understand new material because they don't understand basics? They need drill, drill, and more drill in basics; they have had ample opportunity to work problems but they can never bring themselves to open the workbook.

We have found a synergistic arrangement -- our small Computer Club and our regular mathematics classes each provide something of benefit to the other. The result is greater than the sum of the parts. The arrangement is quite simple; Computer Club members sometimes visit one or more classes to describe a topic or to demonstrate an application -- other times an entire class or part of a class goes to the Computer Room to observe a demonstration or for a 'hands-on' session.

The original purpose was to generate interest in the Computer Club to maintain membership. As the club continued and its members, in learning to use the computer, improved their general math knowledge, they started helping other students. The bilateral benefit became a fact as interaction between the Computer Club and regular classes benefited both.

After setting the stage with some background, I'll give a few examples that will be of interest to everyone. Then there will be some suggestions for others to build on and, finally, there is a look at other possibilities.

BACKGROUND CONDITIONS

First, a description of the situation and conditions we are talking about, so you will be able to relate the problem and solution to your own situation. With this background you can determine the degree of similarity, and you can arrange whatever adaptations are best.

The Bishop's Schools are a fairly small pair of girls' and boys' independent schools, grades 7 through 12. Total enrollment is 400. A couple years ago we acquired a Monroe Model 1666 desk top programmable calculator, and immediately built a Computer Club around it. Club members first spent their time learning to use the machine in all its modes -- as a keyboard calculator, using a stored program entered from the keyboard, and using a stored program entered by way of punched cards. After they learned to use the machine, club members began seeking ways to put their abilities to use.

Our upper school offers, in addition to various arrangements and combinations of two years of algebra and one year of geometry, a year of advanced algebra and a year of calculus. It can be seen that the small size of our student body (which means a small senior class), puts a limit on the variety of electives which can be offered in the Math Department. Addition of a computer course would have strained registration in other classes to the point where our already small classes (many classes have five or six students, and twenty is the absolute maximum) could not be justified. Therefore, a Computer Club for those interested was the right answer for our situation.

WORKSHOP

The first opportunity for club members to put their abilities to use was a mathematics workshop in basics. There always seem to be some students who were absent the day their classes were taught to add, multiply, subtract, and divide. Those same individuals are not likely to be the ones who would take a book and work problem after problem until they learn the skill. That's one of the reasons why they are lacking in basics today. But working with a computer -- that's something else again! They have to sign up for a turn at the computer -- to have the computer give them problems and check their answers.

Say a student wants to practice addition of whole numbers. A club member loads the computer from a set of punched cards and then the computer and the student who needs help do all the rest themselves. The student first enters the upper and lower bounds of the numbers with which he wants to practice addition and then a built-in random number generator selects the first problem to be printed. The student punches her answer on the keyboard; if it's right the computer prints out the next random problem, if it's wrong an error message is printed and the student tries again. The random number generator can produce without recycling more problems than anybody would want in a lifetime.

There is a similar program for giving practice in subtraction, and it can be arranged to give problems with positive answers only, or to include problems in which a positive subtrahend is larger than a positive minuend. A wider range of practice can be obtained by using the addition program described, and enter negative lower bounds for numbers in the problems resulting in a mixture of both arithmetic and algebraic addition.

In the same way, students can ask club members to load a program for giving them multiplication and another for division problems. There are also programs for giving practice working with fractions.

Students work the simplest problems in their heads. For others they are encouraged to work the problem on paper and then enter the answer on the keyboard. In addition, after the machine prints out a problem the keyboard is available for use as a calculator; the student can work an entire problem, or one step, or a group of steps on the keyboard. This feature is important because the reason some of the students need this help is that they never memorized their multiplication tables completely and accurately and it is common for them to make the same mistake every time they try a certain problem. As mentioned, if the wrong answer is followed by the CONTINUE button, machine and student stay in a loop and never move on to the next problem.

The point is that some students, when opening a workbook and seeing pages and pages of problems have all they can do to force themselves to spend a bit of time on them. These are the students we especially want to help. A large number of that subgroup will find new interest in learning basics if a computer gives them problems and checks their answers.

PRECISION

One thing our classes learned from club members is that when a procedure is to be implemented literally, without benefit of common sense, it's necessary that the procedure be precise. It is so easy to get into the habit of sloughing through steps without really understanding what's correct and wrong. The human mind is correct a surprising percentage of the time when it grabs for something that "seems right"; when it's correct a given percentage of times its host passes the course.

After Computer club members did enough programming to appreciate the need for precise thinking, they talked to classes and entertained them with tales of their own frustrations in debugging a program. Students got quite involved in the explanations and were pleased when they could spot errors such as "forgot to reset the counter before beginning a new series," or some similar oversight. Some club members are quite good at leading the class to "find" the error.

For a related situation, consider that old familiar error of saying that the square root of $a^2 + b^2$ is a+b. This error persists without limit in spite of explanations, brainwashing, and low grades. Strangely, some students know that the square root of $3^2 + 4^2$ is 5, and they might even go through the steps correctly, but they so often follow the wrong procedure when using symbols or other numbers. But let a computer point out that one procedure is wrong and one is correct . . .!

Here's what happened. Under guidance of Computer Club members, some class members worked up a flow-chart which properly squared, added, and then took the square root, as shown in Figure 1. At the same time some students who had just (for the hundredth time) written 7 as the equivalent of the square root of $3^2 + 4^2$ were helped to prepare a flow-chart which would cause the computer to follow the same steps they had taken, as shown in Figure 2. The rediculousness of this process was immediately obvious to some students, and they learned their error without going any further.

Club members then wrote programs and came back at the next class meeting to explain how the program steps followed the steps in the flow charts. Everyone went to the Computer Room where the programs were loaded along with a framework program that directed the computer to accept a value of a from the keyboard, print it, accept a value of b from the keyboard, print it, print the answer by the process of Figure 1, print the answer by the process of Figure 2, and return to accept a new value for a. Students took turns entering a's and b's. A few entered just one or two pairs and left but by far most of them entered many, many pairs. Almost all took the printed answers along and apparently looked them over because several students made comments when we met

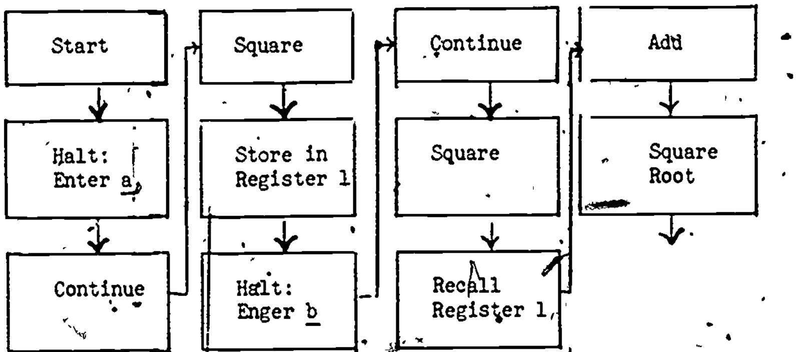


FIGURE 1. Flow chart for correct evaluation of square root of the sum of squares.

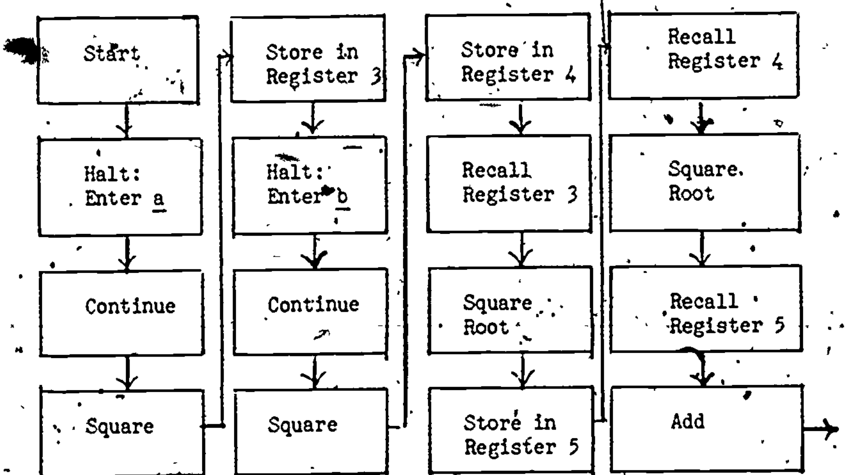


FIGURE 2. Flow chart of incorrect process that many students use.

again.

That same square root problem was buried in a routine test a month later and no one (not one) made the same old error.

Of course, the students could go through the same procedure manually. They did, many times. And they understood when the error was pointed out. Many had worked the expression both ways on a hand calculator and understood the difference. But the error persisted. It seems there is magic in having a computer show the difference between right and wrong.

It is interesting to note that most students would first enter numbers such as 3 and 4 for a and b. Then they would try pairs which were not necessarily Pathagorean numbers, such as 8 and 9, or 20 and 30. Finally they would enter numbers they would never try manually, such as 7628495, or 5.194736. Perhaps it was the computer's ability to handle these numbers without a pause that earned it the student's respect. Sometimes they entered the same pair several times, as if to see if the computer was just guessing and might not give the same answer repeatedly.

Admittedly the computer was just giving numerical examples -- not a rigorous proof. It is better for overall understanding and application if the item learned is understood from a basic proof. But the proof had been given and those who were inclined to understand did so; for the others the choice was learn from numerical examples or not at all.

Along the same lines, misunderstandings regarding signs were eliminated, leaving the only sign errors as those due to plain sloppiness. The background was the same -- students had been not quite sure of the procedure and were getting by (rather admirably) with a high degree of judgment.

The method was different in that only the correct procedure was flow-charted and programmed. Class members made up problems and, with the help of club members, prepared flow charts -- club members wrote and explained programs, allowing for keyboard entry of variables. A typical problem would be such as:

$$x = a+b - c + (d+e) - (f+g)+[h+i-j-(k+l-m)]$$

Some problems were simpler than this and some included fractions and exponents.

Students prepared lists of values to give the letters -- some positive and some negative, and then entered those values. We had some interesting sessions showing why the computer's answers were correct, but in the end the students believed and learned the rules and procedures. When the computer told them that 5 minus a negative 7 is 12, they were convinced.

And then there is parentheses and grouping. Club members wrote programs to demonstrate differences such as between ab^n and $(ab)^n$. Although they were simple programs, they benefited club members and class members alike. Others we looked at included $\frac{wxy^{-1}}{ab}$ and $a/b + c$.

FAMILIES OF CURVES

Some students had difficulty in grasping the total significance of a change in the constant in equations such as

$$y = k(\frac{1}{x})$$

$$\text{or } y = k \log x$$

the standard way has been to make up a table of sets of X and Y pairs, calculated while holding k at a certain value. Then, make up another table of pairs while holding k at another value. To do this for 5 or 6 values of k, and then plot the points and sketch the curves with a reasonable degree of accuracy on a large chalk graph, required that an entire class period be set aside. Or, the teacher could take the finished product to class and make a brief explanation of how it was produced. The disadvantages of either choice are obvious and, worse of all, a significant number of students still missed the concept.

But, once again, the computer worked its magic, and reached just about everyone, and it involved a very little deviation from the standard approach. First, the students heard a standard explanation of what we were doing and then they were shown how the computer would calculate X-Y pairs for given values of k. Next, a preplanned program was loaded and the computer was put to work printing out a series of tables. A sense switch allowed us to choose either to enter incremental values of k and x (and then the whole series of tables would be finished automatically), or we could enter a k followed by selected values of x until we sent the program back for another k. After the tables were printed, students plotted the points quickly, accompanied by nods of understanding from the others.

Then, using both table and graph, it was pure pleasure to explain how to work, "If y varies as the inverse of x, and if y is 12 when x is 25, what is x when y is 24?" Suddenly there was mass understanding of the fact that any particular pair of X-Y numbers is found at just one place on the graph, and the graph for only one value of k goes through that point.

Of course, they could have seen the same thing by calculating the tables themselves or by accepting tables the teacher or someone had calculated. They could have seen it, and many did, but when the computer talked, everyone listened.

GAMES

The first reason for introducing computer games to the students was to generate interest in the Computer Club. It worked well. But a more important benefit turned out to be that many students began to look forward to talks from club members on what new game they were developing. Usually the students were not interested to the point of wanting to learn to program such games, but they were fascinated by the idea of playing games with a machine. And while being fascinated, they happened to pick up a few bits of knowledge during those talks of what others had accomplished on the computer.

They also learned a lot about logic and reasoning. For example, one of the simplest games is, "Guess a number." In this game, one player enters a number and then the opponent makes guesses; with each guess the computer tells the guesser whether he is too high or too low. When the correct number is guessed, the computer tells how many guesses it took and then returns for the first player to enter a new number. The point is that students quickly learned to minimize their guesses by eliminating large blocks of numbers at a time. They also learned, when entering the number to be guessed, to try anticipating the strategy the other player would use, and to avoid numbers that would be the midpoint of a logical block of numbers.

We progressed through more complicated games, each demonstrating some particular point, and each teaching something as a side benefit.

Programming of tic tac toe was quite a challenge on this small computer and therefore everyone involved learned a considerable amount about strategies and about programming. Those who worked on the program became unbeatable whenever they played a conventional game of tic tac toe.

A very important topic learned from this game was the use of literals. Since our computer does not have matrix subscripting capability, we used powers of two to represent each of the nine locations. Therefore, any given sum of literals representing occupied spaces uniquely identified the combination of occupied spaces. Once they learned how to have the machine identify the spaces occupied by itself and by the human player, the club members were able to complete the program so that the computer would find, in its memory, the correct move for every situation.

It was a very simplistic approach, and the computer didn't learn a thing -- but the club members learned a considerable amount. Besides learning the difference between numbers as numerics and as literals, this game taught them how to think through a strategy. And our Philosophy teacher was interested in some discussions on thinking that resulted. Students got to seriously looking into the question of following specific instructions as opposed to following policy instructions. Many of them started questioning their own thought processes and there was a perceptible jump in the number of students who sought a basic understanding of what they were learning, rather than trying to memorize a set of steps to follow. The benefits of programming this game were noticed in all departments.

CONCLUSION.

Just the same as we see people of all ages say, "I wouldn't take that from him" when we know they would "take it" from another person, we see students opening their minds to an explanation when they see that a computer has accepted it and operates according to it.

There is a bilateral benefit to having Computer Club members help out in regular classes. It is clear that regular classes learned from these interactions. In addition, many good questions were asked during club presentations, and the club members learned to say that valuable phrase, "I don't know but I'll find out and let you know." They learned to say it and they found out and reported back. Each group triggered something in the other group. Of course, the Computer Club members benefited the most because they were generally members of both groups.

Our big limitation was the computer's capability. If it worked in a simpler language, most students could have learned to write their own programs and could have gained considerably more out of the entire exercise. The relatively small amount of storage limited the types of programs, especially games, that could be used. On the other hand, limited capability causes the programmers to examine alternative procedures and to learn to optimize programs. It also leads to some discussions and investigations on what could be done with additional capability.

Regarding the workshop programs, more capability would definitely be of benefit in allowing for diagnosis and instruction subroutines. It would also become possible to keep count of a student's errors and to work some statistics on improvement rate.

A school that has an X-Y plotter should find it interesting to let students see points as they are being graphed at the same time they are being calculated. The significance of each step might reach some because of its immediacy. However, the number watching this live on-line action would have to be limited, and there might not be a net gain when considering the time lost by having to rerun the graphing repeatedly for small groups. Having a student or teacher do the graphs on an overhead projector or a large chart for an entire class did not seem to have enough drawbacks to say that the lack of an X-Y plotter was a serious problem.

Improvements can always be made, even without adding capability. In the workshop programs mentioned earlier, perhaps an "I give up" routine would be worthwhile -- provision for the student to admit he's reduced to guessing so that the computer can give the correct answer and get on with the next problem. Or perhaps it could branch automatically into this routine after a specified number of errors are made on any one problem.

One pitfall to watch out for when members of the Computer Club are explaining a program to a regular class; there is a tendency for a student's interest to manifest itself in a barrage of questions that, on examination, go into inappropriate amounts of detail. Club members are glad to answer such questions but nothing is gained if the answers do not really lead to a clearer understanding of the method.

Even though there are no doubt other improvements that could be made, the idea of using a Computer Club to complement regular math classes seems to yield the benefits expected, plus several side benefits.

TEACHING THREE-DIMENSIONAL DESIGN PRINCIPLES USING LOW COST COMPUTER AUTOMATED DISPLAY TECHNIQUES

Art Chapman, Assistant Professor
Phil Winninghoff, Assistant Professor

School of Architecture and Environmental Design
California Polytechnic State University, San Luis Obispo, CA 93407

ABSTRACT: With the advent of low cost graphical display devices, it is now possible to purchase at reasonable cost (under $20,000) sufficient computer power and peripheral devices for the generation and manipulation of 3-D images. Using a stand alone system of mini-computer, digitizer and cassette auxiliary storage devices, environmental design students experience form generation in a more comprehensive and expeditious manner than is possible using conventional studio methods. A variety of computerized form generation techniques are discussed.

One of our responsibilities as instructors of environmental design, which covers five degree programs (Architecture, Architectural Engineering, Landscape Architecture, City Planning and Construction Engineering), is instruction in basic principles of three-dimensional projection and form generation. These principles are directly interspersed in several of our introductory courses and subsequently applied in many of the advanced courses. At the introductory level, a great deal of emphasis is placed upon students acquiring a proficiency in spatial perception. This instruction is of critical importance in environmental design education since most building design or urban design problems are associated with the definition, shaping, grouping and description of three-dimensional forms.

The approach used to deliver three-dimensional projection instruction is to start by examining two-dimensional views of solid objects, and to have the student describe the objects three-dimensionally by the use of conventional isometric and perspective hand drawings. The student then finds out if his perception of the object is correct by submitting his solution to an instructor who evaluates the student's work. Subsequently, the student is provided with an opportunity to compare his work with that produced by his colleagues as well as a correct solution obtained from the instructor. Unfortunately, this procedure is subjected to a time delay of usually four or five days before the student receives the desired instructional reinforcement from the instructor's evaluation. It is most important that this time delay be reduced to a minimum. If the instructional reinforcement could be received by the students quickly, the impact would enhance the three-dimensional visualization learning process. To accomplish this end by placing the responsibility of instant feedback on the instructor is limiting in that it requires the course material to be restricted to a defined set of problems, all of which have master answer sheets. However, in order to give the course some degree of flexibility in content, it is desirable to make changes in the problems each school term.

Another negative feature of this process is that when the student mechanically describes a three-dimensional form it is usually with only one view of that form and by this it is assumed he has mastered a visualization of the object. It may be the case that much of the form hidden from view or only partially in view is not understood and quite possibly the student chose a particular view because it was the only one he could draw accurately.

An effective solution we have found to the problems is the utilization of an electro-mechanical drawing device composed of a digitizer connected directly to a programmable calculator which is in turn connected to a digital plotter. This configuration is capable of converting two-dimensional drawings into three-dimensional images. Students then check their work directly against the solutions provided by the device. It is the nature of this type of electronic system developed that it requires a person to trace the appropriate lines in the two-dimensional views before a three-dimensional image can be obtained, which is a similar process to the conventional mechanical technique where we require the student to redraw elevation views to familiarize himself with the object. As the student traces a two-dimensional line, its three-dimensional counterpart is produced. Accordingly, the student recognizes instantly whether the correct line is being traced as the three-dimensional image emerges. If the student has any innate ability to think three-dimensionally, this reinforcement will improve his ability considerably. It also quickly exhibits visualization deficiencies.

After investigating many manufacturers of programmable calculator equipment desired, we found at the time (mid-1974) that the Hewlett-Packard would be most cost effective and satisfy our needs for single station use. The items purchased were as follows:

ITEM 1. 9830A Calculator:

This item was chosen because it used a standard programming language and subsequently provided the students an opportunity to learn a standard programming language as well as make use of the graphic capabilities of the device. This capability also allows the equipment to span into other areas of our instructional program when not being used for the application described in this proposal.

ITEM 2. 9866A Printer:

The programmable calculator required pre-programming for various applications. This programming was not extensive because of the simplicity of the programming language. Adequate programs for this application can be written in one day. It was essential that a printer be provided with the calculator to obtain listings of programs as they were developed.

ITEM 3. 9864A Digitizer:

This equipment serves as the input device for coordinate information.

ITEM 4. 9862A Plotter:

This plotter was chosen for this project for the following reasons:

1. It provides hard copies which are required for review by the instructor.

2. It interfaces with the 9830A calculator with the maximum simplicity.

3. It does not require special paper.

ITEM 5. Option 272 - Read Only Memory:

This provides the interface between the various components.

This entire package was under $20,000. Approximately 300 students have required projects on the system throughout the academic year, plus there is use by a great many students who have the option of using this equipment for three-dimensional projects in advanced courses. Attachment 1 provides a brief users' manual for the basic perspective generating system.

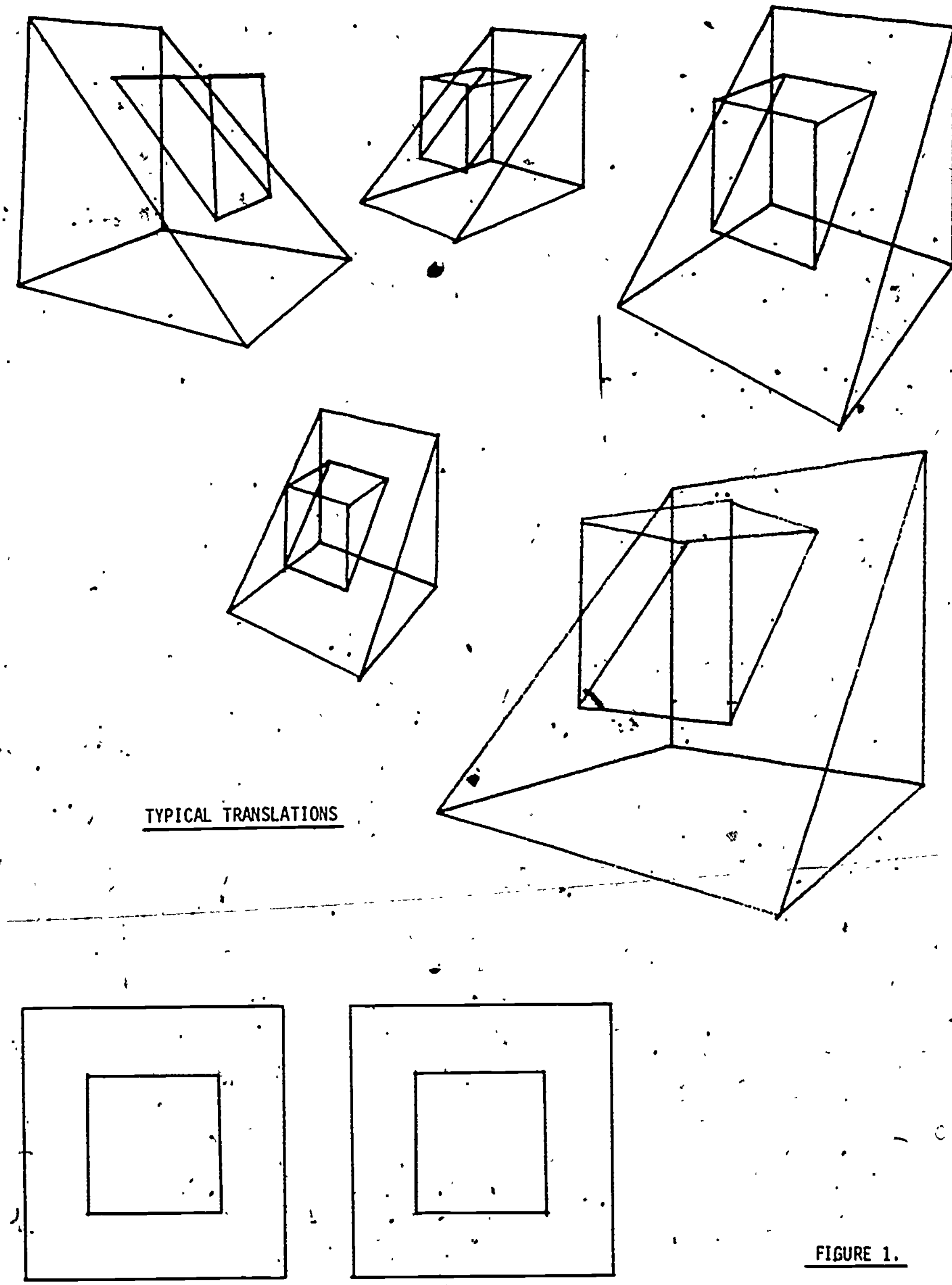
Figure 1 is an example of a typical plan-elevation to three-dimensional translation performed by a beginning student.

More advanced uses include the making of animation movies (see Figure 2). This is accomplished by picking a series of view points that can sequence a viewer through a scene. Photographs are then taken of the individual frames, then the photographic sequence becomes the movie.

A third major use is in basic design investigations. A common introductory problem is the concept of form generation based on abstract unit growth which requires that repetitive primary units be interrelated in such a way that they establish logical patterns determined by sets of established rules. Growth concepts are tested on the system by a repetitive design program which allows the user to define a primary unit--any three-dimensional form--and attach another primary unit to any edge, surface or point on the original unit and continue to repeat this attach process. As this procedure takes place, the entire object can be rotated with the perspective program thus achieving rapid form generation and visualization.

Two deficiencies can be noted in the system. First, because of the number of students involved, the single work station concept is confining. We are in the process of expanding. Second, when it is necessary to make several plots, either to achieve the proper angle of view desired or to group objects into scenes and produce an animation sequence; there is a short waiting period for a number of drawings to be generated. This plotting problem could be eliminated, at a considerable expense, using a refresh type cathode ray display, which is capable of dynamically regenerating views, instead of the plotter. But this eliminates the hard copy output which is necessary in a student environment.

This system has generated a good deal of student enthusiasm. They gain insights about perspective which can only be acquired through more tedious exercises. Plus, they have the possibility of producing form animation and growth generation drawings that would be impractical to produce in any other manner.



TYPICAL TRANSLATIONS

PLAN

ELEVATION

FIGURE 1.

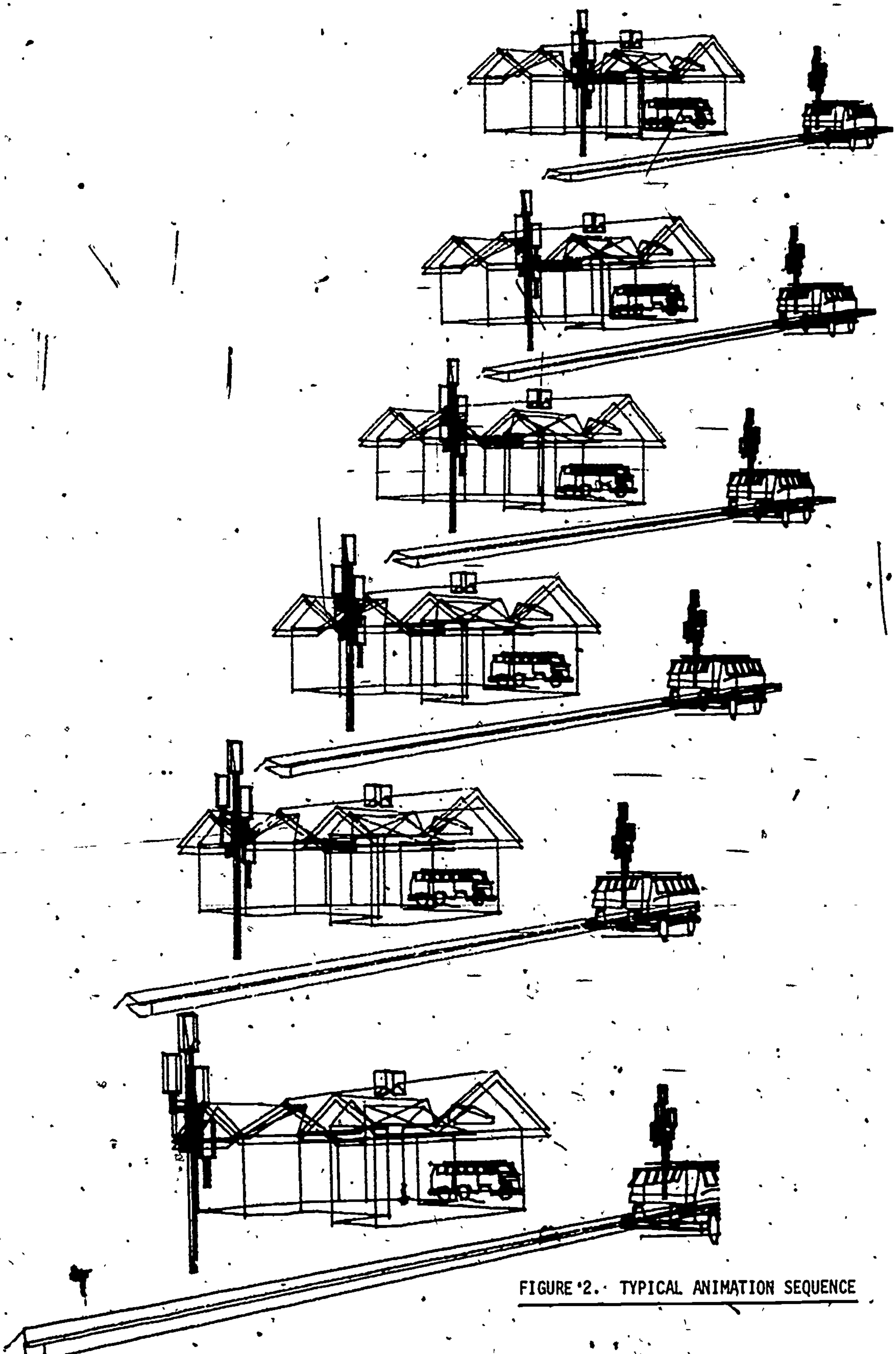


FIGURE 2. TYPICAL ANIMATION SEQUENCE

ATTACHMENT 1.

ABBREVIATED PERSPECTIVE SYSTEM MANUAL

INTRODUCTION

This manual is intended for use by students and faculty in the operation of the perspective system. The system consists of a Hewlett-Packard 9830A calculator, a 9862A plotter, a 9864A digitizer, a perspective system cassette and data file cassettes.

Each data file tape is capable of storing 45 separate objects with up to 250 lines each. As each object is stored on the data tape the user is asked ~~for a~~ protect code which will assure that no subsequent user can copy over that file. When a file is copied over, all of the previous information is replaced by what is currently in the data portion of the calculator's memory. Copying is accomplished with a STORE command from the digitizer's menu.

OPERATION

The function of the perspective system is to allow users to trace plans and elevations on the digitizer, and have the lines they trace translated into perspectives on the plotter. The first step in the procedure is to align the plan drawing in the proper orientation for the view that is desired and to place the elevation on the ground line. The plan drawing must lie entirely within the range of the two lines radiating from the eye station.

The following figure illustrates a standard chart that consists of two sheets and is taped to the surface of the digitizer. The drawings should be placed under the standard chart to give the digitizer cursor a flat, obstruction-free surface to be moved on. The cursor must remain close to the digitizer surface or it will lose its reference position and ~~a beep will~~ sound. If this occurs, the cursor's position can be easily reestablished by placing it at point "0" on the chart and pressing button "0."

START-UP PROCEDURE

1. Turn on the calculator, plotter and digitizer.

2. Insert the perspective system cassette.

3. Press "LOAD"; after ├ press "REWIND."

4. Remove the system cassette and insert the data tape.

5. Press "RUN" followed by "EXECUTE."

The calculator should respond with "SET 0 AT ORIGIN AND EXECUTE" shown on the display panel. Next the cross-hairs of the cursor should be moved to the intersection of the picture plane and the center of vision on the digitizing surface. Press button "0." If it is not already on, the red light on the cursor will light up. The next step is to press "EXECUTE" on the calculator keyboard. This should cause the message "BEGIN DIGITIZING AFTER THE BEEP" to appear on the calculator display. The screen will go blank when the calculator is waiting for an entry from the digitizer.

OPERATION

The user decides upon what operation he wants the system to perform. Commands are given through the digitizer cursor from the plan or elevation for drawing operations, or from a menu at the lower right corner of the chart. Some commands (e.g., rotation, picture plane movement and height of viewer) are entered through the calculator keyboard (with the exception of the STOP command). These commands are asked for by a message displayed on the display screen.

RECOVERY

If at any time the user wishes to interrupt the system, he can do so by pressing "STOP." The procedure for continuing the operation at a place where the system will ask for further instructions is "CONTINUE 100" followed by "EXECUTE."

INSERTING DATA TAPES

It is not necessary to turn the calculator off to insert a different data tape. The procedure is to place the cursor on the bottom box of the menu and press "S" (a labeled button on the digitizer cursor). After the tape has rewound, exchange it with the other tape and press "RUN EXECUTE."

DRAWING PROCEDURE

First, be sure that paper is on the plotter bed and the chart hold button is down. If the drawing procedure is begun directly after the calculator displays "BEGIN DIGITIZING AFTER THE BEEP," no picture plane movement or eye height information will have been entered. In this case, the calculator will establish the picture plane five inches away from the viewer and his eye, three inches above the ground at the scale of the chart. The user has the option of starting the drawing by placing the cursor on either the plan or elevation. If he chooses the elevation, nothing will happen at the plotter, but the red light on the cursor will blink when he presses the "S" button on the digitizer. This tells him that he has made an entry to the calculator which tells it the relationship of the elevation of the point to the ground. If he then takes the cursor to the plan view and presses the "S" button on the digitizer, the plotter arm will move the pen to the proper place on the plotter bed and set the pen down. If he moves the cursor to another point in plan and presses "S" a line will be drawn in perspective parallel to the ground at a distance established when "S" was set at the elevation. If "S" has not been initially set at the elevation, the line will be drawn on the ground.

To draw vertical lines the cursor is taken to the elevation where the height is reestablished and then back to the same point in plan. To draw sloped lines the cursor is brought back to a different point in plan after the height has been reestablished.

To go to a new point without drawing a line, the cursor is taken to the "PEN UP" box on the menu and "S" is pressed. This action lifts the pen. It will remain raised until it has been repositioned to a new point on the drawing, at which time it will go back down.

If the user makes a mistake, and draws a line he does not want, he can move the cursor to the "DELETE LINE" box on the menu and press "S." The pen will raise and be repositioned at the start of the deleted line. This line will be removed from the data file.

STORING DATA FILES

At any time in the drawing operation the data created by the user can be stored on the data tape. This is done by setting the cursor over the "STORE" box on the menu and pressing "S." The calculator will ask for the file number which is to be keyed in at the keyboard. If the file is open, that is, if its protect code is zero, the calculator will ask for a new protect code and store the data. Protect codes are any four-digit numbers. If the file has already been assigned a protect code, the user must be able to key it in when asked (before the calculator will store data in that file). Storing data into a file completely erases what was previously there and replaces it with everything currently in the data portion of the calculator's memory.

LOADING DATA FILES

The "LOAD" instruction is used to bring data that has been previously stored back into the calculator's memory. Before new drawing data can be added to the data set, it must be redrawn or repositioned as discussed in the following sections.

SUMMARY OF COMMANDS USING THE DIGITIZER AND THE MENU

PEN UP - Raises pen

DELETE LINE - Removes the last line drawn from the data set and repositions the pen to the start of the deleted line. If the last command was a PEN UP command, "S" on the digitizer should be pressed twice.

SCRATCH - Erases the data set.

REDRAW - Asks for view information and redraws the data set currently in the calculator's memory.

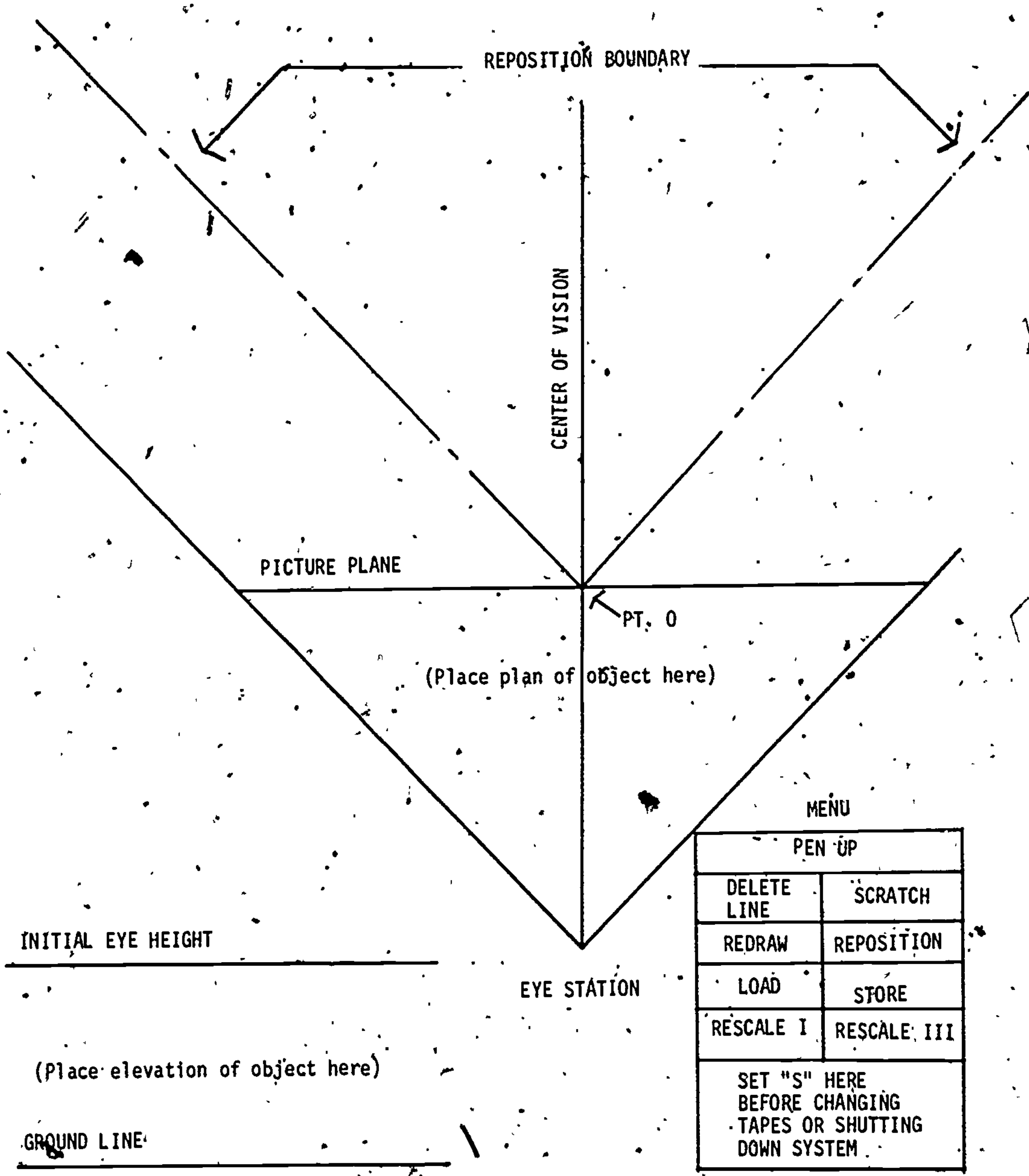
REPOSITION - Asks for positional, as well as view, information and redraws the data set currently in the calculator's memory.

LOAD - Brings a data set from the data tape into the calculator's memory.

STORE - Saves the data set currently in the calculator's memory on the data tape.

RESCALE I - Rescales all the values in the data set currently in memory by the specified scale factor.

RESCALE III - Rescales all the values in memory by a different specified scale factor for the width, depth and height of the object.



STANDARD PERSPECTIVE DRAWING CHART