ED 127 986 IR 003 954

AUTHOR Winiecki, Roger D.
TITLE A Computer-Mediated Instruction System, Applied to Its Own Operating System and Peripheral Equipment.
PUB DATE Aug 76
NOTE 10p.; Paper presented at the Association for the Development of Computer-Based Instructional Systems Summer Conference (Minneapolis, Minnesota, August 10-12, 1976)

EDRS PRICE MF-$0.83 HC-$1.67 Plus Postage.
DESCRIPTORS Autoinstructional Aids; *Computer Assisted Instruction; *Computer Science Education; Higher Education; Individualized Instruction; *Instructional Systems; *Instructional Technology; On Line Systems; Time Sharing
IDENTIFIERS ADCIS 76; BASIC; Resource Sharing Timesharing System

ABSTRACT

Each semester students in the School of Health Sciences of Hunter College learn how to use a computer, how a computer system operates, and how peripheral equipment can be used. To overcome inadequate computer center services and equipment, programed subject matter and accompanying reference material were developed. The instructional system has a supervisor program, a student record file, and up to twenty instructional programs each consisting of up to nine levels of instruction. Direct use of the error handling facilities and the system (SYS) functions are provided by the BASIC-PLUS language and Resource Sharing Timesharing System. (CH)

A Computer-Mediated Instruction System, Applied to its Own Operating System and Peripheral Equipment

Roger D. Winiecki

School of Health Sciences, Hunter College
City University of New York

Problem

Each semester, about 40 to 60 students in the undergraduate and graduate programs of the School of Health Sciences of Hunter College need to begin to learn how to use a computer*, its operating system and peripheral equipment.

They have a wide spectrum of abilities, self-confidence, motivation, and need-to-know. They discover their needs at various times of the semester. Their class and work schedules allow them to use the computer only at widely various times of the day and week. They perceive their needs variously and so want to begin by studying various facets of the computer system. They have heard and read about computers, but would rather do things than be told about them. When they go looking for jobs, they like it to be said of them that they know how to use all the peripheral equipment on the computer --- and their prospective employers like it, too.

Resources

The computer center staff is minimal, the manuals inadequate, the physical quarters cramped, and the computer frequently overloaded.

Solution Components and their Objectives

The solution has three components:

1) a system of standardized computer programs and record file designed to facilitate computer-mediated instruction
2) instructional subject matter embedded in those programs
3) a reference manual

The objectives of the instructional system, apart from its subject matter, are as follows:

a) allow instructor or student to set individualized goals
b) facilitate speedy access by initially naive students to instructional programs
c) monitor and record student progress toward goals
d) allow student some control over the sequence of instruction
e) allow student to restart a task from where he stopped in a prior session
f) allow author use of the BASIC-PLUS language and RSTS system functions with only minimal restrictions, to facilitate subject-matter programming as described below

*A Digital Equipment Corporation PDP-11/20 with a wide variety of terminals and other peripheral equipment, supervised by DEC's RSTS V4A (Resource Sharing Timesharing System)

A common objective of the programmed instructional subject matter and of the manual is to help users of a particular computer learn how to use its operating system and peripheral equipment.

The programmed subject matter is designed to perform these functions:

a) recommend to the student appropriate alternatives when the instruction he has requested is either unavailable due to equipment being busy or impractical because of his lack of prerequisites
b) refer the student to a section of the manual appropriate to the instruction requested
c) involve the student in active hands-on demonstration of the capabilities and limitations of the computer's operating system and peripheral equipment
d) facilitate student experimentation with sofware and selected hardware, to
   i) develop the habit of and techniques for learning by direct observation
   ii) develop realistic confidence in the ability of the operating system to handle his errors without damage to other than possibly his own program and of the hardware to withstand certain operator errors without failure
e) tailor the manual's procedures according to existing conditions, in order to avoid unnecessarily frustrating the student and other users
f) monitor performance of procedures, fault-isolate and report errors to the student, and recommend corrective procedures
g) command the system to record completion of intermediate levels of instruction

The manual contains, for each major subject of instruction,

a) a statement of capabilities and limitations
b) a guide to locations and functions of controls
c) step-by-step instructions for use
d) suggested activities for user familiarization
e) a list of further references

System

The system consists of a supervisor program, a student record file, and up to twenty instructional programs, called tasks. (See Figure 1.) When a student logs on and runs the supervisor program, the computer asks the student for identification. If it finds no record of this student's prior use of the system, it records his identification and asks if he would like to set his own goals. If yes, the computer stores the goals supplied. Otherwise it stores a standard set of goals. (If the instruction has been assigned as part of a course, the instructor will have already entered the student's identification and appropriate goals.) It also places in core common (a small scratchpad preserved from overlay during chaining) a pointer to this student's record, so that he will not have to identify himself each time one program chains another. Having identified the student, it then shows him his achievement goal and his current achievement (or restart) level for each task, and asks which task he wants to pursue next. If the task chosen is valid (exists), the supervisor program chains that task.

Each task consists of up to nine units or levels of instruction embedded in about 1K words of bookkeeping and errorhandling overhead which is the same for all tasks. The levels are to be done sequentially. A student who has completed one or more levels in a prior session is started on the next level. When the student either aborts the task (by use of CTRL/C) or completes the level corresponding to his goal for that task, the supervisor program is chained to allow him to choose another task.

System Applied

Any person who asks to begin learning how to use the computer is handed a manual and advised to begin at the beginning. The first section of the manual shows the locations of terminals, and tells how to find a free one and how to log on and run $HUNT. $HUNT is the supervisor program for the instructional system which in this application is called the Treasure Hunt. There are two reasons for this name:

1) It sounds more appealing than "obstacle course", which it also resembles.
2) It resembles a treasure hunt. It rewards attainment of each intermediate goal by supplying a clue. The student must use the clues from all assigned goals to derive a "combination" to be set into the "lock" on the "treasure chest" --- that is, the switches on the front panel of the CPU. When the system senses that the chest has thus been "cracked", it "spills treasure" through the plotter. (See Figure 5.)

Importance of Errorhandling Facilities and System Functions

In the following three examples of subject matter programming considerations the importance of the instructional programmer's access to errorhandling facilities and operating-system special functions (SYS functions), unobstructed by an "author language", becomes clear.

Example #1:

In order to proceed to a different task, the student must determine whether the device required by that task is available. This is a potential inconvenience to students in two situations: First, five of the terminals available to students for participation in the hunt are located in a room separate from the computer and its other peripheral equipment. Only experienced users might know a method for determining the availability of a device in another room. Hunt participants cannot be expected to know such a method. Second, even though a device may appear on visual inspection to be available, it may actually be logically assigned to a particular user's job.

Therefore, in most cases the task programs check availability of any peripheral required and advise the student appropriately. These checks are made possible by author access to the errorhandling facilities of the BASIC-PLUS language. A routine used for such an initial availability check can then be used within the task to monitor device status as the student manipulates the device.

For instance, the task dealing with DECtapes checks each of the two to eight DECtape drives for various conditions. If the task finds an off-line drive, it advises the student to proceed using that drive.

If no off-line drive is found, it advises the student to either investigate the possibility of making a drive available or proceed to another task. Within the task, the same routine that performed the initial availability check is used repeatedly to monitor the status of all the DECtape drives in order to make sure the student has mounted the proper tape on the proper drive with the proper setting of its switches.

Example #2:
The student of any computerized instructional system must be expected to make errors. In most systems these errors are made in typing answers to questions posed by the system, which compares each student answer to anticipated answers in order to decide whether the student's answer is "correct". This mode of instruction is used frequently throughout the Treasure Hunt.

But in some cases the answers to such questions may depend on which terminal the student is using or on the current status of the computer's operating system and peripherals, and a wide variety of answers might be acceptable. In such a case it is generally simplest for the author (and most instructive for the student) to just try out the answer or have the student try out the answer and process any resultant errors with errorhandling routines.

In cases such as the DECtape task mentioned in Example #1, in which the student's response consists of manipulation of hardware, answer-matching must take on a character quite different from that of typed-response matching, and the instructional programmer's direct access to errorhandling facilities is vital even to verify a correct response.

In addition to giving the instructional system a flavor of "awareness" of its current environment (with minimum maintenance by its author), this approach allows the student to make errors and immediately see that the computer and its operating system are not damaged by those errors. Hopefully, this contributes to student willingness to learn by experimentation.

Example #3:
To abort any running program, the properly trained user types "C" while holding down the CONTROL key, the combination known as CTRL/C. The author has frequently noted users who, even after several months of regular use of the computer, still try to perform CTRL/C by pressing both keys simultaneously. As might be expected, only about 50% of their attempts are effective.

Therefore the task dealing with common features of terminals requires the student to execute CTRL/C repeatedly, until five consecutive successes have been logged. The task program can log each success without abortion because it uses a SYS function to trap CTRL/C and then handles it as a recoverable error. It uses another SYS function to snatch single characters as they are typed rather than waiting for the student to press the RETURN key. In this "single-character submode" it detects the character "C" when the student fails to use the CONTROL key properly, and prompts him to hold the CONTROL key down while typing "C".

Programming for such system resilience and monitoring of student manipulation of hardware would be extremely difficult if not impossible without direct use of the errorhandling facilities and SYS functions provided by the BASIC-PLUS language and RSTS operating system. For this reason the small amount of overhead in each task and the brief delays in chaining are easily tolerated.

System Limitations and Suitability for Other Applications

In evaluating this system (apart from its subject matter) for possible use in other applications, the following factors should be considered:

1) Environment and installation procedure
   a) The system programs run on a DEC PDP-11/20 mini-computer with a 28K core memory under the RSTS V4A operating system, using DEC's BASIC-PLUS language and RECORD I/O.
   b) The supervisor program should be compiled, protection-coded as a privileged program, and stored toward the beginning of the system library in order to minimize directory search time during chaining.
   c) The record file and task programs may be stored in a private account, but should be entered at the beginning of the account's directory as in b) above.
2) Structure
   a) The number of tasks is limited to 20. The sequence of tasks is determined by the student.
   b) The number of levels in a task is limited to 9. The sequence of instruction within a task is serial by level.
3) Records
   a) Several students may use this system simultaneously without endangering each others' records. (This is made possible by the UPDATE option of DEC's RECORD I/O.)
   b) Records regarding a student's assignment and progress are each limited to one number (0-9) for each task. No provision is made for recording a degree of mastery of any particular level.
   c) No provision is made for storage of unanticipated answers or other student feedback.
   d) The system can easily be made to certify completion of the student's assignment, as well as retaining the record for later use.
   e) An instructor can easily change a student's assignment without affecting his progress record.
   f) If a computer-generated problem is to be solved or continued other than during that run of the task in which it is generated some portion of the student's name or ID number may be used as an argument of the function used to generate the problem. This saves record space and allows some variation in the problem from student to student while always "remembering" what problem was presented to a particular student. (In the Treasure Hunt the clues are generated and remembered in this way.)
4) Restrictions on the author
   a) Certain variables are reserved for system overhead.
   b) CTRL/C is trapped to enable the student to quickly return to the supervisor program without providing identification again.

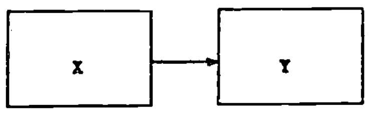
c) Core common is partially occupied by student identification for use by chained programs.

5) Possibilities to try when the subject matter overflows the structure
   a) Divide the subject matter into separate hunts with different names.
   b) Expand an individual task or tasks, retaining serial organization:
      i) Organize the subject matter as a quiz with remediation. Program the quiz as a task, and chain other programs for remediation. (See Figure 2.)
      ii) Divide the subject matter among as many programs as necessary, with each program containing an integral number of tasks plus the usual record-keeping overhead. (See Figure 3.) (Note that, for faster chaining to restart a student at an an advanced level, the first in the series chains directly to the program containing the desired restart level.)
   c) Use an additional supervisor program or programs in place of one or more of the original tasks, allowing up to 20 parallel sub-tasks (each with up to 9 levels) in place of each replaced task. (See Figure 4.)

<u>Evaluation</u>

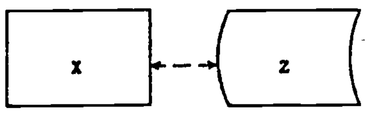
The system and its instructional subject matter are to be evaluated during fall semester of 1976 on the basis of (1) comments written by students during the hunt and (2) their replies to a brief questionnaire after completion of the hunt. Revision, addition, and further evaluation will continue.

Notes to Figures 1-4:

1) 

means program X chains program Y

(To "chain" means to "initiate from one program loading and execution of another program in place of the first".)

2) 

means program X has read/write access to storage Z

3) "Per" means "similar in format and function to".

4) ".REC" denotes a student record storage file.

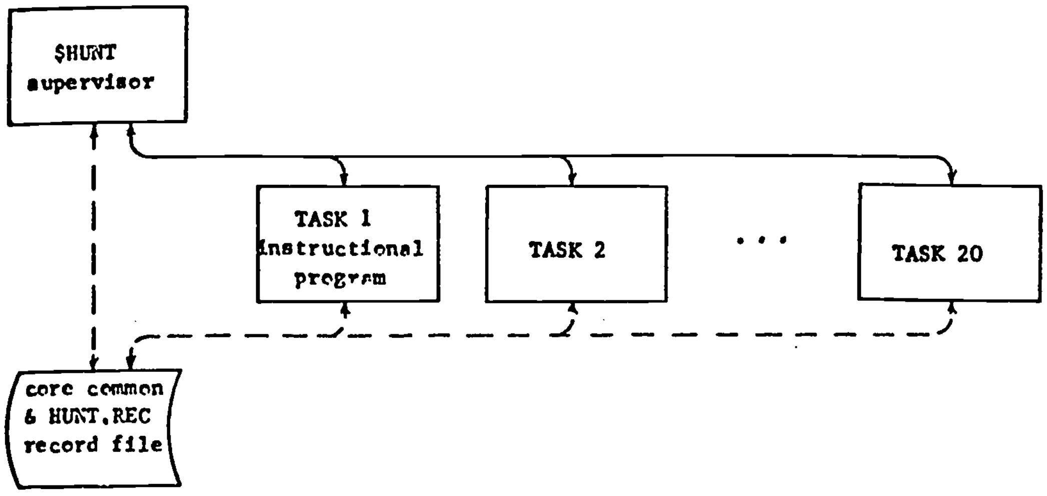


Figure 1. Standard Structure of System

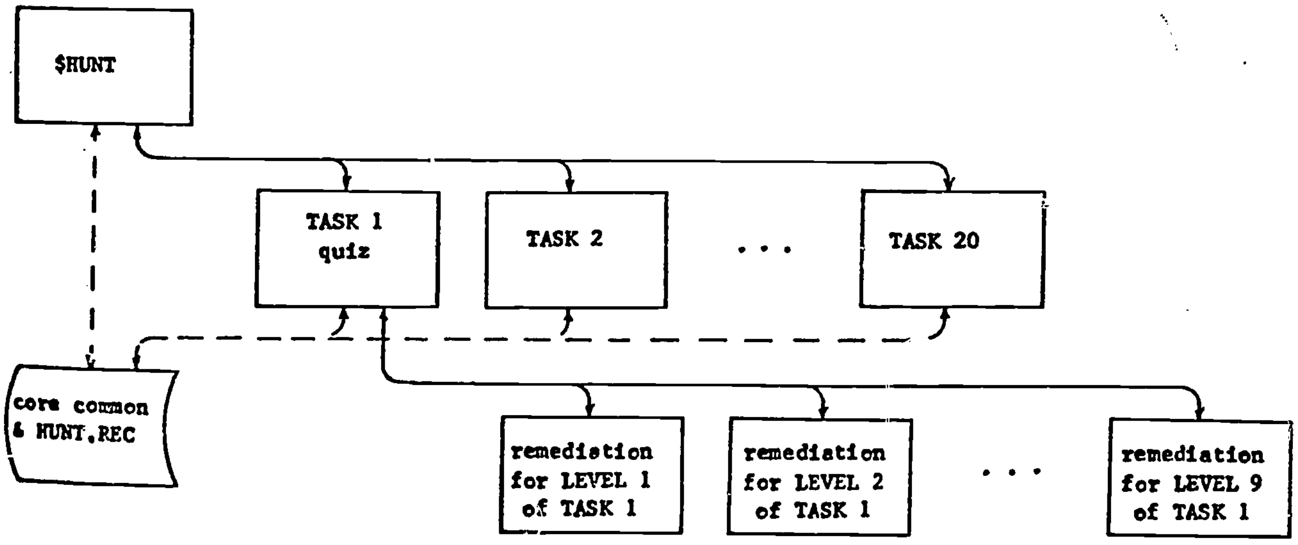


Figure 2. TASK 1 Expanded for Quiz with Remediation

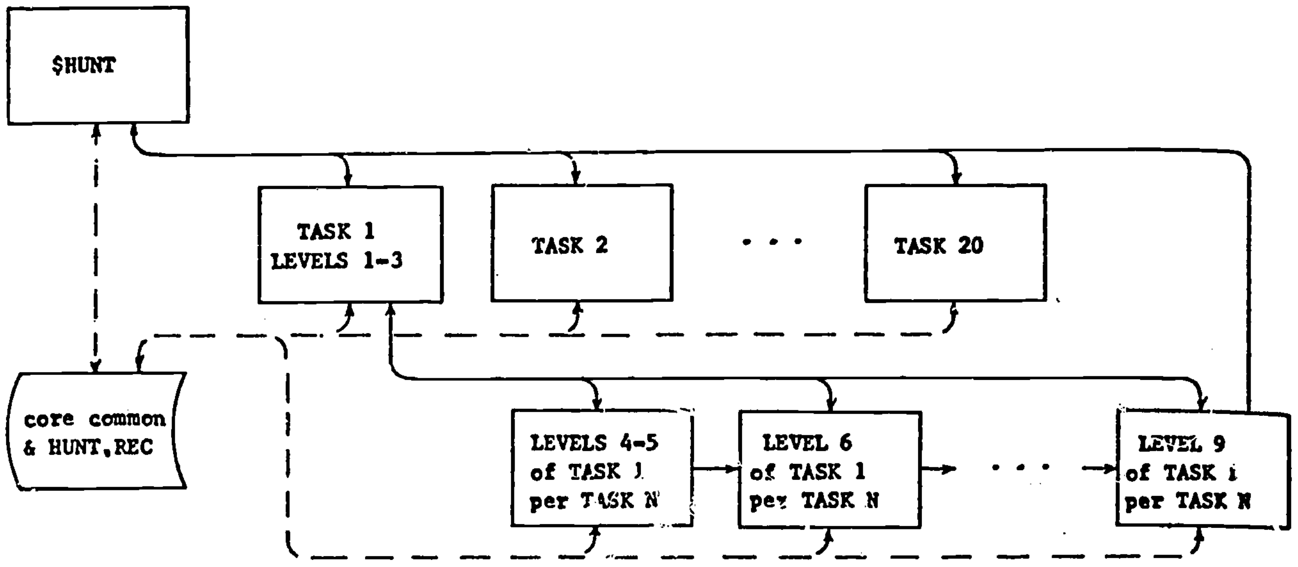

$HUNT
TASK 1 LEVELS 1-3
TASK 2
. . .
TASK 20
core common & HUNT.REC
LEVELS 4-5 of TASK 1 per TASK N
LEVEL 6 of TASK 1 per TASK N
. . .
LEVEL 9 of TASK 1 per TASK N

Figure 3. TASK 1 Expanded for Large Sequential Levels

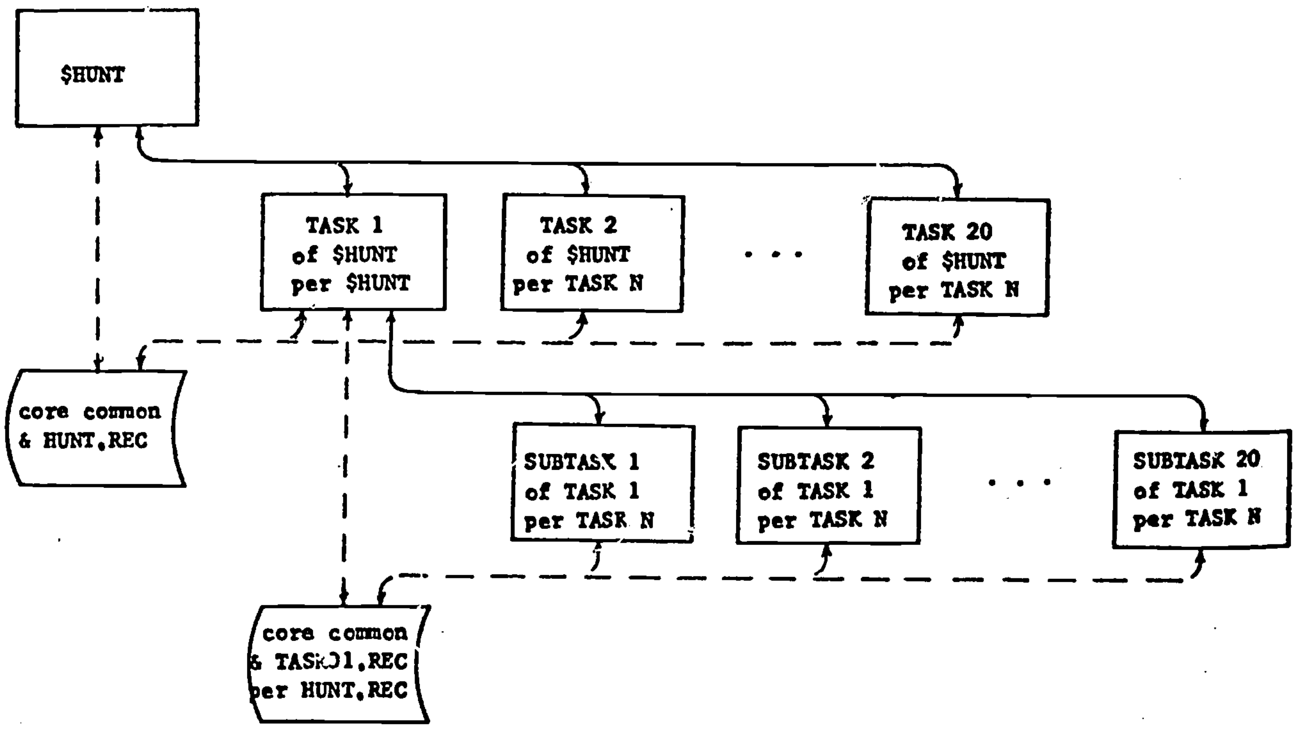

$HUNT
TASK 1 of $HUNT per $HUNT
TASK 2 of $HUNT per TASK N
. . .
TASK 20 of $HUNT per TASK N
core common & HUNT.REC
SUBTASK 1 of TASK 1 per TASK N
SUBTASK 2 of TASK 1 per TASK N
. . .
SUBTASK 20 of TASK 1 per TASK N
core common & TASK01.REC per HUNT.REC

Figure 4. TASK 1 Expanded for Parallel Subtasks

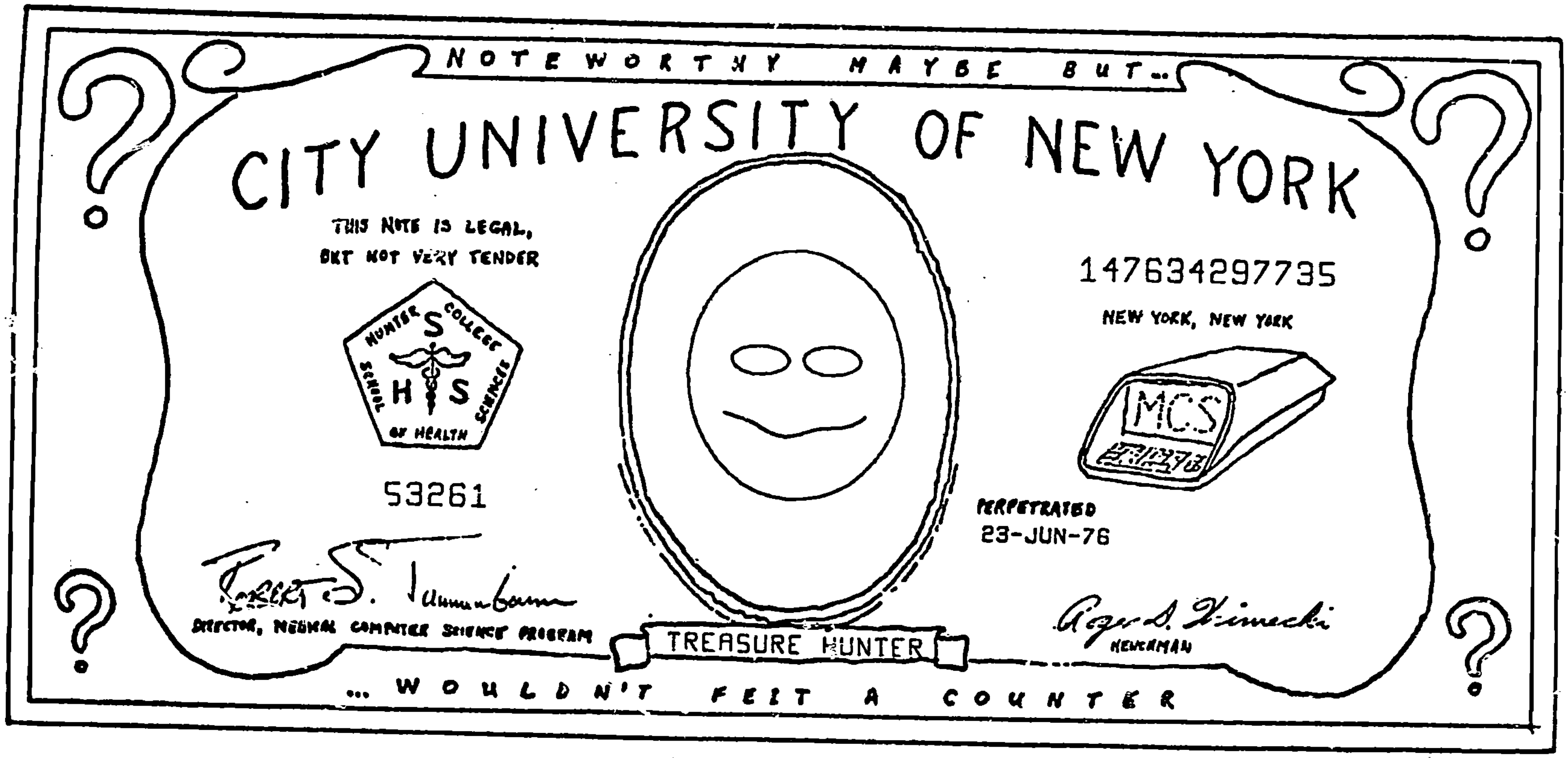

NOTEWORTHY MAYBE BUT...
CITY UNIVERSITY OF NEW YORK
THIS NOTE IS LEGAL, BUT NOT VERY TENDER
HUNTER COLLEGE SCHOOL OF HEALTH SCIENCES
HSS
53261
147634297735
NEW YORK, NEW YORK
MCS
PERPETRATED 23-JUN-76
DIRECTOR, MEDICAL COMPUTER SCIENCE PROGRAM
TREASURE HUNTER
... WOULDN'T FEIT A COUNTER

Figure 5. "Treasure"

Portions of Treasure Added by Plotter:

number set into CPU switches

caricature

student name

date

goals achieved