SESSION VI COMPUTER-ASSISTED INSTRUCTION: A SYMPOSIUM

JONATHAN VAUGHAN, *Hamilton College, President*

A multimethod generative CAI system for remedial mathematics

JOHN W. COTTON, JOHN P. GALLAGHER, MARC HOPKINS,
and SANDRA P. MARSHALL

*Graduate School of Education, University of California, Santa Barbara
Santa Barbara, California 93106*

A computer-assisted instruction system for remedial mathematics has been written to yield four teaching methods based on two orthogonal variables: (1) With ad hoc frame orientation (AFO), training modules have a fixed order for all students, strict performance criteria must be met before moving to new modules, and students can ask only for problem solutions. With information structure orientation (ISO), the first two AFO conditions are dropped and students can also request partial answers and hints. (2) A confluent (affective) mode contrasts with didactic instruction by adding use of given names, self-report of moods, relaxation procedures, some communication with another student, and so on.

This report describes a computer-assisted instruction (CAI) system that has been developed to permit four modes of instruction in remedial arithmetic and algebra. These modes result from orthogonal variation of two binary dimensions.

One dimension is a version of information structure oriented (ISO) instruction vs. a version of what Carbonell (Note 1) called ad hoc frame oriented (AFO) instruction.¹ AFO instruction may be considered traditional CAI, with a great deal of control of lessons exercised by the instructor or curriculum developer.

This report is based on work done under Contract N00123-77-C-0087 from the U. S. Navy Personnel Research and Development Center in San Diego, California. Appreciation is expressed to Lynne Cantlay for consultation in the development of the confluent procedure, to Seth Breidbart for programming and advice in the development of the current LISP interpreter, to J. Dexter Fletcher and John Wolfe for their guidance while serving as contract monitors, and to Lynn Brock for providing a taped version of LISP from which the present interpreter could be developed. An indirect, but very important stimulus for this research was the first author's participation in the Concept Attainment and Rule Induction section of the 1973 Information Processing Psychology Workshops sponsored by the Mathematical Social Sciences Board and held at Carnegie-Mellon University. Special thanks go to the section directors, Lee W. Gregg and Herbert A. Simon, and to a third host, Allen Newell, all of whom provided intellectual stimulation and personal assistance strengthening our commitment to interactive computing in psychology.

ISO instruction has its origins in the field of computer science, where artificial intelligence specialists such as Carbonell (Note 1) and J. S. Brown and Burton (1975) have made important innovations. They have developed systems that permit intelligent dialogue between computer and student, as well as allowing a good deal of student initiative in selection of topics to be learned and methods of learning. Barr, Beard, and Atkinson (1975) performed the first experimental comparison of AFO and ISO systems and used them to teach students the BASIC programming language. The present report describes a working CAI system that can produce either AFO or ISO instruction, thus permitting further experimentation to evaluate the relative effectiveness of the two methods.

The second dimension manipulated is affective vs. didactic instruction. G.I. Brown (1971, 1975) has called for the explicit merging of cognitive and affective emphases in instruction, terming this merger "confluent education."

In the present project, it is hypothesized that holding somewhat informal conversations between computer and student may help the student to develop efficient learning strategies, possibly by preventing fixation on inappropriate activity. Similarly, encouraging the student to express his or her feelings about mathematics and about computer-assisted instruction may be beneficial. The other side of this instructional dimension, didactic teaching, is defined by default; it is instruction

focused upon explicit intellectual tasks to be mastered, without emphasis upon emotional factors.

GENERAL DESIGN AND IMPLEMENTATION OF INSTRUCTIONAL SYSTEM

The CAI system developed in the past year can provide on-line instruction for computer terminal use with topics such as negative numbers, factors and prime numbers, inequalities, simple linear equations, and fractions. Its structure bears some similarity to that of Uttal's generative CAI system for analytic geometry (Uttal, Rogers, Hieronymous, & Pasich, 1970).

The system is programmed in a modified version of Harvard's UNIX LISP (partly obtained from the Navy Personnel Research and Development Center, San Diego, and partly developed by one of the present authors), having an expanded list of hardwired functions plus LINEREAD, SLEEP, ASCII, CHECKF (like the University of California, Irvine, LISP 1.6 LOOKUP), and SUPERG (an improved garbage collection procedure). It runs under the UNIX operating system on the UCSB Computer Systems Laboratory and Computer Center PDP 11/45 computers.

The problem-generating, responding, and feedback cycle is as follows: Instructional items are generated by programs relatively specific to the module in which they are contained. For example, a fraction problem is generated by the fractions module. Numerical fraction problems may be of Level 1 (Term Operator Term), Level 2 (Term Operator (Term Operator Term)) or ((Term Operator Term) Operator Term), or Level 3 ((Term Operator Term) Operator (Term Operator Term)). Here a term may be taken informally to mean a quantity such as $2/3$ or $1-1/2$. An operator is one of the four basic arithmetic operators for addition (+), subtraction (-), multiplication (X), and division (/). In addition to the numerical problems, two other kinds of problems exist for most modules: word and story problems.

Word problems are almost identical to numerical or algebraic problems, but use more words to ask the questions. For example, a Level 1 fractions numerical problem might ask:

$$1/3 \times 3/4 =$$

(with $1/3$ being the first term, $\times$ being the operator, and $3/4$ being the second term), whereas the corresponding word problem might be:

What is the product of $1/3$ and $3/4$?

Story problems involve real-world quantities and events. A story problem for fractions might say, "The

gas gauge on your car indicates that the tank is $1/4$ full. If a full tank holds 20 gal, how many gallons are in the tank?" (This distinction between word and story problems is unique to us. The reader is warned that other authors, such as Gilkey and Koffman, 1974, use "word problems" to label what we call "story problems.")

Once the structure of a level of numerical problem or type of word or story problem has been established, it is possible to generate quasirandom instantiations of members of that class of problem. The top-level monitor and curriculum driver for the system specify what type of problem is to be presented and the problem generator then generates the specific problem to be presented. Customarily, the student uses scratch paper as necessary in developing an answer and then types that answer into the computer via a Lear ADM-3 or a Perkin-Elmer Data Systems 1100 terminal keyboard. The monitor routes this response to an English interpreter to be sure it is an answer rather than a request for some form of help, translates it into list notation, and refers it to the solution analyzer. The solution analyzer codes the responses as correct, incorrect, or partially correct, and returns to the monitor, which arranges for appropriate feedback to the student and storage of the coded response. Depending somewhat upon the specific teaching method to be employed, the monitor and curriculum driver next check the student history, decide whether a member of the same or a new class of items is to be presented next, and have the problem generator generate an appropriate new item.

The basic cycle may be interrupted or elaborated by certain kinds of student requests: In all teaching conditions, students may voluntarily end the session early, request a general set of help instructions, request the answer to the current problem, or request display of the process of solving the current problem. In the ISO conditions, students may also request a new module or submodule, request the answer to a calculation that may be of help in the larger problem at hand, or request hints (the number of available hints being limited by the characteristics of the problem) as to how to proceed with the current problem. Data on requests for new modules or submodules and on requests for hints are stored in the student histories. Once the special request has been satisfied, instruction continues in a natural sequence, except that early log out leads to nothing new until the next session.

Unless stated otherwise, this article describes the system as it operates under the didactic condition when paired with ISO.

DESCRIPTION OF CURRICULUM

The curriculum of the USCB CAI system is intended

Table 1
Description of Curriculum Modules, with Sample Problems

Level	Sample Problem	Restrictions
A. Negative Numbers Module		
1A	$-2 \times 3 =$	Any operator. Positive and negative integers from 1 to 9.
1B	$-(2 + -3) =$	Any operator. Positive and negative integers from 1 to 9.
2A	$(-8 + 2) \times 7 =$	Any operator (except division within parentheses). Positive and negative integers from 1 to 9.
2B	$-3 \times (6 - 5) =$	Any operator (except division within parentheses). Positive and negative integers from 1 to 9.
Word	Is the product of a negative integer and a positive integer positive or negative?	
Story	On Monday, Andy had a bank balance of \$52. On Tuesday he wrote a check for \$23. On Wednesday he wrote a check for \$36. What was his balance after writing the two checks?	
B. Factors and Prime Numbers Module		
1	What are two factors of 24?	Numbers from 2 to 50.
2A	The set of all factors of 36 is:	Selected numbers with at least three factors.
2B	What are the prime factors of 36?	Any nonprime number under 50.
Word	The complete factorization of a certain number is $\{3 \ 2 \ 2 \ 1\}$. What is the number?	Any prime number under 50.
Story	A 6 by 8 rectangle is covered with squares which are all the same size. What is the largest size square that can be used?	Select numbers from 6 to 50 which have a common factor other than 1.
C. Decimals Module		
1	$\$2.43 + \$0.28 =$	Up to two digits before decimal point. Two digits after decimal point. Addition or subtraction. Dollar sign present or absent.
2	$4.21 \times 7.2 =$	Up to two digits before decimal point of multiplicand or multiplier. Up to two digits after decimal point of multiplicand or multiplier. Values from 2 to 9.
Word	Find the equivalent fraction. Type the value of its numerator. $.23 = \frac{\quad}{100}$.	Numerator from .01 to 1.99.
Story	Steel rods measuring 7.03, 8.25, and 6.5 in. were cut from a rod 25 in. long. How many inches were left after the three pieces of rod were cut off?	
D. Absolute Values Module		
1	$ -41 =$	Positive or negative integers from 1 to 99 in absolute value.
2	$ 24 - (-35) =$	Pairs of positive or negative integers from 1 to 99 in absolute value. Intervening operator either + or -.
3	$ -12 - -22 =$	Pairs of positive and negative integers from 1 to 99 in absolute value. Intervening operator either + or -.
Word	If the absolute value of a number is greater than the number itself, is the number itself negative or positive?	
Story	No story problems.	
E. Fractions Module		
1	$1/3 + 1/2 =$	Addition, subtraction, multiplication, or division. Terms are either proper fractions with denominators from 2 to 9 and numerators 0 to 9 or mixed fractions with an integer from 1 to 3 plus a proper fraction of the kind just stated. Terms may be negative or positive. Same as for Level 1.
2A	$1 - 1/2 \times (1/4 - 2/3) =$	Same as for Level 1.
2B	$(1/2 + 1 - 1/5) \times 2/7 =$	Same as for Level 1.
3	$(1/2 + 1/4) \times (2/3 - 1/5) =$	Same as for Level 1.
Word	Which is not equal to the others? $1/6$, $4/5$, or $4/24$?	All three quantities between 0 and 1 in value. One of the three is the reduced form of one other. The third has a different value.

Table 1 Continued

Level	Sample Problem	Restrictions
Story	A turtle traveling from San Francisco to Los Angeles travels at the rate of $3/4$ mph. How many miles will it go in $1-1/2$ h?	
F. Exponents Module		
1	$3 ** 2 =$	Base 1 to 5. Exponent 0 to 6.
2	$2 ** (2 - 1) =$	Base 1 to 5. Each digit in exponent from -2 to $+2$. Zeros not displayed. Base 1 to 4. Zeros not displayed.
3A	$(3 ** 2) ** -1 =$	Base 1 to 4. Exponents -2 to $+2$.
3B	$(2/3) ** 3 =$ (Do not perform the division until all exponentiation is complete.)	Multiplication or division of the base numbers, which must each be an integer between -2 and $+2$.
Word	The third power of 3 is 27. What is the third power of 2?	
Story	A certain perfect cube has its height, width, and depth equal to 4 in. How many perfect 1-in. cubes will fit inside it?	
G. Square Roots Module		
1	$25 ** (1/2)$	All exact squares from 1 to 225. Also all such squares divided by 100 or multiplied by 100.
2	$250 ** (1/2)$	Numbers in the range from .01 to 22,500. No more than two places after the decimal point for any problem. Answers rounded to the first digit after the decimal.
3	$25 ** (3/2)$	All exact squares from 1 to 225.
Word	The square root of 16 is 4. What is the square root of 36?	
Story	A cube holds 27 perfect 1-in. cubes. How tall is the cube?	
H. Inequalities Module		
1A	Please choose a or b below: (a) $-2 > -3$ (b) $-2 < -3$	Inequalities in opposite directions in a and b. The same two distinct numerical values in a and b, each from -1 to -9 or 1 to 9 .
2A	If $A > B$, which is larger?	Either direction of inequality. Letters different but otherwise arbitrary. "Larger" or "smaller."
2B	If $A > B$, is $A + 4 < 4 + B$?	Same digit on each side of inequality. Different letters on each side of inequality. Same letters for the two inequalities. Either direction for each inequality.
2C	If $A < C$, is $-2A > -2C$?	Same positive or negative multiplier for each side of second inequality. Either direction for each inequality.
3	If $A > C$ and if $C > B$, is $A < B$?	First two inequalities of same direction (arbitrarily chosen) and third inequality unrestricted in direction.
Word	If X plus 2 is greater than 6, name about as small a value of X as you can think of.	
Story	If Bill is older than Jim and Jim is older than Ned, is Ned older than Bill?	
I. Simple Linear Equations Module		
1	$5 + X = 10$. Solve for X.	Two not necessarily different integers, each from 1 to 10. Operator may be $+$, $-$, $\times$, or $/$.
2A	$2X + 5 = 8$. Solve for X.	Three not necessarily different integers, each from 1 to 10. Operator may be $+$, $-$, $\times$, or $/$. No squared terms permitted. A unit coefficient will not be shown.
2B	$5a - 3a = 6$. Solve for a.	One integer from 2 to 9. Operators may be $+$, $-$, $\times$, or $/$. A unit coefficient will not be shown.
3A	$2a \times b = c$. Find a.	
3B	$2a + 5 = c$. Find a.	Two not necessarily different integers, each from 1 to 10. Operator may be $+$, $-$, $\times$, or $/$. A unit coefficient will not be shown.
Word	If two times a number is increased by three, the result is four less than the number. What is the number?	
Story	Joe's coin collection contains only nickels and dimes. The total amount of money in his collection is \$2.50. How many nickels does he have?	

Table 1 Continued

Level	Sample Problem	Restrictions
J. Polynomials Module		
1	If $Z = 3X + 2XY$ with $X = 1$ and $Y = 2$, what is the value of Z ?	Two terms on right-hand side of equation. Coefficients from 1 to 5; may be either positive or negative. XY may be replaced by a squared term, X^2 .
2	There are two factors of $9X^2 + 5X - 2$. Type one of them. (After getting one correct:) Now type the other factor.	Three terms on left-hand side of equation for quadratic equations. Four terms allowed if expressions involving X , Y , and XY (but not X^2 or Y^2) are used. Integer solutions only; no solution outside the range from -4 to $+4$.
3A	If $9X^2 - 6X + 1 = 0$, find two possibly identical factors of the left-hand side, but do not type them. Then type a solution from one of those factors. (After getting it correct:) If there is another distinct solution, type it. Otherwise, type none.	Middle term on left may be missing. Integer solutions only; none outside the range -4 to $+4$.
3B	If $12 - 4X + 3Y - XY = 0$, find the solution for X . (After getting it correct:) Find the solution for Y . No word or story problems.	Integer solutions only; none outside the range -4 to $+4$.

Note—Modules A, B, E, H, and I are used in the experiment about to be conducted.

to include higher level arithmetic processes and lower level algebra. Because one item whose substance is in our curriculum

$$(9 - X = 4. \text{ Solve for } X.)$$

has been rated as low as Grade 1.5 by Suppes, Goldberg, Kanz, Searle, and Stauffer (Note 2, pp. 10-26) and nonetheless is comparable to items in a UCSB remedial mathematics course, no attempt has been made to label items by grade level. Rather, items within modules are grouped by levels based on apparent complexity, and the 10 projected modules of the system are ordered for AFO groups on the basis of including inferred prerequisites of later modules in earlier modules. Table 1 summarizes all projected modules, with existing modules identified at the bottom.

Problems within a module are normally generated at random, subject to being at the desired level (Level 1A and Level 1B are considered variants on the same level; similarly for other letters attached to level labels). Typically, students begin with Level 1 of a module and work with it exclusively. Then they receive Level 2 problems intermixed with word and story problems. Finally, students receive Level 3 problems intermixed with word and story problems of the same difficulty as before.

The flavor of the curriculum is further conveyed by Table 2, which shows four of the five different basic word problems for the fractions module, together with alternate wordings, and 4 of the 10 basic story problems of that module. Possible variations in numerical entries are not shown, but random generation of such numbers is employed in setting up calculations to perform.

DETAILS OF ISO SYSTEM DESIGN

Since most features of the CAI system are present under the ISO condition, a detailed description of ISO system operation is presented first. Figure 1 summarizes this operation, which will be described more fully as this section continues.

The first phase to operate is the initial stage on the computer for a given student. As soon as a potential student is identified, he or she is assigned randomly to one of four experimental conditions and a computer file is created for that student with a program called MAKFIL. Initially, this consists of the student's name,

Table 2
Sample Word and Story Problems for the Fractions Module

- W1. Which one is not equal to the others: $1/6$, $4/5$, or $4/24$?
- W2. Which is *larger* (smaller): $1/4$ or $2/3$?
- W3. Convert $3/2$ to a *mixed number*:
(an improper fraction) (its reduced form)
- W4. What is the *product* of: $-2/3$ and $-5/6$?
(difference between) (sum of)
- S1. *Isabel* worked for $2-1/4$ h on Friday and $2/3$ h on Saturday. How many hours did she work?
- S2. A *motorcycle* travels 60 mph. How many miles will it travel in $3/4$ h?
- S3. *Rachel* had a piece of wire $3-2/3$ ft long. She cut $5/6$ ft from the piece to use in hanging a picture. How many feet of wire were left?
- S4. On a hike *Stanley* walked $1-1/6$ miles the first hour. At this rate, how many miles can he walk in $2-3/5$ h?

Note—*Italicized terms are variables which change when new items are generated, subject to internal consistency requirements. Parenthesized terms are substantive alternatives to italicized terms.*

Figure 1. Flowchart of ISO system operation in UCSB computer-assisted instruction unit. (Drawing modelled after Figure 2 of Uttal et al., 1970.)

teaching condition, and date of birth (to be used in the confluent condition). During the first computer session (the session after the pretest), the student is taught how to type appropriate messages on the computer terminal and how to use the special aids provided within the teaching condition. This lesson also includes a description of the way course material will be presented and advice on how to learn it. Special informal conversation is generated in the confluent condition, even though the system does not permit true understanding of natural language emitted by a student. Session 1 closes with some direct instruction in mathematics.

External Bookkeeping, SIGNIN, and SIGNOUT

MONITOR is an executive program that is involved in most aspects of the system and is active in these phases as well as the actual training sessions of the experiment. SIGNIN is a program for letting the student begin a session. SIGNOUT is a program for letting the student end a session or having the computer end it for the student. At the end of each attempted problem, MONITOR asks the student if another problem is desired. The student activates MONITOR by typing "(HELLO)."

The English Interpreter

The English interpreter handles a moderate variety of student input, first changing input from the terminal

into list notation where required, as in the case of responses in words or requests for the computer to perform a preliminary calculation with the COMPUTE routine. Then it checks for alternate forms for basic input words, accepting "y" and "n" and other variants of "yes" and "no" in place of those words, for example. The interpreter also has a misspelling routine that accepts alternate spellings of words such as "answer" and "compute." Then a key-word check is performed in order to recognize student calls for routines such as COMPUTE and to route such calls to the appropriate utilities.

Answers to mathematical questions are sent to the solution analyzer for scoring and feedback. Individual match programs exist in the solution analyzer for each mathematics module. In most modules an equivalence program exists for the purpose of finding alternate acceptable answers or answers that are basically correct but need minor improvement. Answers to conversational questions in the confluent condition are evaluated to some degree on the basis of key-word matching. The results of this matching somewhat affect the direction of subsequent conversation.

The Item, Response, Feedback Cycle

When it is time to present items, MONITOR calls CONTROL to decide whether to continue with an earlier category of items or to begin a new module or submodule. CONTROL checks the student file and

current day's data to see the status of the student's progress. In the ISO condition after Day 1, the student is asked by MONITOR whether he or she wants to continue with the previous day's module. The student can stay on a chosen module throughout the experiment if he or she wishes. However, it may be that a criterion of a certain number of successive correct responses on a module has been reached, a criterion deemed to represent substantial overlearning. In that case, CONTROL arranges for the student to be told that he or she has apparently mastered that module and that a move to new material is worth considering.

Once the current module and level of items to be presented is selected, MONITOR loads that module, and the curriculum generator generates an appropriate problem, presents it on the terminal screen, and calculates its answer. The student either answers the question and receives feedback (right or wrong) or calls a utility program (ANSWER, SOLVE, HELP, HINT, COMPUTE, LEVEL, NEW MODULE, or QUIT), to be discussed more fully in the next section of this report. If an answer is given by the student, the computer records the item and response category, making it possible to know if the criterion for offering to change the module has been reached.

A response is stored as correct, correct except for needed simplification, or incorrect. If simplification is needed, the student is told the type of error and asked if he or she wants to correct the original answer. Scoring of the correction is recorded as an answer to a new problem. After an incorrect response, the student is asked some variation of, "Do you want to try this problem again?" Therefore, several answers may be offered by the student before moving on to a new problem.

If the criterion has been met, the student is asked about the possible change to new material. If a different module is selected, its program is loaded and the problem generator for that module is called. Otherwise, the current problem generator is called again, and the item, response, feedback cycle is repeated.

Before each new cycle begins, there is a check to see if the scheduled time for the session has been exhausted. If so, then a TIMER program automatically signals to the student that he or she should sign out. The session's data are stored at this time. SIGNIN and SIGNOUT are both student activities, with the customary log-in and log-out processes reserved for the experimenter. One reason for having the experimenter log in and log out is to reduce the complexity of student activities. A second reason is that it is desirable to keep a computer terminal in operation continuously while running several student sessions. It is necessary to store data and call new files between sessions, but unnecessary to log out for one student and log in for the next.

Student-Activated Routines (Utility Calls)

ANSWER routine. The student is normally told only

whether a given answer is right or wrong. However, either before or after responding to a problem, the student can type "answer" and the correct answer will be displayed. If the request is made before an attempted correct answer, the response is recorded as "Give up" and MONITOR arranges for another item to be presented in the usual fashion. If the request is made after an incorrect answer, "Give up" is recorded as a further response following an error and MONITOR arranges for another item to be presented.

COMPUTE routine. This routine permits partial solution of a problem before the student offers a full answer. Suppose a problem is:

$$3 \times (1/4 + 2/3) = .$$

The student can say,

$$\text{compute } 1/4 + 2/3,$$

and watch the terminal screen for the resulting partial answer prior to doing a final multiplication by 3. This aid to the student helps to reduce the purely mechanical demands of the curriculum and is in conformity with recent emphasis upon the use of hand-held calculators in elementary mathematics courses.

HELP routine. This routine is essentially an introduction to other student aids. When called, it displays the same information about recommended study procedures, and about ANSWER, COMPUTE, HELP, HINT, NEW MODULE, and SOLVE, as were given in the first day's instructions. Following a command, the initiative remains with the student. Typically, a student will follow that command by calling for one of the aids just mentioned.

HINT routine. The HINT command can be made either before or after attempting an answer to a problem. Cumulatively more complete hints are possible by calling HINT repeatedly, if early hints do not seem adequate to lead to the correct answer. The structure of the list of hints typically matches the steps in the computer solution (see "SOLVE routine," below), helping to give the ISO system its information structure orientation.

NEW MODULE routine. This routine allows the ISO student to ask for a different curriculum module from the one currently being taught. After typing MODULE, the student is given a choice of the available modules. After that choice, the student chooses the difficulty of items (hard, easy, or mixed) to be presented. To change difficulty but maintain the same module, the student types LEVEL, and again is given the choice of three levels.

QUIT routine. The SIGNOUT program is activated at any time the student fails to accept an offer of displaying another problem. Typing QUIT or BYE also activates this program.

SOLVE routine. The utility command SOLVE is closely related to SUPERSOLVE, a program for finding

Table 3
Partial Production System for Solving Fraction
Problems Using SOLVE

P1	SOLVE	Solution=term ((problem = (term =) $\Rightarrow$ (satisfy SOLVE))
P2	SOLVE	((problem not = (term =)) (problem has nested term) $\Rightarrow$ (frame = nested term) (setgoal NEST))
P3	SOLVE	((problem not = (term =)) (problem has no nested term) (problem has more than two terms) $\Rightarrow$ (frame = term1 operator1 term2) (setgoal FIRST2))
P4	SOLVE	((problem not = (term =)) (problem has no nested term) (problem has only two terms) $\Rightarrow$ (frame = problem) (setgoal LAST2))
P5	NEST	((frame = term) $\Rightarrow$ (problem = substitute frame for nested term in problem) (satisfy NEST))
P6	NEST	((frame = (term op term)) $\Rightarrow$ (setgoal COMBINE))
P7	FIRST2	((frame = term) $\Rightarrow$ (problem = substitute frame for term1 operator1 term2 in problem) (satisfy FIRST2))
P8	FIRST2	((frame = term op term) $\Rightarrow$ (setgoal COMBINE))
P9	LAST2	((frame = term) $\Rightarrow$. (problem = (frame =)) (satisfy LAST2))
P10	LAST2	((frame = term op term) $\Rightarrow$ (setgoal COMBINE))
P11	COMBINE	((frame = term) (term is in reduced form) (numerator less than denominator) $\Rightarrow$ (satisfy COMBINE))
P12	COMBINE	((frame = term) (term in reduced form) (numerator greater than or equal to denominator) $\Rightarrow$ (setgoal IMPROPER))
P13	COMBINE	((frame = term) (term not in reduced form) $\Rightarrow$ (setgoal REDUCE))
P14	COMBINE	((frame = term op term) (op = +) $\Rightarrow$ (setgoal ADD))
P15	COMBINE	((frame = term op term) (op = -) $\Rightarrow$ (setgoal SUB))
P16	COMBINE	((frame = term op term) (op = $\times$) $\Rightarrow$ (setgoal MUL))
P17	COMBINE	((frame = term op term) (op = /) $\Rightarrow$ (setgoal DIV))
P18	ADD	((denominator of term1 = denominator of term2) $\Rightarrow$ (frame = sum of term1 and term2) (satisfy ADD))
P19	ADD	((denominator of term1 not = denominator of term2) $\Rightarrow$ (setgoal LCM))
P20	SUB	((denominator of term1 = denominator of term2) $\Rightarrow$ (frame = difference of term1 and term2) (satisfy SUB))
P21	SUB	((denominator of term1 not = denominator of term2) $\Rightarrow$ (setgoal LCM))
P22	MUL	((term1 or term2 is mixed) $\Rightarrow$ (setgoal IMPROPER))
P23	MUL	((term1 and term2 are not mixed) $\Rightarrow$ (frame = product of term1 and term2) (satisfy MUL))
P24	DIV	((term1 or term2 is mixed) $\Rightarrow$ (frame = mixed term) (setgoal IMPROPER))

Table 3 Continued

P25	DIV	((term1 and term2 are not mixed) $\Rightarrow$ (setgoal INVERT))
P26	DIV	((frame = term) $\Rightarrow$ (satisfy DIV))

solutions as needed in checking student answers and in facilitating the operation of ANSWER, HINT, and SOLVE. The SOLVE process either generates (for fractions and for negative numbers and linear equations) or simulates (for the remaining modules) a production system (Newell, 1973) for solving each problem. Table 3 shows such a system for fraction problems. The computer starts with the SOLVE goal of Production 1 (P1) and then looks down the list in Table 1 for the first condition (left side of an arrow) which is met. Then its corresponding action (right side of arrow) is performed, and a new search from the top of Table 3 is begun. This continues until the full task is solved.

Figure 2 indicates successive goal-stack structures (rows of Figure 2) developed by following the procedures of Table 3 plus the productions of 27 LCM, 28 IMPROPER, 29 INVERT, 30 INVERT, and 31 REDUCE (omitted from Table 3 to save space). The production number associated with each previous top goal and future section is shown on the line before action is taken.

A similar process is involved in the use of HINT and SOLVE routines in story problems. In an investigation of processing strategies for solving algebra word problems, Hinsley, Hayes, and Simon (Note 3) found evidence that people employ various "problem-type schemata" or "frames" that assist in translating story problems from prose to equations, or sets of equations, which may then be directly solved. More specifically, they found that: (1) people can categorize problems into types; (2) people can categorize problems without completely formulating them for solution; (3) people have a body of information about each problem type which is potentially useful; (4) people use category identifications to formulate problems in the course of actually solving them. One might hypothesize that people solve story problems by the following process: (1) Identify the appropriate problem-type schema. (2) Identify and assign values provided by the problem to the proper nodes, or empty cells, of the schema. (3) Carry out the operations on these values as specified by the schema.

Early assistance in solving story problems provided by HINT and the initial stages of the step-by-step solutions generated by SOLVE are largely structured by the theoretical process mentioned.

DETAILS OF AFO SYSTEM DESIGN

Differences in Function from ISO System

The present ISO system permits the student to select his or her own order of modules to study at the begin-

set SOLVE

SOLVE	set NEST (P2)				
SOLVE	NEST	set COMBINE (P6)			
SOLVE	NEST	COMBINE	set ADD (P14)		
SOLVE	NEST	COMBINE	ADD	set LCM (P19)	
SOLVE	NEST	COMBINE	ADD	LCM	satisfy LCM (P27)
SOLVE	NEST	COMBINE	ADD		satisfy ADD (P19)
SOLVE	NEST	COMBINE	set IMPROPER (P12)		
SOLVE	NEST	COMBINE	IMPROPER		satisfy IMPROPER (P28)
SOLVE	NEST	COMBINE			satisfy COMBINE (P11)
SOLVE	NEST				satisfy NEST (P5)
SOLVE	set LAST2 (P4)				
SOLVE	LAST2	set COMBINE (P10)			
SOLVE	LAST2	COMBINE	set DIVIDE (P17)		
SOLVE	LAST2	COMBINE	DIVIDE	set INVERT (P25)	
SOLVE	LAST2	COMBINE	DIVIDE	INVERT	set MUL (P29)
SOLVE	LAST2	COMBINE	DIVIDE	INVERT	MUL
SOLVE	LAST2	COMBINE	DIVIDE	INVERT	satisfy MUL (P23)
SOLVE	LAST2	COMBINE	DIVIDE	INVERT	satisfy INVERT (P30)
SOLVE	LAST2	COMBINE	DIVIDE		satisfy DIVIDE (P26)
SOLVE	LAST2	COMBINE	set REDUCE (P13)		
SOLVE	LAST2	COMBINE	REDUCE		satisfy REDUCE (P31)
SOLVE	LAST2	COMBINE			satisfy COMBINE (P11)
SOLVE	LAST2				satisfy LAST2 (P9)
SOLVE					satisfy SOLVE (P1)

Figure 2. Goal stack trace for " $1/2 / (2/3 + 1-1/6) = .$ "

ning of a session or later by use of the NEW MODULE command. While most AFO systems have relatively few items, the generative feature of the present system leads to a much larger pool from which training items are selected. This should somewhat increase the generality of teaching effects in the AFO condition; research on the effects of this change may be desirable in the future.

The current AFO system not only prevents student selection of specific modules to study, it also prevents use of the COMPUTE and HINT commands and use of the full SOLVE routine. None of these aids is frequently used in other AFO systems. In place of the original SOLVE routine, the AFO system uses a modified SOLVE for a sample problem rather than the current problem. Thus, the student must infer how the model solution method can be modified for use with the specific problem presented. The AFO condition excludes from the HELP routine mention of the COMPUTE and HINT routines. These routines are rendered inoperative by a series of flags activated by the student's experimental condition found in his or her data file at the time of signing in for a session.

A Note on the Implementation of Student-Activated Routines

The UNIX operating system installed on the PDP 11/45 computers employed here permits the allocation of approximately 32K words of 16-bit memory for any single process. The set of files necessary for the operation of a single module in the AFO

condition plus the LISP interpreter can operate in an environment of this size with some overlaying of programs required. As Harvard's LISP is an interpreter, and the size of the programs required for the operation of ISO student-activated HINT and SOLVE routines is large, overlaying these routines each time they are called results in excessive time delay and introduces such problems with LISP's memory management that the overall process soon fails. Consequently, for each curriculum module, there exists a pair of files stored on disk. Each file contains an entire LISP interpreter and the functions required for the HINT or SOLVE routines for that module.

When a student in the ISO condition types HINT or SOLVE, the monitor writes information to a data file and asks UNIX to activate the new LISP process containing the routine requested. This second process reads the data file and completes the request. UNIX then returns control to the original LISP process. This procedure greatly reduces response time at the cost of a significant use of disk storage. A number of the LISP files permit speeding up instruction and item presentation.

Use of Performance Criteria in Item and Topic Selection

Recordkeeping and control of the curriculum in either AFO or ISO instruction requires categorization of item and response types. Whenever an item is presented to a student, the subsequent response is characterized as falling into one of the following categories: 0 (correct), 1 (correct except for sign or a need to be further simplified), 2 (incorrect), 3 (skip), A (answer requested), C (COMPUTE command given), F (HINT command given after hints exhausted), G (gave up; this can only be a second response to a problem), H (HINT command given), L (LEVEL command given to change difficulty level), M (MODULE command given), P (HELP command given), Q (QUIT command given), S (detailed solution requested). More than one response category may be scored on an occurrence of a problem because of student use of various aids to instruction.

The above discussion implies that specific student responses are not stored by the computer. The answers stored by the rules just given permit considerable analysis of the quality of student answers and of the frequency of use of student-initiated utilities such as ANSWER and SOLVE.

The response record is stored as the value of a variable called RCAT, together with a characterization of the item, the variable called QCAT. The QCAT has as its value the level and type of item, the curriculum module involved, and other relevant information. For example, the fractions module has six elements on the QCAT tag for an item. Those elements specify: (1) that fractions are involved; (2) the level and sublevel

of a numerical problem (if present); (3) whether mixed fractions are involved; (4) whether negative fractions are involved; (5) whether the item is a numerical, word, or story problem; and, (6) whether the item is an immediate repetition of the previous one (e.g., because of a partially correct answer leading to another try at the problem).

For an AFO student to complete a curriculum module, that student must meet a performance criterion on each part of that module. Also, criteria must be met before moving from one section of a module to another section of the same module. One cannot receive instruction in the AFO condition for a higher level part of the module until the appropriate performance criteria (not shown in this article) have been reached on all lower level parts. Word and story problems are somewhat independent of the numerical problems with specific level assignments. Typically, word and story problems are not presented until criterion has been reached at Level 1. Thereafter, word and story problems, if used for that module, are randomly interspersed with numerical problems. The performance criterion for word problems or for story problems may be reached either before or after the criterion for Level 2 or Level 3 numerical problems.

DESIGN OF CONFLUENT (AFFECTIVE) PROCEDURE

The confluent procedure is intended to make the computer seem like a friendly teacher. In all sessions students are frequently called by their first names. At the beginning of a session, they are asked how they feel; on Session 2 and thereafter, they are reminded of their feelings from the previous session. In Session 1, they are each asked to give the computer a personal name. After student-oriented instructions in Session 1 on how to use the computer and how to learn the mathematics lessons, students are told their astrological sign and asked if they have a certain personality characteristic popularly associated with that sign. This discussion continues until the student either agrees to a characterization or has rejected a list of four or more such stored properties.

In Session 2 confluent students find the computer asking one member of a pair to leave the terminal and join the other confluent member at a second terminal to receive joint instructions. Then the pair engages in a face-to-face review of the previous lesson on computer use and study procedures, after which the members of the pair again begin work at separate terminals.

In all sessions a sequence of three successive errors on mathematics problems leads to a period in which the student is instructed in a fantasizing procedure or a body-stretching procedure intended to produce relaxation before further study. Confluent exercises are presented near the beginning of each module.

REFERENCE NOTES

1. Carbonell, J. R. *Mixed initiative man-computer instructional dialogues* (BBN Rep. 1971). Boston, Mass: Bolt, Beranek, & Newman, 1970.
2. Suppes, P., Goldberg, A., Kanz, G., Searle, B., & Stauffer, C. *Teacher's handbook for CAI courses* (IMSSS Tech. Rep. 178). Stanford, Calif: Institute for Mathematical Studies in the Social Sciences, 1971.
3. Hinsley, D. A., Hayes, J. R., & Simon, H. A. *From words to equations: Meaning and representation in algebra word problems* (CIP working paper No. 331). Pittsburgh: Carnegie-Mellon University, 1976.

REFERENCES

- BARR, A., BEARD, M., & ATKINSON, R. C. The computer as a tutorial laboratory: The Stanford BIP project. *International Journal of Man-Machine Studies*, 1976, 8, 567-596.
- BROWN, G. I. *Human teaching for human learning*. New York: Viking, 1971.
- BROWN, G. I. The training of teachers for affective roles. *National Society for the Study of Education Yearbook*, 1975, 74(Part 2), 173-203.
- BROWN, J. S., & BURTON, R. R. Multiple representations of knowledge. In D. G. Bobrow & A. Collins (Eds.), *Representation and understanding*. New York: Academic Press, 1975.
- GILKEY, T. J., & KOFFMAN, E. B. Generative CAI in high school algebra. In A. Guenther, B. Levrat, & M. Lipps (Eds.), *International computing symposium 1973*. New York: North-Holland, 1974.
- NEWELL, A. Production systems: Models of control structures. In W. G. Chase (Ed.), *Visual information processing*. New York: Academic Press, 1973.
- UTTAL, W. R., ROGERS, M., HIERONYMOUS, R., & PASICH, T. *Generative computer-assisted instruction in analytic geometry*. Newburyport, Mass: Entelek, 1970.

NOTE

1. The term "ad hoc frame oriented CAI" (AFO CAI) was introduced by Carbonell (1970). To say that the frames are ad hoc ("after the fact," or "with respect to a specific thing") may be inaccurate, but to say that the generation of feedback to student responses is ad hoc (compared to Carbonell's procedures) is acceptable and justifies Carbonell's usage. The linguistically fastidious reader may choose to ignore the words for which AFO is an acronym.