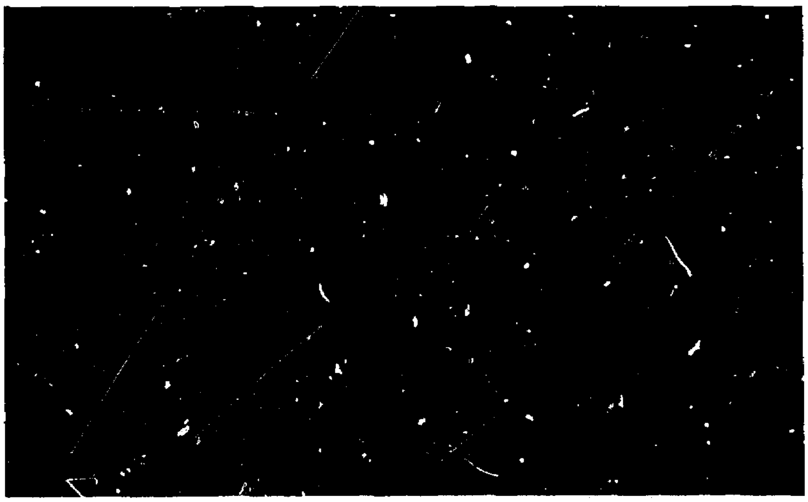
DOCUMENT RESUME

ED 089 675 IR 000 387

| | |
|---|---|
| AUTHOR | Evans, Steven; Klahr, David |
| TITLE | An Educational Assembly System For Student Executed Educational Design: Toward a System of Computer Constructed Education. |
| INSTITUTION | Carnegie-Mellon Univ., Pittsburgh, Pa. Graduate School of Industrial Administration. |
| SPONS AGENCY | Ford Foundation, New York, N.Y.; International Business Machines Corp., New York, N.Y. |
| PUB DATE | Apr 74 |
| NOTE | 32p.; Paper presented at the American Educational Research Association Annual Meeting (Chicago, Illinois, April 15 through 19, 1974); Reproduced from best available copy |
| EDRS PRICE | MF-$0.75 HC-$1.85 PLUS POSTAGE |
| DESCRIPTORS | Autoinstructional Aids; *Computer Assisted Instruction; Computer Oriented Programs; Curriculum Design; Curriculum Development; Independent Study; Individual Instruction; *Individualized Curriculum; *Individualized Programs; Instructional Materials; *Man Machine Systems; Program Descriptions; *Student Centered Curriculum |
| IDENTIFIERS | CCE; Computer Constructed Education; EAS; EASSEED; EAS Student Executed Education Design; *Educational Assembly System; Semantic Net |

ABSTRACT

A computer-based system--the Educational Assembly System for Student Executed Educational Design (EASSEED)--is used to generate individualized curricula for students with varying needs and goals. It combines advances in management and computer technology to define the processes of curriculum development and thus moves toward an explicit operational description of a normative theory of curriculum development. EASSEED functions by directing students to instructional materials, the completion of which fulfills the goals set by the student. The system is organized into content and structure modules; the former direct the student to materials related to achieveable goals while the latter decompose broad goals into progressively more manageable sub-goals until a level is reached at which a suitable instructional material is available. Goals are processed by the system's semantic net; all terms in the system are located in the net and pointers lead the student to higher order, equivalent, and implied terms. The system provides the basis of computer constructed education, for it can be used as a course prerequisite, supplement, or substitute, for course design and subject area definition, as a resource evaluator, and for generalized program support. (LB)

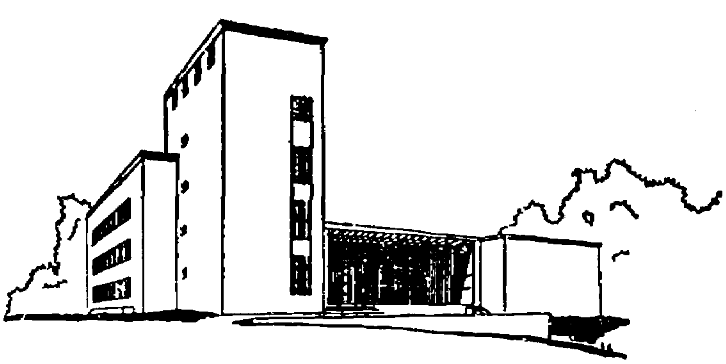


# Carnegie-Mellon University

PITTSBURGH, PENNSYLVANIA 15213

## GRADUATE SCHOOL OF INDUSTRIAL ADMINISTRATION

WILLIAM LARIMER MELLON, FOUNDER



# An Educational Assembly System for Student Executed Educational Design: Toward A System of Computer Constructed Education

Steven Evans
David Klahr
Carnegie-Mellon University



Presented at session #22.09
of the American Educational Research Association
1974 Conference, Chicago, Illinois

Abstract

This research represents an effort to move toward an operationally defined theory of curriculum construction. A computer based system is presented whose action is to generate highly individualized curricula for a wide range of students with diverse backgrounds, characteristics, and educational objectives. The goal is to simulate an omniscient, perceptive, indefatigable human educational consultant and curriculum designer. This research presents an operational example of what has been called computer constructed education (CCE).

Uses for this currently operational system are briefly discussed, including use as a course supplement or substitute, course design, and educational resource evaluator. The current system's status is also noted.

An Educational Assembly System
for Student Executed Educational Design: Toward A
System of Computer Constructed Education

Steven Evans
David Klahr

Carnegie Mellon University

Introduction

The project to be described represents an attempt to deal with a hitherto untreated aspect of education: the creation and distribution of individualized curricula for diverse educational goals as desired and formulated by the student, where every curriculum is tailored to each student's characteristics.[1/] The goal of the project is to construct a system that can generate, upon demand, sequences of information about educational materials and properly organize such information in a curriculum best suited to each individual user.

There are several justifications for this specific goal. The virtue of increased efficiency in education need not be argued here; we take it as given when two-thirds of the 2300 institutions of higher learning in the U.S. are in what the Carnegie Commission on Higher Education calls "financial difficulty". The benefits of individualization have been acknowledged and pursued by an amazingly diverse range of educators, from Robert Glaser (Bolvin and Glaser, 1968) to Ivan Illich (1971).[2/] Our primary justification comes from the observation that one of the most important and sophisticated tasks of the live educator is the determination of the appropirate goals of his students and the selection and organization of materials to achieve those goals. This is not to say that the job is often done well, but rather that one of the reasons that it

---

1/ This work was supported by grants from the Ford Foundation and the International Business Machines Corporation.

2/ A severe critic of "schooling", Illich suggests a redistribution of education, away from centralized institutions such as schools, through "learning exchanges" that, in our view, might become feasible through massive applications of technology such as we are suggesting.

is usually done poorly is that it is extremely difficult. It requires that the educator have access to an enormous amount of potential material, an accurate assessment of the needs and abilities of the students, and a means to relate the two.

The problem is not the availability per se of information about educational materials, for that comes in a never ending stream from publishers, colleagues, special information services, professional groups, libraries, etc. Information is available not only about conventional print materials, but about every other form of instructional technology ranging from video tapes to computers, from special approaches such as simulation games (Zuckerman and Horn, 1970) to collections of course outlines (IBM, 1969) to reviews of reviews (Zinn, 1970).

The problem is the intelligent selection of materials that are relevant to specific educational needs, goals, and abilities. The history of technology in education has been to transform scarce labor inputs into reproducible and readily disseminated capital. From the invention of printing to the utilization of computer technology, the goal has been to efficiently distribute the scarce talents of the best humans in a field. EAS/SEED [3] is such an attempt.

## Overview of the System

The EAS is a system that can generate a curriculum best suited for each individual user. The potential user is a student with an educational goal ranging from something quite specific to a totally general (or ambiguous) goal.

---

3/ EAS/SEED: Educational Assembly System for Student Executed Educational Design

The only constraint is that the goal should be achievable by completion of some sequence of educational materials, e.g., books, lectures, films, courses, seminars, tapes, articles, or problem-sets. Given such a goal, the EAS functions as would an enlightened educational consultant who had a vast awareness of most areas, subject material, job requirements, etc., and who had the time to serve the particular needs of this individual student. Such a consultant would be expected to suggest a program of actions tailored to the student, the completion of which would accomplish the given goal.

The basic components of the system can be divided into two parts: a) structured descriptions of educational goal and material (modules), and b) a network of word relationships (a semantic net). Both of these are created by various subject-matter experts. The user inputs his goal, including information about the area, level, time he wants to spend, etc. Then the system interacts with questions about possible inconsistencies, his prerequisites, and other relevant information. The EAS programs attempt to "understand" (as will be explicated later) the student's goal and then searches for modules that satisfy the goal and the side constraints. Further interactions may occur between student and system. Finally, the student is presented with the optimal curriculum. At this point, he can recycle at any desired level of detail, or he can leave the system and pursue his curriculum. (Note that the EAS does not retrieve the actual materials; it directs the student to them. In fact, it is more accurate to say that the product which the system generates is a study guide, individualized to a particular goal and student.)

To summarize, the goal is to maximize the capability for distribution of educational resources of all kinds, upon demand of the student, in a highly individualized and tailored fashion (while maintaining economic feasibility, of course). The solution takes the form of an elaborate and perceptive educational consultant (albeit a programmed one). Through an interaction with the users, the system can appropriately evaluate their needs, and has at its disposal a huge and varied universe of resources to recommend, whereby the educational goal of the user can be achieved. It chooses from its repertoire, based on its internally created model of the domains of knowledge, a suitable and accessible sequence that would best serve the particular user of the system at that time. Although cognizant of the dangers of acronym pollution (Sailsbury, 1971), we feel that our system can best be described as an operational example of computer constructed education or CCE (Evans,1974). We shall pursue some of the details of such a system in the sections that follow.

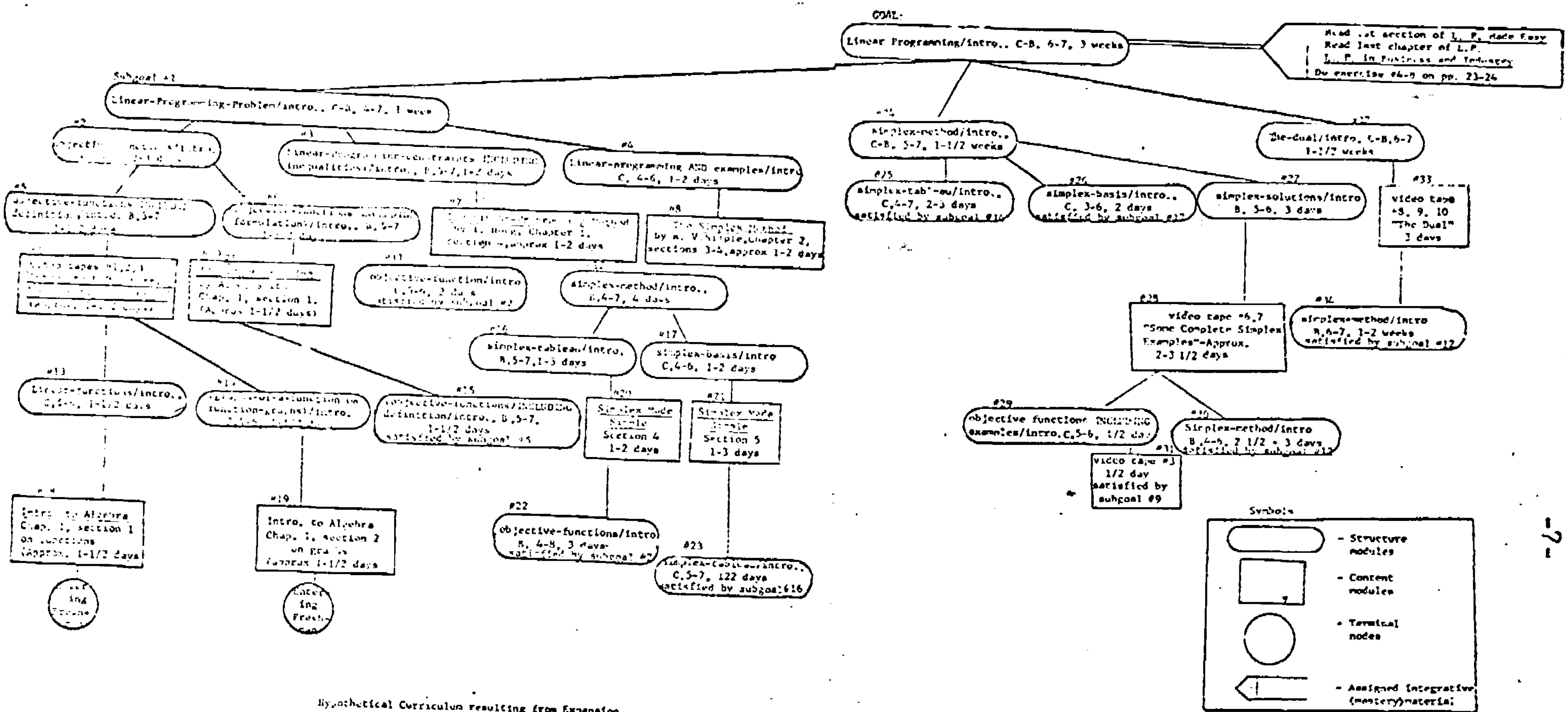
System Organization

One of the main problems involved in creating the system is achieving compatibility with the various pieces of information that need to be compared. The system needs to compare prerequisities that the student has accomplished with the target goal that the student has, as well as with other goals the student may be assigned as a result of his desired goal that is being fulfilled. In addition, there is the comparison of all these goals with goals the system has stored that it can serve (such goals are "served" from the MENU, the module entry universe, which holds all the encoded modules the system can describe to the student.) All these issues of compatibility are resolved by the definition of an educational goal. All goals are described in a limited, formatted mode of expression, which includes several parameters. These parameters include the area about which the goal is concerned, the level of difficulty that the goal is set at, the mastery level, motivation level, the media, and the time spent for the goal. These parameters are set depending on the type of goal being encoded. A prerequisite described by the student gives the area he covered, the media that was involved, the time that was spent, etc. The goal the student desires is a listing of the area wanted, the media preferred, the time wished to be spent, etc. As a result of defining an educational goal, all prerequisites are given as goals accomplished, the target goal is the goal desired, and the other goals needed for the desired goal to be accomplished become subgoals to facilitate the accomplishment of the goal. The specification of a goal, given in this formatted form, is the language of goals; it is the lingua franca of the system. The most complicated part is the area part of the goal which gives the subject matter or area of the goal. Concept or area names can be joined with such connectives as "or", "and", and "including". The satisfaction of a desired goal by supplying the necessary subgoals so that the goal can be accomplished by that particular student is the objective of the system. It may use the prerequisites of the student (as subgoals already accomplished in lieu of re-assigning them). It may also need to assign sub-sub-goals to accomplish the subgoals. An example of the output of the system with the sub-sub... subgoal expansion is found in Figure 1. The details of the repeated assigning of more and more lower-level goals until a level is reached so that the student can handle the initial goals is the subject of another organizational delineation which we shall now pursue: content and structure modules in the module entry universe (MENU) of the system.

Given a particular goal, a corresponding module's purpose is to fulfill that goal. Thus a module is described by the goals it fulfills. Hence each module represents a possible goal that a student may have, and to encode a module is to encode a (possible) goal. We divide modules into two types: content modules and structure modules. Content modules are associated with specific material, the accomplishment of which satisfies the goal which the module corresponds to. In such cases particular resources are specified to be retrieved in order to satisfy the goal. The "prerequisites" section of a content module asks "what is needed to handle these résources?

Structure modules are often associated with more general goals that are in turn dependent upon general subdomains. In this case, goal satisfaction does not require that resources be retrieved directly; instead there is the need to specify subgoals, whose satisfaction will permit the accomplishment of the original goal.

To exemplify these two types of modules, consider two goals: (1) "I want to learn about 'theory of the firm' at an introductory level" and (2) "I want to learn about 'management science' at an introductory level". If we wish to encode modules to satisfy these two goals, in the first case we may encode the book by the same name. The mastery




Hypothetical Curriculum resulting from Expansion of the goal: "Linear Programming/intro." C-B, 6-7, 3 weeks

figure 1

of the book will be the suitable action to fulfill the goal. Prerequisites to reading the book might be an awareness of basic economic issues plus some mathematical aptitude. This is a content module. In the second case, we might structure the field of management science as consisting of the subdomains of operations research, economics, and industrial administration. Having done this, we note that the mastery of these subareas at an introductory level is a suitable action whose accomplishment will result in the original goal being fulfilled. Hence, the goal of understanding the material in this module implies fulfilling the subgoals of "learning introductory operations research, economics, and industrial administration". There is no material, per se, that is to be retrieved. This is a structure module.

The distinction between structure and content modules is somewhat arbitrary, but convenient. Structure modules keep decomposing goals into subgoals until a level of subgoals is reached whereby there is some suitable material or physical resource that will satisfy each subgoal. Maintenance of this dichotomy between content and structure preserves the modularity of the system.

Handling Real Goals: Problems of Ambiguity, Ill-structuredness, and Context

With the system as described, the task of the routines would be to compare the desired goal with the possible goals stored in the MENU, selecting one appropriately matched, expanding any required subgoals by re-entering them as yet other goals to satisfy, while checking the student's prerequisites, previously assigned modules, etc. See Figure 1 preceding as an example of this descending expansion.

However there are several difficulties involved. These difficulties will require elaborate mechanisms to aid in their solution or resolution. We shall remark on some of the difficulties, then describe the outline of the attack. Further explication of the mechanisms will be left for the examples that follow.

The first difficulty arises when the student is interrogated for his goal. The goal must be analyzed for consistency and suitability. To do this, we must have an idea of the relevant prerequisites that the student has. In turn, since he can not be expected to know what will be expected of him, we must be able to suggest the main areas he should consider when giving prerequisites. In addition, the suitability of his goal relies on the compatibility of the generality of the area he wishes to investigate and the level of difficulty he wishes to pursue. These measures must be defined in order to make estimates of such aspects of compatibility and suitability.

In addition,a goal may be specified ambiguously or poorly,the student using the wrong or improper jargon for areas he does not understand(which presumably may often be the case since he is requesting educational information about the are
This ambiguious statement must be comparable with the more properly posed descriptions of modules in the MENU, which professionals have encoded. There is also the issue of structural alterations of the goal to permit a match. That is, the goal "A or B" must be matchable against "B or A", and presumably even matchable against "B" as well as "A" since the intended meaning of "or" would allow either one to suffice. There is also the issue of context. In some contexts, an example of systems management

might better be chosen as a module concerning inventory control if the area of interest of the student is management science rather than an example of an ant colony (unless the systems management concern of the student is in such a context). Determining the context is an important aid in eliminating certain classes of ambiguities as well as making better choices for modules for the student.

Finally, there is a need for some kind of interpretation of a request for an area so that related areas may be suggested that are not precisely the one described but may well do for the student (assuming the student may have posed an essentially ill-posed goal, or in some cases one that just does not fit the MENU's capabilities). In either case, the ability to suggest an essentially equivalent alternate of a "best try" is important if the system is to be as flexible as would be expected of a consultant. These problems are treated extensively by the system. An overview of its methods will now be described below as it attempts to handle the problems mentioned above.

The heart of the solution involves the construction of a semantic net with operators on that net. The net is an extremely large collection of terms, including all terms used in the MENU of described modules, as well as many others. Each of the terms have various pointers to other terms. For each term, and its collection of pointers, one of the pointers indicates those terms which encompass or include or imply that term, another pointer indicates all terms which are essentially equivalent to that term, and a third pointer specifies all terms that are implied by, or derived from, or are subsets of that term. If all knowledge were perfectly hierarchical, we would have a perfect tree structure of the description or taxonomy of knowledge. However areas and topics and concepts (which the terms represent) are interconnected, lack a strict ordering, and are in no way strictly hierarchical. Thus the representation of this structure of knowledge involves an interconnected net, convoluted, turning on itself, and in general quite complicated. This interconnected net is the semantic net used. It is created by collecting the mini-world of higher-order, equivalent, and implied terms around any one term any one encoder uses (a specification task required of each encoder who contributes to the MENU each time a module is encoded). The system intergrates all the terms into a cohesive, complete net that includes all the mini-worlds collected and connected together. The details of semantic nets and their construction will not be reiterated at this time (see Evans, 1973 for a full elaboration of the EAS/SEED application). The reader may wish to consult the general description of such nets (see Shaprio, 1971) or particular applications that have been employed elsewhere (Quillian, 1966; Winograd, 1970). In addition to the semantic net is the implicit net defined by the structure and content modules in the MENU, since each module in the MENU may specify other subgoals necessary for the completion of that module. such specified goals being "linked" to that goal; these goals (which possibly correspond to other modules in the MENU) in turn may point to others, etc. Hence any one module in the MENU may point to other necessary lower subgoals satisfied by the MENU. It is important to note that a module, as a goal, points to other <u>subgoals, not other (sub)-modules.</u> The system "evaluates" what modules may satisfy a subgoal, depending on a student's prerequisites, the context of goals already assigned, etc. Each sequence of chosen modules from the MENU is further student-dependent since at any one point, his prerequisites may fulfill a needed subgoal and hence terminate that branch of assigned goals. In addition, the context of his prerequisites and assigned modules together with his responses during the earlier interrogation phase create a particular context for each student which may differ from other students, all of which can further affect module assignment.

Within the context of the assigning of modules from the MENU, the system performs a series of relaxations on the goal, as is necessary, in order to relieve some of the problems mentioned above. These relaxations take the form of a series of semantic and syntactic relaxations. These will be described briefly, by way of example.

Assume for the moment that the goal of the student was "quantitative methods" at an introductory level, a desire of mastery to equal making a B in the study effort, a motivation level of 5 (on a scale of 1 to 9), any media being acceptable, and a time of 2 weeks which he has alloted. The main problem focuses on the area which we shall assume in this example matches no entries in the MENU. (Though the system performs other relaxations on the time, etc. based on an evaluation of the goal, we shall not consider them at this point; we shall look only at the area relaxations.). One such relaxation that would occur, after other efforts failed, would be the expansion of the goal into "quantitative methods or operations research". This additional term is provided by the semantic net which would determine, by operators on the net, that the term added was sufficiently close as to permit an expansion without serious loss of intent. Later, if further difficulties continued to arise in matching the goal, one syntactic change permitted would be changing the goal to read "quantitative methods including operations research". Later, after further difficulties, a possible alteration would include "quantitative methods including linear programming", again such information coming from the semantic net. These relaxations, depending on the collected universe of understanding the system creates from the micro-views of each encoder of the modules, are at the heart of resolving problems of ambiguicy as well as poor or ill-structured goals.

Essentially, the net operationaly defines the concept of "closeness" and "relatedness" of areas or ideas. Aplying these measures reduces the ambiguity to tolerable levels so that modules from the MENU may be picked which are suitable for the goals. Since another problem was a need to cue the student about relevant prerequisites he should consider noting in giving his background, the net is appropriate in this case for aiding the system. The lower (or implied) terms in the semantic net of the terms involved in his goal statement are those sub-areas that are highly correlated with the subgoals needed for that area (or goal). Thus we retrieve from the net the lower terms of the lower terms of ...the lower terms of the terms that describe his goal which gives a good cueing list for the student. Moreover as the student is assigned modules that are needed to satisfy his goal,the terms of the area-part of the modules form a context that defines the kinds of areas and concepts the student will be working with. When two modules with different terms are considered by the system to be suitable, the operators on the net can calculate which terms of each module are closer to (the context of) terms already assigned. This discrimination permits choices of modules which are better fits for the student. In cases where the module has the same terms, and the system is to pick between them, the mini-world around the terms of that module can be investigated by looking at the encoded module. For example, the upper category of one module on systems operations may be biology (which the encoder notes at the time of encoding the module) while the other module of the same name has computer-science. If the student is involved in programming, the net will calculate that computer science is closer to programming that biology is and chose accordingly. In this way the semantic net permits the system to resolve the problems confronting it in choosing proper modules to assign. We shall now consider several examples of part of the system in action. In the examples, we omit the extensive interrogation portion which asks the student questions, analyzes his goal, and records information about the goal, the student, his responses to suggestions made, etc. These procedures however are straightforward and are commonplace in the computer science literature as well as in in many other application areas.

Examples of the EAS in Operation

In the first example to be discussed, the student posed the goal of asking for instruction in the area of "quantitative methods including linear optimization." [The special word "including" is used in the system to mean "with particular emphasis on the sub-area of"]. This goal was posed at the introductory level, with a mastery desired of the equivalent of making a C, with a motivation level estimated at 6-8, and time desired of 4 1/2 weeks, using any media. Such information is usually abbreviated by the system as:

"quantitative methods including linear optimization/intro., C, 6-8, 4 1/2 wks."

Note that media is dropped if no preference is specified.

As part of the first use of the semantic net (which, recall, stores the combined collection of cognitive maps of the various experts who have coded modules for the system), the system prompts the student for prerequisites that may be pertinent with respect to this goal. In this test, the areas prompted for included:

linear optimization
quantitative methods
dual solutions, dual problems, duality
linear programming
objective functions
constraints
initial solutions
change-of-basis
sensitivity
simplex-method
.
.
.

Given the student's goal, there are no such modules described this way or which even use some of these terms. Using syntactic expansions of the goal together with semantic relaxations (which will be pursued further), the program considered six possible choices of modules that might satisfy this goal. These included those with the area parts given by:

linear programming including prime solutions
linear programming including problem formulation
linear programming including geometrical solutions
linear programming including simplex-method
linear programming dual solutions
linear programming

Side-stepping the five erroneous ones (which focused on some particular aspect of linear programming rather than treating it in general as was intended by the goal), the program properly chose the module "linear programming", introductory, etc. [One might note that it also was not confused with a module within the current universe described by "operations-research" even though this is another virtual synonym with the term "quantitative methods" which does occur in the goal statement.] However, the system discovers that the available module is for a period of time much less than the desired goal's time; on this point, the module is rejected, and hence a time failure is noted. Since this is the only related module the system finds suitable (area-wise), it later would be reported as the best try though deficient in time. Again we note also what the system did not do. Since the time was out of range, the system allowed a small variation to see if that would be sufficient to make this goal fit. After this relaxation, and its failure, the system checked to see if the evaluation of the goal suggested that the original time request might have been a (suspected) poor choice. The system finds that no flags were set in the evaluation of the goal vis à vis time. Hence, failure is reported. In other cases, either small variations in the time permit a match, or goal evaluation has made us suspect, and larger time variations are allowed if trouble occurs on "time".

In the next example, the same goal is inputted, but this time, the time is altered. As will be seen, we shift to 1 1/4 days. The goal is "quantitative methods including linear optimization/introduction, Pass, 3-5, 1 1/4 days." For this similar goal a module is found that is satisfactory; it is given by

Linear programming/introduction, Pass, 3-5, 1 day.

In turn this goal has several subgoals, which expand into sub-subgoals. These are given in Figure 2. Several points can be observed in the expansion. First, when a module was assigned, and then later found appropriate again, the system reassigned it rather than assigning yet another module. When the system could find no module that sufficed to fulfill subgoals, it created dummy modules, in effect indicating to the student the necessary accomplishments he would have to achieve. In this particular test, one subgoal differs with respect to level from another; otherwise it is identical. We note that the system did not falsely relax the level (one of its options) thereby using one or the other module in both places. Often, what the program does not do is as important as what it does do. In addition, different heuristics govern the treatment of the original (student's) goal in comparison to the subsequent subgoals needed to be satisfied. In general, subgoal modules are evaluated differently than the top-level

- target goal: quantitative methods including linear optimization introductory,1, 3-5, 1 1/4 days
  accepted goal: linear programming/intro.,1, 3-5, 1 day
  module:
  (#1)
  - target subgoal: linear optimization or primal simplex method/intro.,2,4-6, 1/4-1/4da.
    accepted: linear programming including simplex method/intro.,1,2-6,2-4hr.
    module
    (#2)
    - target sub-sub-goal: linear programming including problem-formulation/intro.,1,2-6,1-2 hours
      accepted module: linear programming including problem-formulation/intro.,1,2-6,1-2 hours
      (#3)
      - target sub-sub-goal: linear algebra or matrices/intro., 2-3,4-9,1-1 1/2 weeks
        no accepted module; dummy module assigned directing the student to: linear algebra or matrices/intro., 2-3,4-9,1-1 1/2 weeks
        (#4)
    - target sub-sub-goal: linear algebra or matrices/intro., 2-3,4-9, 1-1 1/2 weeks
      assigned module #4
  - target module: linear programming including duality/intro.,2,5-7, 1/4 - 1/2 day
    accepted: linear programming including dual solutions/intro., 1,3-5,3-5 hours
    module(#5)
    - target subgoal: L.P. including problem formulation/intro., 1,2-6, 1-2 hours
      assigned module #3 to fulfill this
    - target subgoal: linear algebra or matrices, intro.,2-3,4-9,1-1 1/2 weeks
      assigned module #4
    - target subgoal: linear programming including simplex method/intermediate,1,2-6,2-4 hours
      no acceptable module found due to level; assigned module #6

Figure 2

An expansion of the goal

linear programming/intro.,1,3-5,1 day

goal concerning the relaxation allowed on the goal parameters.

The next example re-enters this same goal once again, but in this case, the student indicates a prerequisite of introductory linear algebra pursued at a mastery level of an A (encoded as "4") in the course, a motivation level of 6, for 6 weeks duration:

linear algebra/intro., 4, 6, 6 weeks, mini-course.

We note that a subgoal of "linear programming including problem-formulation" included the subgoal:

linear algebra or matrices/intro., B-C, 4-9, 1 week.

The program determines that the prerequisite will suffice (in fact the prerequisite claimed is much more than enough---a fact that does not confound the evaluation), and so assigns the claimed prerequisite, producing the tree given in Figure 2 with the exception that we now have modules for linear algebra, as shown in Figure 3.

In the next examples, we have a desired goal of the form "(linear programming or operations research) including simplex method," where the associated level desired is advanced, with mastery of C (or "2"), a motivation level estimated to be 7, and time to be invested of 6 weeks. This particular goal points up the capability of the system to handle very complex area-parts in the statements as well perform quite extensive and elaborate syntactical and semantic transformations on such goals. Some of the more obvious transformations included:

( (operations research including linear programming) including simplex method )

( (operations research or linear programming) including simplex method )

operations research including simplex method

linear programming including simplex method

simplex method or linear programming

.

.

.

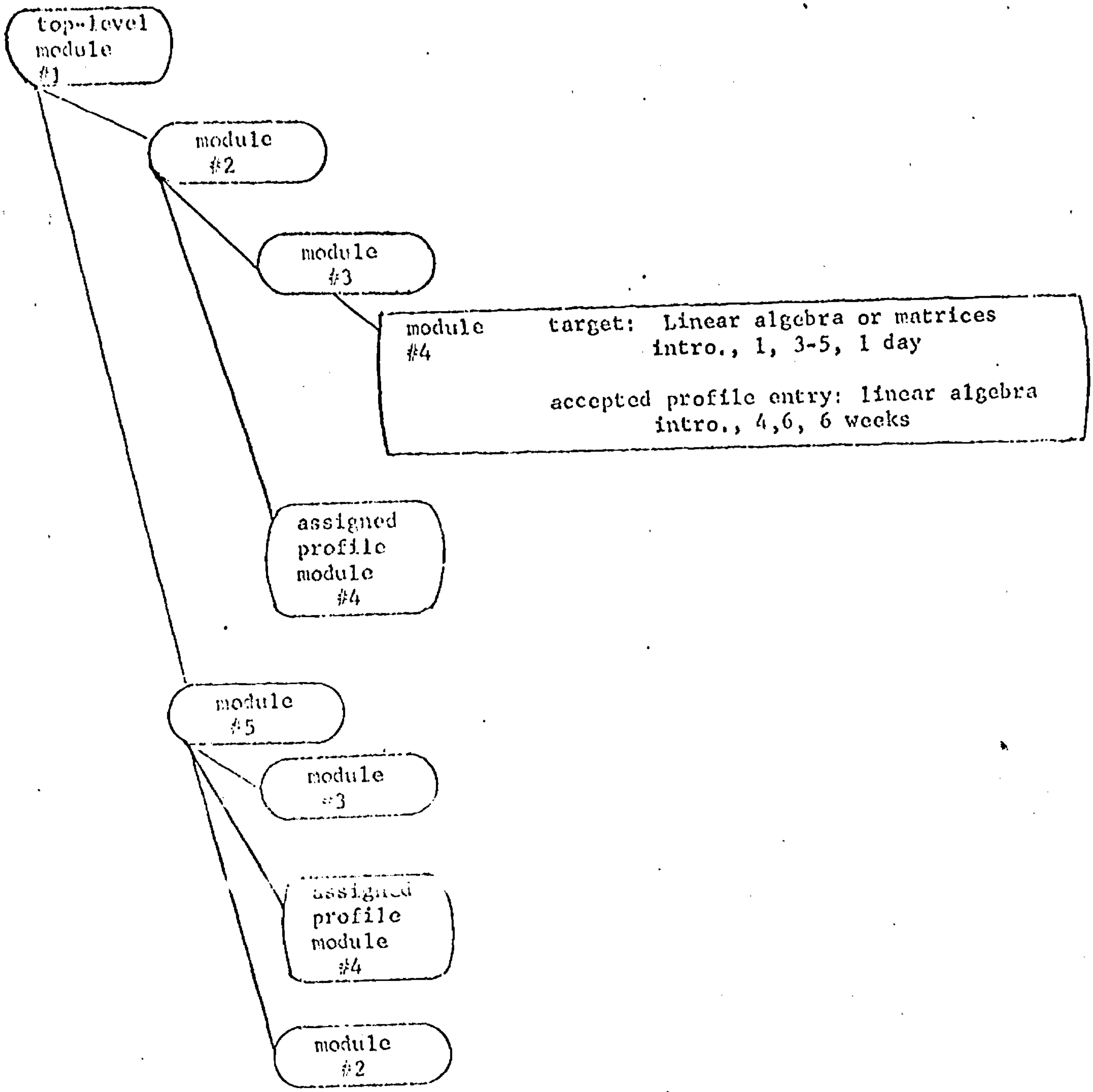



figure 3

Expansion of Goal Using Accepted Profile Entry

In addition to syntactic transformations, the crucial semantic alterations are formed ; some simple ones include:

linear programming including prime solutions

quantitative analysis including linear optimization

.
.
.

In this particular test, the program converged to two modules, described by the areas "linear programming including prime solutions" and "linear programming including simplex method". Since these two candidates both had a level of "introductory" versus the desired "advanced", and since there was not sufficient reason to relax the desired goal to the above goals, the program correctly terminates with a description of its failure to find a module with proper "level". In the next test, the desired goal was similar, but with a reduced level, so that a "hit" could be expected - and a reduced time to make the goal well-posed (reducing the number of flags that might be posted). The system again focuses on these two modules, choosing the best choice of the two, "linear programming including simplex method". It proceeds to create the rest of the curriculum part of which is given in Figure 4.

In another test, the system considered the general goal area of "operations research", generating the first level of subgoals as given in Figure 5. Each of these in turn were expanded.

Other search procedures are in evidence in these tests, though it would take a series of tests in which the desired goals and the module universe differed by some slight variations in certain parameters to cause these differences to appear. For instance the system chooses those modules whose subgoals appear satisfiable by the system over modules whose subgoals (or a smaller percent of whose subgoals) do not seem satisfiable. In addition, each parameter of mastery, motive and media is optimized against each other, all other things being equal, in addition to handling the complicated cases where some of each of the parameters top-level target goal: (linear programming or operations research) including simplex method/introductory, 2.7, 1/2 day

assigned module #1: linear programming including simplex method/introductory, 1, 2-6, 1/2 day

subgoal: linear programming including problem-formulation/intro., 1, 2-6, 1-2 hours

accepted module #2: linear programming including problem formulation/intro., 1, 2-6, 1-2 hours

sub-subgoal: linear algebra or matrices/intro., 2-3, 4-9, 1-1 1/2 weeks

no modules acceptable: assign dummy module #3

subgoal: linear algebra/introductory., 2-3, 4-9, 1 1/4 week

accepted module #3

figure 4

Partial satisfaction of top-level goal

- Operations-research/1, 1, 3-5, 1 1/2 weeks
  - linear-programming/1, 1, 3-5, 1 day
    - linear-programming INCLUDING prime-solutions/1,3-5,1/3-2/3 day
    - linear-programming INCLUDING dual-solutions/1,1,3-5,1/3-2/3 day
  - transportation-problem/1, 1, 3-5, 1/2 day
    - transportation-problem INCLUDING prime-solutions/1,1,3-5, 1/4 day
    - transportation-problem INCLUDING dual-solutions/1,1,3-5, 1/4 day
  - critical-path-method/1,1,3-5,1 day
    - critical-path-method INCLUDING, problem-formulation/1,1,3-5, 1/3- 2/3 day
    - critical-path-method INCLUDING, solution-formulation/1,1,3-5, 1/3- 2/3 day
  - decision-theory/1,1/3-5,1 day
    - decision-theory INCLUDING problem-formulation/1,1,3-5,1/3-2/3 day
    - decision-theory INCLUDING solution-interpretation/1,1,3-5,1/3-2/3 day
  - forecasting-models/1,1,3-5, 1 day
    - forecasting-models INCLUDING problem-formulation/1,1,3-5,1/3-2/3 day
    - forecasting-models INCLUDING solution-techniques/1,1,3-5,1/3-2/3 day
  - scheduling-problems/1,1,3-5, 1/2 day
    - scheduling-problems INCLUDING problem-formulation/1,1/3-5, 1/4 day
    - scheduling-problems INCLUDING solution-ideas/1,1,3-5, 1/4 day
  - inventory-problems/1,1,3-5,1 day
    - inventory-problems INCLUDING problem-formulation/1,1,3-5,1/3-2/3 day
    - inventory-problems INCLUDING solution-interpretations/1,3-5,1/3-2/3 day
  - operations-research INCLUDING applications/1,1,3-5,2 days
    - O. R. INCLUDING (applications AND inventory-problems)/1,1 3-6, all, 4-6 hours
    - O. R. INCLUDING (applications AND decision-theory)/1,1, 3-6, all, 4-6 hours
    - O. R. INCLUDING (applications AND critical-math-method)/1,1, 2-6, all, 1-2 hours
    - O. R. INCLUDING (applications AND linear-programming)/1,1,, 3-6, all,

Figure 5

Partial expansion of subgoals of the top-level : Operations Research are satisfied to varying degrees. This search goes on in conjunction with relaxation of parameters if goal analysis prompted us to anticipate trouble on some particular parameter. Finally, the system makes discriminations along "context", such a context being built up from previously assigned modules as well as other information collected during the goal-input phase. Such information is requested on a "need to know" basis, where a heuristic recipe estimates the amount of information that will be requested. Such a recipe number (corresponding to levels of inquiry) based on the inputted goal, etc., is calculated by the system.

Although the complete outputs of all the tests run have not been included here, the feasibility of the design proposed for an education assembly system for student-executed educational design has been confirmed by subsequent experimentation. In addition, the encoding and collecting of cognitive maps has been successful as well as its operationality demonstrated. The complete results of an EAS/SEED experimentation involving several uses of the system will be the subject of another report. We next turn our attention to some of these uses of such a system.

Uses of an EAS

Currently, we are considering seven uses of our system in a test program. These are presented below.

Use A: Intra-Course Supplement

Assuming that the faculty member has supervised the creation of modules and a net corresponding to the particular course he is teaching, he may wish to offer independent studies as part of the course (perhaps toward the second half after introducing the basic material, etc.). He may wish to allow more motivated students to investigate a large number of related, adjacent, or more specialized areas with which he does not intend to concern the whole class. Such a supplementary capability would be available via the EAS system. It would act as an independent consultant in the course (or to offer another analogy, a knowledgeable Ph.D. student as an assistant in the course, though in this case, constantly available).

Use B: Full Course Use

A faculty member may wish to give (or see supported) a particular area in his field of interest. However, he may not wish to offer another course in that area (especially in addition to his regular load for the semester). The EAS system can accomodate such a desire. Since the system has as one of its central features a semantic capability in order to deal with fuzzy, ill-posed, ambiguous, or poorly posed inquires, it can handle students who wish to pursue some area but who have no real expertise in that area (for otherwise they might very well proceed entirely on their own with no assistance from anyone). The faculty member, by suitably supervising the net construction, can make such an independent studies course available. Such a course is given by and supervised by him; but it is largely unattended and requires little resource investment by him once the areas have been encoded. Thus a university may move from an environment of repeated course production towards one of course management where professors manage the student's progress rather than regulate it.)

Use C: Prerequisite Resolution

Some universities already use an informal subsystem to accomodate students who do not have certain prerequisites. There may be video tapes on the use of TSS, FORTRAN, etc. In addition some departments may also rely on certain mathematics courses taught in the mathematics department for those who need or desire such foundations. This later resource is at times not optimal since there may be partial coverage of the material needed, or in other cases, overkill. The EAS system can accomodate the demand for quite diverse needs for prerequisite subjects that may support, impinge or intersect the particular faculty member's current course material. A properly created net allows the student to access a multitude of prerequisites at many stages (and at various levels, etc.) in subareas, as needed. Not only are such prerequisites made known (or made clear) to him, but the faculty is relieved of the burden of managing such diversions for each course. As a consequence, fewer assumptions need be made about the student, and the student need impose fewer constraints on his range of formal study. Moreover, the effort now directed in courses to establish prerequisites, which is often given limited time or resources (e.g., chapter 0 of the book), may be rechanneled elsewhere.

Use D: Course Design

It is expected that as a faculty member develops a richer and more elaborate network (and as the system accesses other related networks), he then can use the system for course design. By entering the profile of the normative, hypothetical student he expects to teach, as well as the goal that represents the courses' subject area, he can use the curriculum generated by the system as the basis of his own course outline for that subject area. Since the system has access to not only that faculty member's net but other nets as well, the aggregated course production capacity of the faculty becomes a partially shared resource. In conjunction with use C, some of the more unrewarding parts of course generation (i.e. - prerequisite resolution) could be avoided.

Use E: Subject Area Definition

In conjunction with the above use, when certain formal or official areas are entered as goals to the system, the system then presents a uniform way to define the expectations of competence in that area or domain. For example, the area of artificial intelligence may be included as one of the parts of a "systems area" qualifier, at some specified level of competence. The student enters the goal of artificial intelligence at that level (together with the other descriptors allowed), and the resulting curriculum defines the expectations of the school for a student meeting that requirement.

Use F: Resource Evaluation

Again in the same context of a multi-net environment, the administration (or head of a depatment, etc.) may enter certain goals, representing areas or topics he wishes to see supported. Then the resulting curriculum becomes a resource evaluator. The more the system can pull together various parts of many (perhaps diverse) nets and complete the curriculum, the more the total educational system already has the resources necessary to support such a goal. The system is able to indicate the kinds of prerequisites and subgoals it was searching for but failed to find. Where the curriculum indicates missing portions is where resources need to be directed. Thus a vertain amount of inventory control is possible.

Use G: Generalized Program Support

By generalizing the multi-net environment to its natural limit and hypothesizing nets that cover all the areas with which some program is concerned, we can then use the system as the mainstay of the program itself. The student's main task is to move through the net, extracting the curriculum that best suits his goals and completing that curriculum. The whole program becomes defined by the system itself. For example, perhaps a college wished to support a full political science program which it currently did not have for thos students wishing to include the classical areas of political science in addition to subareas already support. However, they may not wish to invest any labor in the project (i.e., permanent faculty position). Then they might generate a net and collect the resources that were described in the net (e.g., books, films, courses at other colleges, journal articles). The EAS would act as the supplement program, making available suitable curricula, as appropriate, for a wide range of inquiry. With a minimal updating, a classical political science program could be made available, at a supplemental level. This use could of course be applied to the school or university level too.

Though these uses represent some of natural applications of an EAS at institutions of higher learning, there are other types of uses as well as other environments. One such use is university accounting. An administrator might put in hypothetical goals of interest. Assuming every module was assigned a cost, the resulting curriculum could be a measure of the cost to fulfill that goal, as all the individual module costs were totalled. Other areas have been suggested for application, including the use of such a system in a legal analysis setting. However we shall not pursue such applications further at this point in time.

Conclusions

The EAS/SEED system introduces another application of recent advances in methodology combined with current computer technology. Moving past management and analysis of instruction and education, this effort attempts to define and develop the processes themselves involved in the creation of the curricula. In this sense, this research effort is an attempt to move toward an explicit formulation of an operational description of a normative theory of curriculum design. The success of such an effort is the degree to which true computer constructed education (CCE) can be achieved.

Finally, we would note that the system absorbs and organizes the most significant part of its information and data store, the semantic net, by virtue of relying on large numbers of localized area experts, whose views are coagulated into a cohesive whole by the system. In effect, the system then is a hybrid problem-solver, effectively drawing on the cognitive maps of educators in order to attempt to duplicate some of their intelligent behavior. Thus the problem-solving synthesis involves a man-machine mix. If the cognitive maps of the best educators are captured in the semantic net as the driver of the "intelligent" portion of the system, then the system will indeed distribute scarce talents over a wider range (though not without some degradation of performance). An efficient distribution then would aid one of our primary goals of contributing to educational efficiencies.

But as noted, this is accomplished only by a deeper understanding of the processes, methods, and organization of curriculum creation. Computer constructed education attempts to answer this question in an operational mode. This system is a first step toward such an understanding.

# REFERENCES

1. Bolvin, J. O. and R. Glaser. Developmental Aspects of Individually Prescribed Instruction, Audio Visual Instruction, 13:828-831, 1968.

2. Evans, Steven. The Structure of Instructional Knowledge: An Operational Model, Instructional Science, 1974, vol. 2, no. 4, 421-450.

3. Evans, Steven. The Conceptual Blueprints of the Processes and Structures for Computer Constructed Education, Working Paper, Carnegie-Mellon University, Pittsburgh, 1973.

4. Illich, Ivan. Deschooling Society, New York; Harper and Row, 1971.

5. International Business Machines Corporation, Curricula Involving Computers in Collegiate Schools of Business, White Plains, N. Y.: IBM Technical Publications Dept., 1969.

6. Quillan, M. Ross. Word Concepts: A Theory and Simulation of Some Basic Semantic Capabilities. In M. Minsky (ed.) Semantic Information Processing, Cambridge: MIT Press, 1968.

7. Salisbury, Alan B. Computers and Education: Toward Agreement on Terminology, Educational Technology, 1971, 11(9), 35-39.

8. Shapiro, S. C. A Data Structure for Semantic Information Processing. Unpublished Ph. D. dissertation, University of Wisconsin, Madison, Wisconsin, 1971.

9. Winograd, Terry. Procedures as a Representation for Data in a Computer Program for Understanding Natural Language. Ph.D. Dissertation, M.I.T., Boston, 1970.

10. Zinn, Karl. An Evaluative Review of Uses of Computers in Instruction. Project CLUE (Computer Learning Under Evaluation), University of Michigan, 1970 (Draft).

11. Zuckerman, David W. and Robert E. Horn. The Guide to Simulation Games for Education and Training, Cambridge, Mass.: Information Resources, Inc., 1970.