DOCUMENT RESUME

ED 077 467 HE 004 303

AUTHOR	Klahr, David
TITLE	Educational Research and Development Unit Graduate School of Industrial Administration, Carnegie-Mellon University. Third Annual Report.
INSTITUTION	Carnegie-Mellon Univ., Pittsburgh, Pa. Graduate School of Industrial Administration.
SPONS AGENCY	Ford Foundation, New York, N.Y.
PUB DATE	Dec 72
NOTE	71p.
EDRS PRICE	MF-$0.65 HC-$3.29
DESCRIPTORS	*Conceptual Schemes; *Curriculum Development; Educational Innovation; Educational Objectives; Graduate Study; *Higher Education; *Individualized Curriculum; *Models; Systems Approach

ABSTRACT

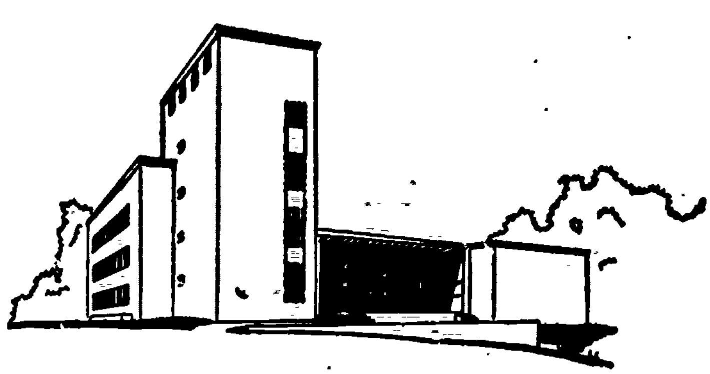
The Educational Research and Development Unit (ERDU) of the Graduate School of Industrial Administration at Carnegie-Mellon University was formally established in January 1970 by a 5-year grant from the Ford Foundation. This report describes the research and development activities in which ERDU was engaged from January to December 1972 for the Ford Foundation. The principal activity during that year was the further development and programming for the Educational Assembly System. (EAS), a system that can generate information about educational materials organized in a curriculum best suited to each individual user. The data base of the system consists of 2 parts: (a) structured descriptions of educational goals and materials, and (b) a network of work relationships. 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., and the system interacts with questions about possible inconsistencies, prerequisites and other relevant information. Finally the student is presented with the optimal curriculum. (HS)

Carnegie-Mellon University

PITTSBURGH, PENNSYLVANIA 15213

GRADUATE SCHOOL OF INDUSTRIAL ADMINISTRATION

WILLIAM LARIMER MELLON, FOUNDER



EDUCATIONAL RESEARCH AND DEVELOPMENT UNIT
GRADUATE SCHOOL OF INDUSTRIAL ADMINISTRATION
CARNEGIE-MELLON UNIVERSITY

Third Annual Report

to the

Ford Foundation

Submitted by

David Klahr
Director, Educational Research and Development Unit

Pittsburgh, Pennsylvania

December 1972

The Educational Research and Development Unit (ERDU) of the Graduate School of Industrial Administration at Carnegie-Mellon University was formally established in January 1970 by a five year grant from the Ford Foundation. In July 1970 an additional two year grant was received from the IBM Corporation for work on a specific project within the general purview of ERDU. This report describes the research and development activities in which ERDU has been engaged during January - December 1972. Prior reports (Klahr 1970, 1971) summarize the earlier activities.

The principal activity during this year has been the further development and programming for the Educational Assembly System (EAS). Although work has continued in the evaluation of student attitudes, faculty productivity, and teaching effectiveness, it has been proceeding at a much slower rate than previously. This report consists of two sections, a description of our progress on the EAS, and a brief summary of the status of the other evaluation projects.

THE EDUCATIONAL ASSEMBLY SYSTEM

I. Overview

Our previous reports have described the overall purpose and scope of the EAS. In this section we present a summary in the form of questions and answers.

Q: What is the Educational Assembly System (EAS)?

The EAS is a system that can generate information about educational materials organized in a curriculum best suited to 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. 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.

A highly simplified sketch of the basic components is presented in Figure 1. The data base consists of two parts: a) structured descriptions of educational goals and materials (modules), and b) a network of word relationships (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, prerequisites and other relevant information. The EAS programs attempt to "understand" (see Question C) 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.)

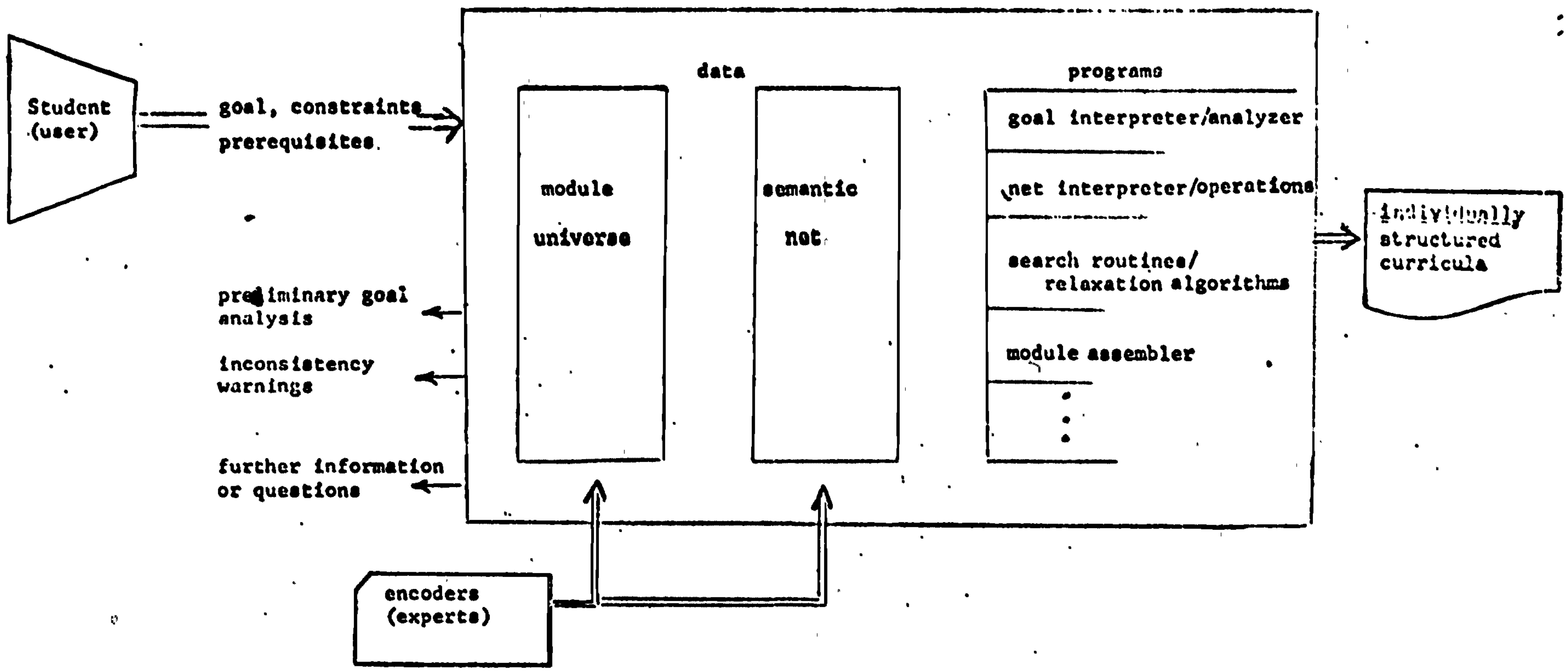


Fig. 1. Simplified schematic of EAS components

B: How does it differ from a printed collection of curricula? Couldn't we just keep the various curricula in a file and let the student browse through them, choosing the one he wants?

Since the system exists as a computer program, we could indeed hand simulate the entire operation. Consider what it would require. Imagine that some area requires three subgoals to be accomplished before the stated goal can be considered to be satisfied. For instance, a student might be expected to look into linear programming (LP), decision-theory, and applications in order to accomplish a goal of introductory operations research (OR). In turn, each of these subareas might involve three subareas. For example, applications might include inventory, scheduling, and forecasting. In turn these might each refer to three different pieces of work, other subdivisions, etc. Thus we have 40 modules forming one specific curriculum in OR. Notice, however, that the subgoal of LP could have been a valid goal in its own right. Similarly, one might have wanted to pursue just the inventory applications in OR. Hence, any one of the various subtrees below any node is also a possible curriculum. In our example, there are 40 possible subcurricula. In addition, since the student might have any combination of prerequisites, any one of the subgoals of any goal may be omitted in some particular case. In general, there are as many different curricula as there are actually different combinations of the various subbranches. That is, there are as many different curricula possible as there are subsets. Since the number of subsets possible from n items is 2^n, we have a rather large upper bound (2^{40} in our example). In addition, 2^n does not include permutations. Since sequencing is pedagogically relevant in a curriculum, the upper bound is raised. Thus it is clear that when one is talking about between 1,000 and 5,000 modules, the concept of keeping a conventional file becomes, at best, clumsy. In addition, we then add the issue of updating, changing, and editing onto this large (and repetitive) universe to clinch the argument that computerized data management, and heuristic file handling is the only feasible approach. (However, this does assume that the criteria of individualization of study, and personalization of curricula holds. Needless to say, the status quo is one of uniform curriculum summary with limited and specified variation.)

In addition to the sheer-size argument given, the EAS system performs other functions strictly orthogonal to a file's conventionally static conception. The original incoming goal of the student is analyzed and evaluated. In an interaction with the student the EAS generates suggestions for improvement and allows responses by the student. Information gained during this period is used to focus and limit search. The system has the semantic capability of transforming a fuzzy or poorly posed goal into one of a well-defined search procedure in a hugh search space. Ambiguity is transformed into exponentially large possibilities, whose size is then pared (though not quite exponentially) by a heuristic search, relying on the trade-off of speed of the search to compensate for the size of the generated space.

This effort, in real terms, makes the concept of a file substitute inappropriate; it is much like arguing that a 100 by 100 matrix can be inverted by hand by certain efficient algorithms. It is true, but no one would be willing to undertake the task.

C: How does the system "understand" the naive user?

The system's understanding rests on its use of a semantic net (in conjunction with syntatic transformations) developed by the expert encoders. This semantic net contains information about each term used in the description of any educational module. For example, one module may be described, in part by the term OR. The semantic net might hold the information that OR can be considered a subarea of applied mathematics as far as the use of that module is concerned. In addition, the net holds the information that LP is a subarea of OR (just as is the transportation-problem, etc.). Then, when the student asks for an area of applied mathematics that includes applications to transportation-problems, and when no modules are described that way, search operators on the net will derive the possible substitute of OR including applications to transportation problems. Presumably there is a module that has this description. Net operators are part of the system design which we believe will stimulate reasonable semantic inference in this task environment. Similarly, the student may have phrased his inquiry askew to the way the module is described. As an example, the student may have been interested in "OR or simplex-method." Thus syntactic transformation are also applied. The system will search for a module by this name (and probably not find one); however, it will also search for a module with the name "OR INCLUDING simplex-method." Both semantic and syntactic changes in the initial goal are made only when the student's original request cannot be directly satisfied. Such changes are also made for internally generated subgoals when such subgoals are subsequently found to be not directly satisfiable.

D: What uses might an EAS have at a school such as GSIA?

Use i: 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 could 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 your assistant in the course, though in this case, constantly available).

Use ii: 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 inquiries, 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 GSIA 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 iii: Prerequisite Resolution

GSIA already uses an informal subsystem to accomodate students who do not have certain prerequisites. There are video tapes on the use of TSS, FORTRAN, etc. In addition we 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 iv: 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 v: Resource Evaluation

Again in the same context of a multi-net environment, the administration (or head of a department, 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 certain amount of inventory control is possible.

Use vi: 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 we wished to support a full political science program for those students wishing to include the classical areas of political science in addition to the subareas we already support. However, we may not wish to invest any labor in the project (i.e., permanent faculty position). Then we would generate a net and collect the resources that were described in the net (e.g., books, films, courses at Pitt, journal articles). The net 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.

E: Given that the EAS is supposed to do the job of a talented and knowledgeable human, wouldn't it be cheaper just to let humans do the job?

It appears not (or can be made to appear not). At this point, we must make a host of assumptions because there are several educational environments in which the EAS might be utilized. Let [illegible] the EAS with a "conventional" approach in a specific application: the offering of an <u>individually</u> planned "major" in a single traditional area (e.g., psychology, math, etc.). We assume that a university wants to offer such a program to augment their existing program. A major goal of the program is the construction of highly individualized curricula that are tailored to the specific needs and abilities of students. Such curricula are intended to be well structured and to indicate to the student the relations among the various prescribed instructional modules.

1. Cost of EAS

   a. We assume that the <u>structure</u> of an EAS (i.e., the search programs, the goal language, etc.) exists, but that the <u>content</u> area (e.g., psychology) has not yet been encoded. Thus, our cost estimates will reflect the amortization of this initial investment in expensive labor.

   b. The mainstream of a discipline can be captured by encoding approximately 15,000 modules. This estimate is based upon our experience with the areas of artificial intelligence and management science. The appropriate encoding of modules is central to the success of EAS, and requires the skill of top level subject-matter experts. We estimate a total cost of $100,000 to encode a single area as follows:

1,000 modules from top experts:	$ 15,000
4,000 modules from major researchers:	35,000
10,000 modules from graduate students:	30,000
	$ 80,000
Keypunch, tape, computer time (for data entry)	20,000
	$ 100,000

c. The annual cost of storing and maintaining the data base for these 15,000 modules is estimated at $20,000. This is a fixed cost, independent of actual usage, based upon existing storage charges at CMU, assuming the 15,000 modules require 15 million characters of storage, plus some updating.

d. The incremental cost of using the system is based upon the number and duration of system-user encounters. Assume that each student using the system has a session to define his initial curriculum, and another three during the semester to refine and update his curriculum. Assume that each session lasts about 15 minutes and that connect and processing costs are $5.00 for each session. That yields $20 per student per semester. Assuming the student uses the system again during the second semester, we get $40 per student per year.

e. We estimate aggregate costs by assuming two different usages and accounting procedures.

   i. Assume a small number of users (department) and prorate the fixed storage costs across all users. For 500 students we have:

storage costs:	$ 20,000
usage: $40 x 500	$ 20,000
	$ 40,000

$80 per student per year.

ii. Assume there is a large number of users and amortize encoding costs over 2 years. For 2,000 students we would have:

amortized encoding costs (2 year period)	$ 50,000
storage	20,000
usage: $40 x 2,000	80,000
	150,000

$75 per student per year (including 2-year amortization)

2. Cost of Conventional Approaches

a. Consider the labor input of consulting with 500 students so that they can create and follow individualized curricula of the kind produced by EAS. We assume that student-faculty interaction takes somewhat longer than the EAS interaction. Thus, each student consumes about 1 hour of faculty time (in each of four sessions) per semester, or 1 man-day per year per student.

b. Assuming $20,000 per year for faculty member, including fringe, overhead, etc., and 200 working days per year, gives us the cost of that one man-day:

$100 per student per year for "consulting"

c. Another way to pose this is to estimate the cost of acquiring a faculty that can devote 500 man-days to consultation and curriculum planning. Once again, assuming 200 working days per year, we need 2 1/2 man years at $20,000 to meet the demands of 500 students. We require 2 1/2 man years x 20,000 or $50,000 per year, or $100 per student per year.

d. There is no reason to assume any economies of scale in the man-to-man approach, so the per-student costs would be the same for 2,000 students as for 500.

3. Comparison

We can summarize the annual per student cost as follows:

Students N	conventional	EAS without amortizing encoding	EAS with amortizing encoding	
			2 years	5 years
500	100	$80	—	—
2,000	100	--	$75	$60*
10,000	100	--	$47*	$44*

*Based upon calculations similar to Section 1.e.ii

II. Creating the Data Base

As indicated in Figure 1, the EAS requires a data base consisting of descriptions of educational resources that can be searched in a meaningful way when the system is attempting to design a curriculum for a user. In this section we will describe the procedure we have devised to collect this crucial information from subject-matter experts.

A: Area definition

The first step in creating the data base requires decisions about the general size and scope of the knowledge domains in which the system will operate. For our prototype EAS, we have decided to work in the area of management science (MS). The EAS should be able to construct a reasonable curriculum for a range of students whose educational goals vary from introductory to advanced MS with an available time span of approximately 5 to 25 months. In Figure 2 we have listed some approximations of the number of modules this will require, as well as a rough mapping into more conventional academic units. We estimate that approximately fifteen thousand modules will be sufficient to span the areas of knowledge under the general heading of MS.

B: Module collection

These modules must be created and encoded before they can be entered into the system. Our collection strategy is to get a small number of top level modules encoded by subject matter experts (e.g., GSIA faculty).

Top level modules usually describe the accomplishments of an educational goal in terms of the completion of some sub-goals. We call these <u>structure</u> modules. Ultimately an educational goal is described in terms of some actual activity that must take place (reading, problem solving, etc.). We call these

I. Modules Needed

A. types of modules

Area	Level	Time (academic yr.)	Equivalence in mini-courses
management science	introductory	5 months (½ yr.)	9 courses
		10 months (1 yr.)	18 "
	intermediate	15 months (1½ yr.)	27 "
		20 months (2 yr.)	36 "
	advanced	25 months (2½ yr.)	45 "

B. number of modules

approximately 7 topics per course
3 sub-topics per topic
3 sub-sub-topics per sub-topic
63 modules per course

x5 variations of each module

total of 315 modules per course

Assuming 45 courses, we get 14,175 modules.

(In addition, assuming two calculus and one linear algebra course as additional supplement, we add 945 modules.)

Total modules anticipated: 15,120 modules

Figure 2

Size of Data Base for Management Science content modules. Our module collection strategy is to have top-level people devote most of their effort to structure modules - thus defining the general structure of knowledge in their areas. Content modules, and lower level structure modules will be encoded by Ph.D. students, since they are often more familiar with appropriate instructional materials for relatively well-defined and narrow goals.

During this year a substantial effort was devoted to the construction of a set of guides and instructions for the encoders. They are contained in Appendices A, B and C. These encoder guides were then used by several top-level and second-level subject matter experts to encode the first set of modules to be used in debugging the EAS programs. Rather than repeat the substance of the Guides here in the narrative, we suggest that they be read at this point in order to understand the encoder's job.

Several GSIA faculty and Ph.D. students worked as encoders, producing a small core of modules (250) and giving up some data for estimating the labor costs of module creation, as well as suggestions for revisions in the encoding guides.

III. Programming the EAS

A: Status

As of December 1971, the general system design had been completed. As of December 1972, the entire prototype system has been implented and is running on the IBM 360/67 at CMU. The entire system, both the processes and the data base, is written in LISP, a recursive function language, in an interpretive mode. This has provided us with a very flexible and powerful language for the initial implementation of our ideas. However, it has been rather inefficient in the use of computer time and space, and like most developing systems, it has many "patches" throughout. Our next step is to totally redesign the "pure programming" aspects of the system so that we can increase the usable data base and decrease the running time.

B: Examples of the EAS in Operation

In the first test to be discussed, we pose 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 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 above 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 explained below) 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 a 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 test, 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, 1, 1, 3-5, 1 1/4 days. For this similar goal a module is found that is satisfactory; it is given by

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

In turn this goal has several subgoals, which expand into sub-subgoals. These are given in Figure 3. 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. Again we note that the system did not falsely relax the level, thereby using one or the other module in both places. In general, subgoal modules are treated differently than the top-level goal, relaxation being one such variation.

The next test re-enters this same goal once again, but in this case, the student indicates a prerequisite of

linear algebra/intro., 4, 6,6 weeks

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

linear algebra or matrices/intro., 2-3, 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 that was given in Figure 3 with the exception that we have modules for linear algebra, as shown in Figure 4a and b

In the next tests, 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 algebra, 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 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-
(#3) formulation/intro.,1,2-6,1-2 hours

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.,
(#4) 2-3,4-9,1-1 1/2 weeks

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.,
module(#5) 1,3-5,3-5 hours

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 3

An expansion of the goal

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

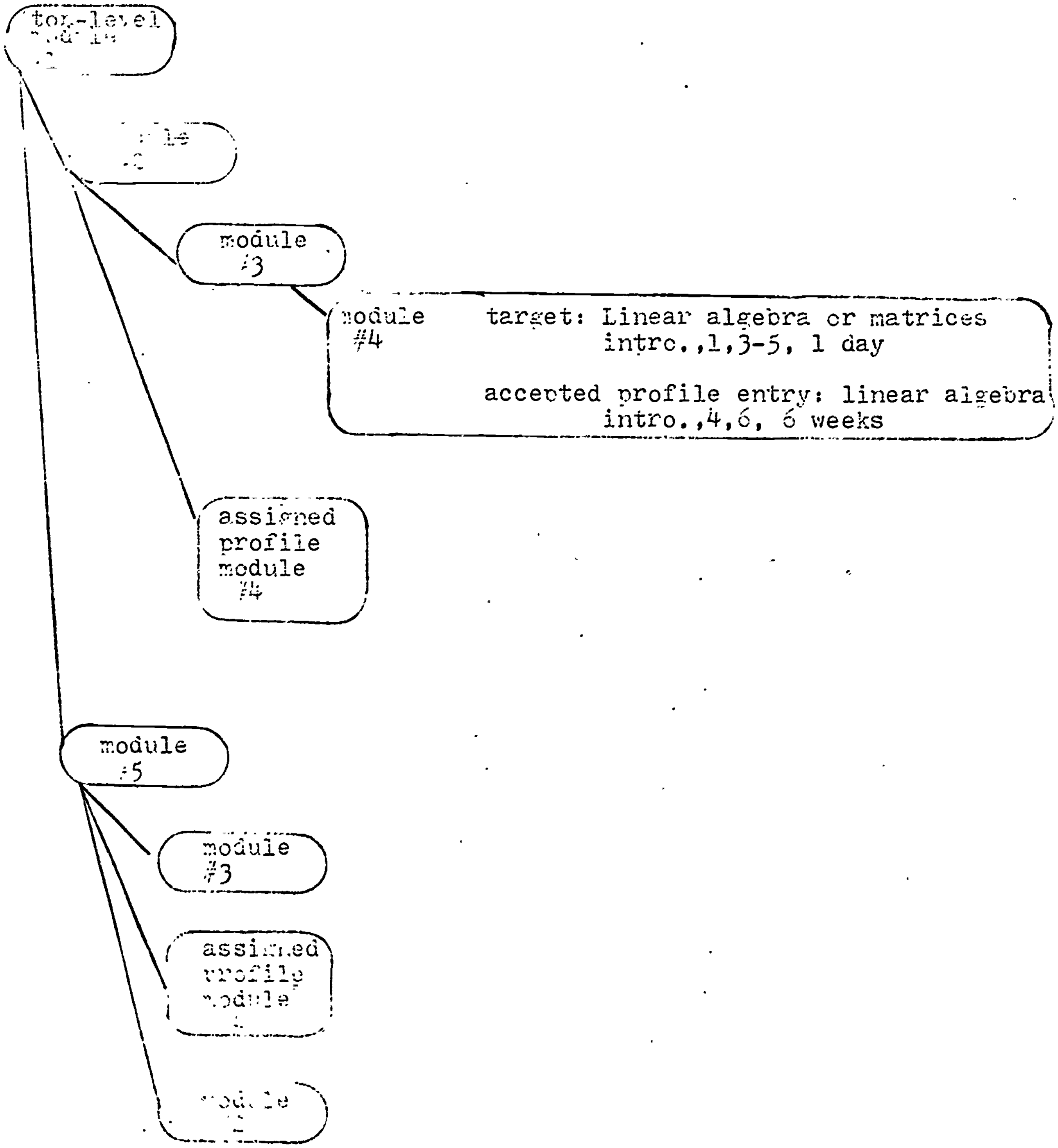


Figure 4a

Expansion of Goal Using Accepted Profile Entry

Top Goal: Quantitative Methods Including Linear Optimization

```
(0 0 *1* (0 [illegible] %%4101 (LINEAR-PROGRAMMING) ((AREA LINEAR-PROGRAMMING) (LEVEL . 1) (TIME ([illegible] 1 [illegible]))
[illegible] (MEDIA (STRUCTURE)))))
    ([illegible] *1* ((AREA (INCL LINEAR-PROGRAMMING DUALITY)) (LEVEL . 1) (TIME (2 2 H) (4 4 H)) (MASTERY [illegible]) [illegible]
        ([illegible] (0 [illegible] [illegible] ((INCL LINEAR-PROGRAMMING DUAL-SOLUTIONS)) ((AREA (INCL LINEAR-PROGRAM
)) ([illegible] . 1) (TI[illegible] ([illegible] H) (5 5 H)) (MASTERY 1) (MOTIVE 3 5) (MEDIA (BOOK)))))
         ([illegible] *1* ((LEVEL . 2) (AREA (INCL LINEAR-PROGRAMMING SIMPLEX-METHOD)) (TI[illegible] (2 2 H) (4 4 H)) (MA
 2 [illegible])))
            ([illegible] *1* (0 [illegible] %64190 ((INCL LINEAR-PROGRAMMING SIMPLEX-METHOD)) ((LEVEL . 2) (AREA (INCL
 SIMPLEX-METHOD)) (TIME (2 2 H) (4 4 H)) (MASTERY 1) (MOTIVE 2 6))))
         ([illegible] *1* ((AREA (INCL LINEAR-PROGRAMMING PROBLEM-FORMULATION)) (LEVEL . 1) (TIME (1 1 H) (2 2 H)
 (MOTIVE 2 6)))
            ([illegible] *1* (0 ASSIGNED [illegible]%%C%2 ((INCL LINEAR-PROGRAMMING PROBLEM-FORMULATION)) ((AREA (INCL LIN
 PROBLEM-FORMULATION)) (LEVEL . 1) (TIME (1 1 H) (2 2 H)) (MASTERY 1) (MOTIVE 2 6) (MEDIA (BOOK)))))
              (0 0 *1* ((AREA (OR LINEAR-ALGEBRA MATRICES)) (LEVEL . 1) (TIME (4) 1 H) ([illegible])) (MASTER
 4 5)))
                (0 0 *1* (1 CLAIMED %64189 (LINEAR-ALGEBRA) ((AREA LINEAR-ALGEBRA) (LEVEL . 1) (TIME (2
 4) (MOTIVE 4)))))
         ([illegible] *1* ((AREA (OR LINEAR-ALGEBRA MATRICES)) (LEVEL . 1) (TIME (40 40 H) (60 60 H)) (MASTERY 2
 ))
            (0 0 *1* (1 CLAIMED %64189 (LINEAR-ALGEBRA) ((AREA LINEAR-ALGEBRA) (LEVEL . 1) (TIME (240 6 [illegible]
 (MOTIVE 4)))))
      ([illegible] *1* ((AREA (OR LINEAR-PROGRAMMING PRIMAL-SIMPLEX-METHOD)) (LEVEL . 1) (TIME (2 2 H) (4 4 H)) (MA
 4 4)))
        (0 0 *1* (0 ASSIGNED [illegible]%%C44 ((INCL LINEAR-PROGRAMMING SIMPLEX-METHOD)) ((AREA (INCL LINEAR-PROGRA
)) (LEVEL . 1) (TIME (2 2 H) (4 4 H)) (MASTERY 1) (MOTIVE 2 6) (MEDIA (BOOK)))))
         ([illegible] *1* ((AREA (INCL LINEAR-PROGRAMMING PROBLEM-FORMULATION)) (LEVEL . 1) (TIME (1 1 H) (2 2 H
 (MOTIVE 2 6)))
            ([illegible] *1* (3 ASSIGNED [illegible]%%C%2 ((INCL LINEAR-PROGRAMMING PROBLEM-FORMULATION)) ((AREA (INCL LI
 PROBLEM-FORMULATION)) (LEVEL . 1) (TIME (1 1 H) (2 2 H)) (MASTERY 1) (MOTIVE 2 6) (MEDIA (BOOK)))))
         (0 0 *1* ((AREA (OR LINEAR-ALGEBRA MATRICES)) (LEVEL . 1) (TIME (40 40 H) (60 60 H)) (MASTERY 2
 ))

            (0 0 *1* (1 CLAIMED %64189 (LINEAR-ALGEBRA) ((AREA LINEAR-ALGEBRA) (LEVEL . 1) (TIME (240 6
 (MOTIVE 4)))))
```

Figure 4b

Program Printout of Curriculum in Figure 4a handle very complex area-parts to 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

•
•
•

A more complete example of syntactic expansion is given in Figure 5. 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)
including primal simplex method

•
•
•

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

Figure 5

Expansion of (LP or OR) including Simplex Method

```
ENINCL*1 (3 (OR OPERATIONS-RESEARCH SIMPLEX-METHOD))
ENINCL*1 (3 (OR SIMPLEX-METHOD OPERATIONS-RESEARCH))
ENINCL*1 (1 SIMPLEX-METHOD)
SEARCHTOP*4*RECRD (5 (OR SIMPLEX-METHOD (INCLUDING OPERATIONS-RESEARCH LINEAR-PROGRAMMING)))
SEARCHTOP*4*RECRD (5 (OR (INCLUDING OPERATIONS-RESEARCH LINEAR-PROGRAMMING) SIMPLEX-METHOD))
SEARCHTOP*4*RECRD (5 (INCLUDING (INCLUDING OPERATIONS-RESEARCH LINEAR-PROGRAMMING) SIMPLEX-METHOD))
SEARCHTOP*4*RECRD (5 (INCL (INCLUDING OPERATIONS-RESEARCH LINEAR-PROGRAMMING) SIMPLEX-METHOD))
SEARCHTOP*4*RECRD (5 (OR SIMPLEX-METHOD (INCLUDING LINEAR-PROGRAMMING OPERATIONS-RESEARCH)))
SEARCHTOP*4*RECRD (5 (OR (INCLUDING LINEAR-PROGRAMMING OPERATIONS-RESEARCH) SIMPLEX-METHOD))
SEARCHTOP*4*RECRD (5 (INCLUDING (INCLUDING LINEAR-PROGRAMMING OPERATIONS-RESEARCH) SIMPLEX-METHOD))
SEARCHTOP*4*RECRD (5 (INCL (INCLUDING LINEAR-PROGRAMMING OPERATIONS-RESEARCH) SIMPLEX-METHOD))
SEARCHTOP*4*RECRD (5 (OR SIMPLEX-METHOD (INCL OPERATIONS-RESEARCH LINEAR-PROGRAMMING)))
SEARCHTOP*4*RECRD (5 (OR (INCL OPERATIONS-RESEARCH LINEAR-PROGRAMMING) SIMPLEX-METHOD))
SEARCHTOP*4*RECRD (5 (INCLUDING (INCL OPERATIONS-RESEARCH LINEAR-PROGRAMMING) SIMPLEX-METHOD))
SEARCHTOP*4*RECRD (5 (INCL (INCL OPERATIONS-RESEARCH LINEAR-PROGRAMMING) SIMPLEX-METHOD))
SEARCHTOP*4*RECRD (5 (OR SIMPLEX-METHOD (INCL LINEAR-PROGRAMMING OPERATIONS-RESEARCH)))
SEARCHTOP*4*RECRD (5 (OR (INCL LINEAR-PROGRAMMING OPERATIONS-RESEARCH) SIMPLEX-METHOD))
SEARCHTOP*4*RECRD (5 (INCLUDING (INCL LINEAR-PROGRAMMING OPERATIONS-RESEARCH) SIMPLEX-METHOD))
SEARCHTOP*4*RECRD (5 (INCL (INCL LINEAR-PROGRAMMING OPERATIONS-RESEARCH) SIMPLEX-METHOD))
SEARCHTOP*4*RECRD (5 (OR SIMPLEX-METHOD (OR OPERATIONS-RESEARCH LINEAR-PROGRAMMING)))
SEARCHTOP*4*RECRD (5 (OR (OR OPERATIONS-RESEARCH LINEAR-PROGRAMMING) SIMPLEX-METHOD))
SEARCHTOP*4*RECRD (5 (INCLUDING (OR OPERATIONS-RESEARCH LINEAR-PROGRAMMING) SIMPLEX-METHOD))
SEARCHTOP*4*RECRD (5 (INCL (OR OPERATIONS-RESEARCH LINEAR-PROGRAMMING) SIMPLEX-METHOD))
SEARCHTOP*4*RECRD (5 (OR SIMPLEX-METHOD (OR LINEAR-PROGRAMMING OPERATIONS-RESEARCH)))
SEARCHTOP*4*RECRD (5 (OR (OR LINEAR-PROGRAMMING OPERATIONS-RESEARCH) SIMPLEX-METHOD))
SEARCHTOP*4*RECRD (5 (INCLUDING (OR LINEAR-PROGRAMMING OPERATIONS-RESEARCH) SIMPLEX-METHOD))
SEARCHTOP*4*RECRD (5 (INCL (OR LINEAR-PROGRAMMING OPERATIONS-RESEARCH) SIMPLEX-METHOD))
SEARCHTOP*4*RECRD (3 (OR SIMPLEX-METHOD OPERATIONS-RESEARCH))
SEARCHTOP*4*RECRD (3 (OR OPERATIONS-RESEARCH SIMPLEX-METHOD))
SEARCHTOP*4*RECRD (3 (INCLUDING OPERATIONS-RESEARCH SIMPLEX-METHOD))
SEARCHTOP*4*RECRD (3 (INCL OPERATIONS-RESEARCH SIMPLEX-METHOD))
SEARCHTOP*4*RECRD (3 (OR SIMPLEX-METHOD LINEAR-PROGRAMMING))
SEARCHTOP*4*RECRD (3 (OR LINEAR-PROGRAMMING SIMPLEX-METHOD))
SEARCHTOP*4*RECRD (3 (INCLUDING LINEAR-PROGRAMMING SIMPLEX-METHOD))
SEARCHTOP*4*RECRD (3 (INCL LINEAR-PROGRAMMING SIMPLEX-METHOD))
SEARCHTOP*4*RECRD (1 SIMPLEX-METHOD)
PARAM*4*SETV (OPERATIONS-RESEARCH)
PARAM*4*SETV ((INCL LINEAR-PROGRAMMING DUAL-SOLUTIONS))
PARAM*4*SETV ((INCL LINEAR-PROGRAMMING SIMPLEX-METHOD))
PARAM*4*SETV ((INCL LINEAR-PROGRAMMING GEOMETRICAL-SOLUTIONS))
PARAM*4*SETV ((INCL LINEAR-PROGRAMMING PROBLEM-FORMULATION))

PARAM*4*SETV ((INCL LINEAR-PROGRAMMING PRIME-SOLUTIONS))
[illegible] (LINEAR-PROGRAMMING)
```

on these two modules, choosing the obviously best choice of the two, "linear programming including simplex method". It proceeds to create the rest of the curriculum as given in Figure 6.

In another test, the system considered the general goal area of "operations research", generating the subgoals as given in Figure 7.

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, all other things being equal, in addition to handling the complicated cases where some of each of the parameters 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 presently operational and is calculated by the system.

Even without the complete outputs of all these tests however, the few tests described above confirm the feasibility of the design proposed for an education assembly system for student-executed educational design. In addition, the feasibility of encoding and collecting cognitive maps has been shown. As an example top-level target goal:(linear programming or operations research) including simplex method/introductory,2,7,. day

assigned module #1: linear programming including simplex method/ introductory,1,2-6, ½ 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, 1-1½ weeks

no modules acceptable: assign dummy module #3

subgoal: linear algebra/intorductory.,2-3,4-9,1½week

accepted module #3

Figure 6

Satisfaction of top-level goal

Operations-research/1,1,3-5,1 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,$\frac{1}{2}$ day
  - transportation-problem INCLUDING prime-solutions/1,1,3-5,$\frac{1}{4}$ day
  - transportation-problem INCLUDING dual-solutions/1,1,3-5,$\frac{1}{4}$ 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 d.
- 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, $\frac{1}{2}$ day
  - scheduling-problems INCLUDING problem-formulation/1,1,3-5,$\frac{1}{4}$ day
  - scheduling-problems INCLUDING solution-ideas/1,1,3-5, $\frac{1}{4}$ day
- inventory-problems/1,1,3-5,1 day
  - inventory-problems INCLUDING problem-formulation/1,1,3-5,1/3-2/3 d.
  - inventory-problems INCLUDING solution-interpretations/1,3-5,1/3-2/3 d.
- operations-research INCLUDING applications/1,1,3-5,2 days
  - O.R. INCLUDING (applications AND inventory-problems) [illegible]
  - O.R. INCLUDING (applications AND decision-theory)/1,1, 3-[illegible],all, 4- hours
  - O.R. INCLUDING (applications AND critical-path-method) [illegible]
  - O.R. INCLUDING (applications AND linear-programming) [illegible]

Figure 7

[illegible] lower-level [illegible]

of a more elaborate curriculum available from encoded modules (not all of which have been entered into the system, giving consideration to space, cost, etc) see Figures 8 and 9.

- computer-science/1, 2-4, 5-8, 3 months
  - ogramming/1, 2-3, 6-8, 4-6 weeks
    - neral-programming/1, 2-3, 6-8, 3-4 weeks
      - prog. INC. languages/1, 2-4, 5-7, 1 1/2 weeks
      - prog. INC. implementation/1, 2-4, 4-7, 1 week
      - program design/1, 3-4, 5-7, 1 1/2 weeks
    - heuristic-programming/1 2-3, 5-8, 2-3 weeks
      - heuristic-prog. INC.methods/1 2-4, 4-8, 1 1/2 - 2 weeks
      - heuristic-prog. INC. heuristics/1 2-4, 4-8, 1/2 - 1 week
  - Operations-Research/1, 2-3, 6-8, 3-4 weeks
    - simulation/1, 2-3, 5-8, 1/2 week
      - simul. INC. discrete-events-models/1, 2-3, 4-8, 1 day
      - simulation-languages/1, 3-4, 5-8, 1 1/2 days
    - probability AND statistics/1, 2-4, 6-8, 1 week
      - prob. INC. density-functions/1, 3-4, 6-9, 1 1/2-2 days
      - prob. INC. distribution functions/1, 3-4, 6-9, 1 1/2-2 days
      - prob. INC. moments/1, 3-4, 6-9, 1 1/2-2 days
    - optimization/1, 1-3, 4-9, 1/2 week
      - opti. AND derivatives/1, 3-4, 5-8, 1 day
      - opti. AND (minima AND maxima)/1, 2-4, 5-8, 1/2-1 day
      - opti. AND Lagrangian/1, 2-4, 6-9, 1 day
    - linear-programming/1,2-4,6-8,1 1/2 weeks
      - linear-programming-problem/1 1-3, 4-7, 1/2 week
      - simplex-method/1, 2-4, 5-8, 1/2-2/3 week
      - duality-theory/1, 2-4, 5-8, 1/3-1/2 week
  - Logic/1, 2-3, 6-8, 3-4 weeks
    - propositional-calculus/1, 3-4, 6-9, 1 week
      - content module/1, 2, 4-8, 2/3-1 1/2 weeks
        - mathematics/1, 1, 1/3, 1 week
    - predicate-calculus/1, 2-4, 6-9, 2-3 weeks
      - content module/1, 2-,5-7,2-3 weeks
        - propositional calculus/1, 1,1-4, 2/3 2/3 week

Figure 8

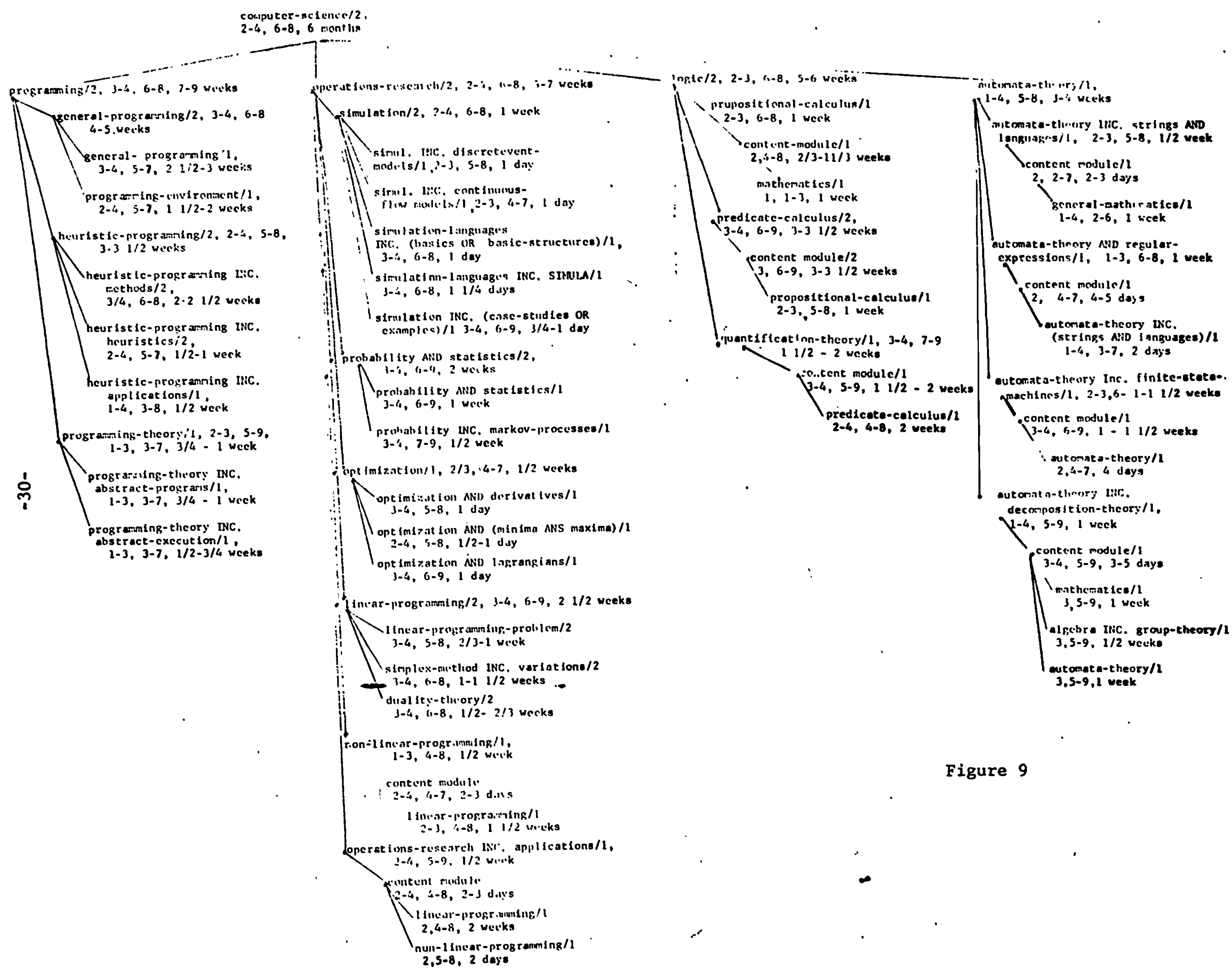

computer-science/2, 2-4, 6-8, 6 months
programming/2, 3-4, 6-8, 7-9 weeks
general-programming/2, 3-4, 6-8 4-5 weeks
general- programming/1, 3-4, 5-7, 2 1/2-3 weeks
programming-environment/1, 2-4, 5-7, 1 1/2-2 weeks
heuristic-programming/2, 2-4, 5-8, 3-3 1/2 weeks
heuristic-programming INC. methods/2, 3/4, 6-8, 2-2 1/2 weeks
heuristic-programming INC. heuristics/2, 2-4, 5-7, 1/2-1 week
heuristic-programming INC. applications/1, 1-4, 3-8, 1/2 week
programming-theory/1, 2-3, 5-9, 1-3, 3-7, 3/4 - 1 week
programming-theory INC. abstract-programs/1, 1-3, 3-7, 3/4 - 1 week
programming-theory INC. abstract-execution/1, 1-3, 3-7, 1/2-3/4 weeks
operations-research/2, 2-4, 6-8, 5-7 weeks
simulation/2, 2-4, 6-8, 1 week
simul. INC. discreteevent-models/1, 2-3, 5-8, 1 day
simul. INC. continuous-flow models/1, 2-3, 4-7, 1 day
simulation-languages INC. (basics OR basic-structures)/1, 3-4, 6-8, 1 day
simulation-languages INC. SIMULA/1 3-4, 6-8, 1 1/4 days
simulation INC. (case-studies OR examples)/1 3-4, 6-9, 3/4-1 day
probability AND statistics/2, 3-4, 6-9, 2 weeks
probability AND statistics/1 3-4, 6-9, 1 week
probability INC. markov-processes/1 3-4, 7-9, 1/2 week
optimization/1, 2/3, 4-7, 1/2 weeks
optimization AND derivatives/1 3-4, 5-8, 1 day
optimization AND (minima ANS maxima)/1 2-4, 5-8, 1/2-1 day
optimization AND lagrangians/1 3-4, 6-9, 1 day
linear-programming/2, 3-4, 6-9, 2 1/2 weeks
linear-programming-problem/2 3-4, 5-8, 2/3-1 week
simplex-method INC. variations/2 3-4, 6-8, 1-1 1/2 weeks
duality-theory/2 3-4, 6-8, 1/2- 2/3 weeks
non-linear-programming/1, 1-3, 4-8, 1/2 week
content module 2-4, 4-7, 2-3 days
linear-programming/1 2-3, 4-8, 1 1/2 weeks
operations-research INC. applications/1, 2-4, 5-9, 1/2 week
content module 2-4, 4-8, 2-3 days
linear-programming/1 2,4-8, 2 weeks
non-linear-programming/1 2,5-8, 2 days
logic/2, 2-3, 6-8, 5-6 weeks
propositional-calculus/1 2-3, 6-8, 1 week
content-module/1 2,4-8, 2/3-1 1/3 weeks
mathematics/1 1, 1-3, 1 week
predicate-calculus/2, 3-4, 6-9, 3-3 1/2 weeks
content module/2 3, 6-9, 3-3 1/2 weeks
propositional-calculus/1 2-3, 5-8, 1 week
quantification-theory/1, 3-4, 7-9 1 1/2 - 2 weeks
content module/1 3-4, 5-9, 1 1/2 - 2 weeks
predicate-calculus/1 2-4, 4-8, 2 weeks
automata-theory/1, 1-4, 5-8, 3-4 weeks
automata-theory INC. strings AND languages/1, 2-3, 5-8, 1/2 week
content module/1 2, 2-7, 2-3 days
general-mathematics/1 1-4, 2-6, 1 week
automata-theory AND regular-expressions/1, 1-3, 6-8, 1 week
content module/1 2, 4-7, 4-5 days
automata-theory INC. (strings AND languages)/1 1-4, 3-7, 2 days
automata-theory Inc. finite-state-machines/1, 2-3,6- 1-1 1/2 weeks
content module/1 3-4, 6-9, 1 - 1 1/2 weeks
automata-theory/1 2,4-7, 4 days
automata-theory INC. decomposition-theory/1, 1-4, 5-9, 1 week
content module/1 3-4, 5-9, 3-5 days
mathematics/1 3,5-9, 1 week
algebra INC. group-theory/1 3,5-9, 1/2 weeks
automata-theory/1 3,5-9,1 week

Figure 9

EVALUATION PROJECTS

The ultimate goal of all of ERDU's projects is to increase educational efficiency. In the case of the work on the EAS, we have taken the route of actual construction of a system that might contribute to such a goal in appropriate educational environments. Other ERDU projects have been directed toward the efficiency goal through different paths.

Our explorations of the use of television technology have resulted in a preliminary survey in the Pittsburgh region of the demand for a "televised masters degree" jointly offered by GSIA and Carnegie Institute of Technology. Preliminary results seem to indicate that there is not enough local demand to warrant our pursuing this any further. Our in-house use of videotape equipment has increased to the point where almost all of our students have been involved in using the medium in some way for educational purposes. However, we now believe that no major break-throughs will result from this sort of technology within the conventional academic structure.

The study of departmental productivity has lead to some preliminary results and has suggested some system changes. The investigation of the sensitivity of research productivity indices to various assumptions about aggregation procedures has demonstrated that any index that is based upon a reasonable set of assumptions will correlate highly with most others. In particular, an index that simply counts total number of papers published ranks faculty in almost the same order as one that corrects for rate, quality of journal and pages.

The general study, of which research productivity was only one component, also required measures of teaching quality, entering student quality and final student quality. During the course of the study it became evident that

CMU did not maintain the records (at least in any accessible form) that would facilitate the investigation of questions of this sort. The University Administration is now designing a new management information system that will keep track of not only the costs of running the university but also the quality of its products: research and education.

PARTICIPANTS

The principal participants in ERDU this year were Professor D. Klahr, Director, S. Evans, Research Associate, and C. Dermody, Senior Systems Programmer. Professors C. Kriebel, G. Thompson and R. Weil assisted in module encoding as did several of our Ph.D. students. J. Bloom supervised a small staff of Masters students in the use of our video tape facility.

APPENDICES

Appendix A. Module Encoders Guide

Appendix B. Content Module Encoding Instructions

Appendix C. Structure Module Encoding Instructions

MODULE ENCODER'S GUIDE: DESCRIPTION OF YOUR TASK

We wish to enlist your expertise in the creation of a data base for an Educational Assembly System (EAS). In this Encoder's Guide we will first describe a few features of the design and operation of the EAS. Then we will describe in detail the task we have set for you and the procedures we would like you to follow.

System Design and Operation

The purpose of the EAS is to generate, upon demand, highly individualized curricula for a wide range of students with diverse backgrounds, characteristics and educational objectives. The ultimate design goal is to simulate an omniscient, perceptive and indefatigable human educational consultant and curriculum designer. The intended user of the system, a student, describes his objectives in terms of a goal statement. Then the system selects - from a large collection of previously encoded educational resources - a sequence of educational modules. These modules are descriptions of educational subgoals whose achievement will satisfy the student's goal. In most cases, the subgoals have their own sub-subgoals, etc. The higher level subgoals are often described using structure modules, which lay out the relationships between the broad areas of knowledge that are related to the student's stated goal. The "low level" goals are more often achieved through content modules; they correspond to specific educational activities (reading parts of books, solving problem sets, taking courses, etc). We shall return to a fuller description of both structure and content models in a later section.

The system requires several kinds of information. First it needs to know about the student: his goals, his background, his preferences, etc. Such information is supplied directly by the individual user during his initial interaction with the system, and it will not concern us further here. Second, it needs to know about the structure of knowledge in the areas for which it is creating a curriculum. For example, it needs to know that the area of linear programming has, as subareas, material on objective functions, on the simplex method, etc., Part of your task as an encoder is to precisely specify your view of these relationships or subgoals in your area of expertise. This Guide will indicate how you can do this in a manner that is intelligible to the EAS through the encoding of modules.

Another kind of information needed by the system pertains to the "meaning" of the terms used by you, by other expert encoders, and by the (typically naive) student. For example, it needs to know that (for some experts) "operations research" is roughly synonomous with "quantitative methods", that it is a subset of "applied math" and "management science", and that "dynamic programming", "linear programming", "statistics", and "queuing theory" are all subsets of it. You will be asked to supply this information for each of the descriptive terms you use. The system will then construct a semantic net by building a file of all such relationships provided by you and the other encoders. The distinction between subgoals and the semantic net will become evident as you begin to actually encode modules.

Your task: overview

As indicated above, the student communicates with the system by specifying, among other things, his educational goal. The system is designed to use goal specifications in another way: in the description of modules. Both content modules and structure modules are encoded in terms of the goals that they satisfy. Your task is to (1) create and encode a module that will satisfy a high-level goal [specified by the system designers] and (2) to encode the subgoals generated [by you] in doing (1). This process may continue for several levels. In order to do (1) and (2), you need to know more about the language in which goals are to be written and about the other information required to encode a module.

Goal statements

To make the task clear, we shall briefly describe a goal statement. (More detail about goals will be provided later). A goal has five parts:

(1) The area gives the subject or knowledge domain that the module considers (e.g., "computer science" or "economics").

Terms may be combined with the operators AND, OR, EXCLUDING, and INCLUDING (to be explained shortly). Parentheses should be used to avoid ambiguity. Some examples of area expressions are "(computer-science INCLUDING programming) AND (mathematical-logic)", "microeconomics OR (macroeconomics INCLUDING public-policy-economic)", "psychology EXCLUDING (math AND statistics)", etc.

The last two goals use the terms INCLUDING and EXCLUDING. The system differentiates these two terms from AND and NOT by its actions in the search for suitable modules. The goal of "psychology EXCLUDING (math AND statistics)" requests or strongly prefers psychology with no math or statistics. If there were only one possible module to choose, and it included some "math AND statistics", then the search algorithm would "relax" the constraints sufficiently to allow this module to suffice, for this case. Thus, EXCLUDING is a weak form of NOT. Similarly, INCLUDING is a weak form of AND. "Psychology INCLUDING math" similarly requests the two, but the system may relax its search, if it is unable to meet the added demand of math. Observe that "psychology AND math" requests both areas in a module. This is not their separate union but rather an intersection or integration of the two.

Note the lack of verbs, prepositions, and modifiers. The <u>area</u> part of the goal is <u>not</u> a description. Rather it is an indicator of the subject material's <u>name.</u> Hence, one must convert such phrases as "wiring of the machine" to "machine-wiring". "Setting up L. P. programs" is converted to "L.-P. INCLUDING formulation". (If necessary, phrases are to be converted to new hyphenated forms as "making of transistors" becomes "transistor-making"). In Figure 1, we have listed appropriate area encodings for several topics.

(2) The <u>level</u> specifies one of three levels of treatment: introductory, intermediate, or advanced. Thus, we may have "economics/introductory" or "programming AND computer science/advanced". The level, given after the "/", modifies everything to the left of "/".

(3) The [illegible] level gives an estimate of the level of accomplishment expected by the [illegible] in [illegible] the module. Level 0, the lowest mastery level is associated with the desire to simply advance one's general knowledge in the area.

Some Examples of Encoding

Topic	Area-part of GOAL
"the art of programming"	programming
"learning to program"	programming
"how to evaluate a canned program with respect to suitability of needs"	canned-routines INCLUDING evaluation
"functions of several variables"	complex - variables
"management science applications of linear programming"	linear-programming INCLUDING management-science-applications or management-science-applications INCLUDING linear-programming (depending on intended emphasis)
"mathematical approaches to decision-making in management	management-science INCLUDING (mathematics AND decision-making)
"use of Lagrangian multipliers"	(optimization-techniques INCLUDING Lagrangian-multipliers) INCLUDING examples
"programming robots"	complex-information-processing INCLUDING robots or artificial-intelligence INCLUDING robots (depending on intent of encoder)
"describing decision making"	probability - theory INCLUDING decision-trees or decision-analysis INCLUDING decision-flow-diagram (depending on intent)
"[illegible]; efficient inventory control strategies"	inventory-control or inventory-control-techniques (depending on intent)

Figure 1

This is roughly equivalent to a pass in a pass/fail course. Level C is one of minimal satisfactory proficiency. It assumes the student can work with or manipulate the material, ideas, etc., or in our analogy, made a C in the course or module. Level B suggests that the student is under full control of the area when done with the module. He can fully manipulate and handle the material, doing the bulk of all typical exercises, answering or responding correctly to most of the issues, etc. This corresponds to a B in a course. Level A, the highest level, suggests the student is "master" of the material. This includes the ability to study or advance by himself or to handle new material in the area. This is analogous to making an A in a course.

(4) The motive represents a estimate of the appropriate degree of seriousness of a student who might use this module. This is given as a range (between 1 and 9 with 1 the lowest, 9 the highest). An example might be "1-3" which suggests a somewhat low motivation in pursuing this area. The range 8-9 suggests a very high degree of seriousness in pursuit of this module. A module with 1-9 indicates that the module is appropriate for any student, regardless of his motivation in pursuing that subject matter.

(5) Finally, the last part of the goal is an estimate of the amount of time allowed for completion of the module. This estimate may be given as a range, in any units appropriate, assuming the student will spend full-time pursuing the goal, exclusively. As an example, consider the full goal: programming/introductory, C-B 7-9, 2 months. This indicates that the student wishes to learn programming at an introductory level, requires that such material impart a mastery level between C and B, and allows a total time of two months for completion. As an encoder, if you were given such a goal, your job would be to specify educational activities for the student that would satisfy this goal.

Structure [illegible]

Given a student's goal, the module's purpose is to enable the student to fulfill his goal. Thus, each module represents a possible goal that a student may have, and to encode a module is to encode a (possible) goal.

[illegible] with [illegible] material. In such cases particular resources are retrieved to satisfy the goal. Thus the "prerequisites" section of the [illegible] these resources?"

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 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 whole accomplishment will result in the original goal being fulfilled. Hence, the goal of understanding the material in this module impliesfulfilling 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.

One might hypothesize that any particular module may be in fact a "mixed case", part content and part structure. This is not the case, however; the mixed case can be decomposed as follows. Assume that an expert wishes to encode the goal "introduction to operations research". He may feel that the student should read the first two chapters of Aronofsky's book, the first half of Feller's book on probability and get an introductory understanding of simulation, linear programming, and statistics. This sequence of actions, which would fulfill the original goal, should be expressed only as a set of subgoals to be fulfilled. Thus for the goal concerned with operations research/introductory, the encoder might require the student to study problem-formulation in the O. R. area, get an introductory understanding of probability, get an introduction to simulation, get an introduction to linear programming, and get an introduction to statistics. These five subgoals are the basis for mastering the goal. In turn, the subgoals above are then treated separately, each as a subgoal to be encoded by itself. To satisfy the first subgoal above, the expert may encode the subgoal by giving the retrieval of Aronofsky's book as <u>one</u> possible satisfaction to this subgoal. Similarly, for the other subgoals. For example, satisfying simulation (subgoal 3) may require knowledge about SIMULA, as well as reading Balintfy's book. Hence, this subgoal is encoded again as a structure module, requiring (1') introductory knowledge about simulation languages and (2') simulation applications. Then (1') is encoded with one particular solution, namely SIMULA, while (2') is also encoded with one possible solution.

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. We particularly do not want SIMULA <u>tied</u> to simulation. Rather we want it tied to simulation languages. We do not want to go into the structure of the varied modules that use SIMULA to update them.

<u>Prerequisite Specification</u>

As explained above for a structure module, the encoder must specify a set of subgoals, the completion of which will satisfy the original goal. More precisely, this is a specification of the sub-domains that comprise the subject matter in the area-part of the goal. It is analogous to a book's section headings, under a chapter that dealt with the area-part of the goal,

It is only a disection of the material itself; it is not a complete elaboration of every piece of material that would have to be specified such that the goal would be completed. In this sense, subgoals are <u>not</u> the same as prerequisites. (Indeed, to give all prerequisites, one would have to specify a reading capability, versatility in English, etc.) That is, all conceivable prerequisites need not be specified as subdomains of the area-part given of the goal. Only the proper subdivisions that bear directly on the area are given; it is up to the expert to evaluate and properly pick such a decomposition. Naturally there will be some variance among experts. This is one reason why more than one expert will encode the same areas.

The place where preliminary prerequisites do get included is in the content modules. When materials are actually retrieved in order to satisfy the subgoals, the need arises for the preliminary prerequisites that will enable the student to handle these materials. In the portion of the content module form entitled "prerequisites", such requirements are given. For example, assume that the student's goal is linear-programming/introductory, C, 6-7, 3 weeks.

Then the encoder must decide on the basic areas of an introduction to linear programming, given the other constraints in the goal. Let us assume he specifies the following subgoals:

(1) objective-functions/introductory, C-B, 6-7, 1 week.

(2) simplex-method/introductory, C-B, 6-7, 1 week

(3) the-dual/introductory C-B, 6-7, 1 week

The three subgoals above would be his choice for the decomposition of the goal. However, left unspecified are the prerequisites of sufficient calculus to understand (1) and enough linear algebra to comprehend (2) and (3).

These prerequisites however are not really an integral part of linear programming *per se.* Thus they do not appear in (1) - (3). [Perhaps some would say that some parts of calculus *are* to be included in the area of linear programming. Then they may have added a fourth subgoal of calculus with a time of four months. This is permissible, if the encoder decides so. Needless to say that he will have not encoded a structure module that would fulfull the original goal given, because of time. Nonetheless it is a valid module. In such a case it simply would represent the best the encoder was able to do with a time overflow in this case of four months.] So let us assume that the three subgoals given are felt to be a proper set. When the subgoal (1) objective functions/introductory is being encoded, the encoder may either choose to decompose that further or choose to specify a content module. Assume that he chooses a content module. Then he indicates some educational resource that adequately communicates the concept.

Perhaps it is the first section of the second chapter of the book Linear Programming. The question posed in the encoder's form is what prerequisites are needed to handle this topic. It is here that the encoder specifies the need for basic calculus. It is at this stage that he knows precisely what is needed, for he knows precisely what kinds of material the student will be studying. Actually one may notice at this stage that it is not calculus at all that is needed. Rather it is an introduction to the simplier concept of "functions". Hence the encoder would be expected to specify this.

Then again , perhaps the encoder has picked a different book written especially for laymen. And this concept is excellently explicated without the notion of any higher mathematics; only high school algebra is needed. Then the prerequisites are so posed. Indeed the actual preliminary prerequisites can not be totally clear unless the actual material to be used is specified.

As another example, imagine the goal of a two week introduction to relativity. The encoder who puts tensor calculus as a prerequisite automatically, is quite mistaken; the expert who chooses the content module may well pick a reformulation that requires nothing more than high school algebra (though an analytical mind, in additon). Hence such preliminary prerequisites come into play when the actual materials are specified. Note that the preliminary prerequisites specified at such time may well be other goals, for example:

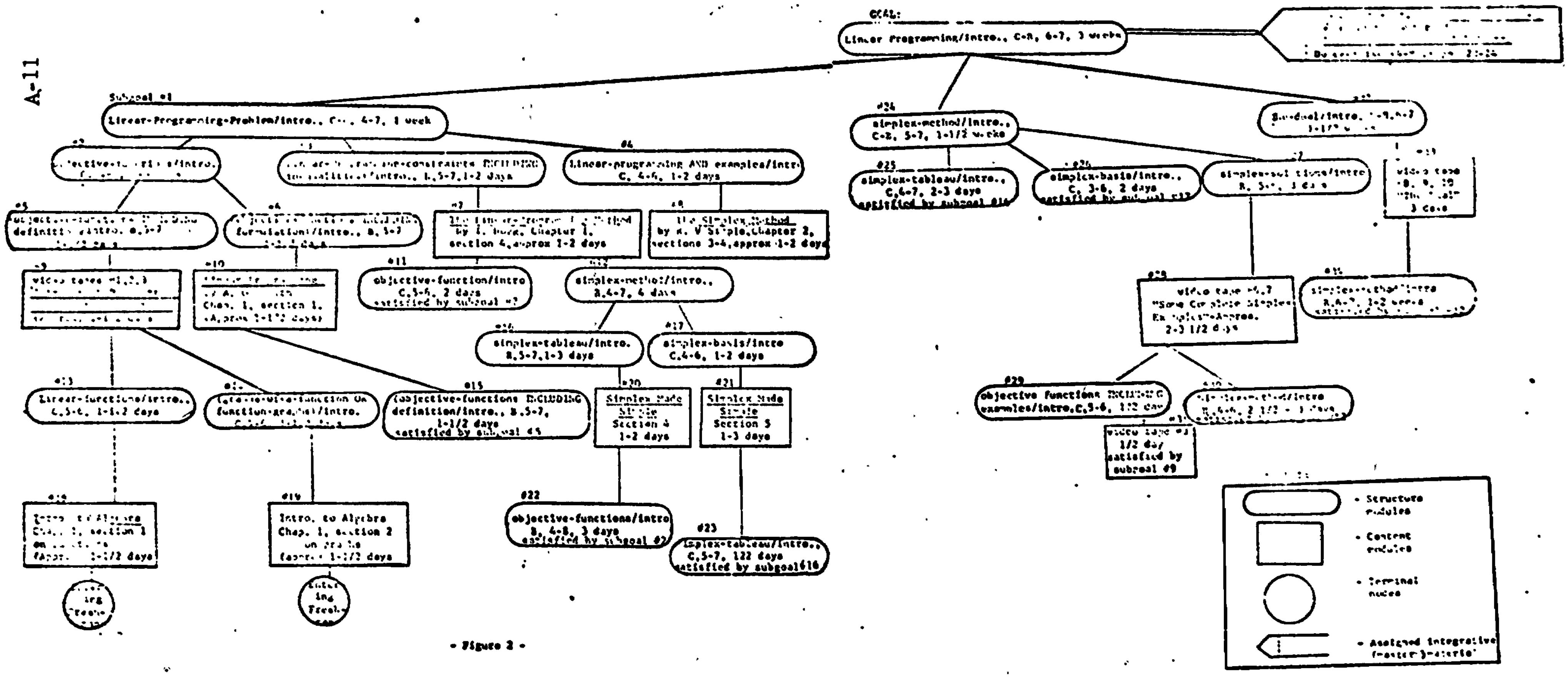
calculus/introductory, C, 6-8, 2 months

These goals in turn are re-entered into the system and are handled like any other goal.

Note that the encoder may (and frequently will) alternate between structure and content modules. The original goal may be completely handled by some book he knows of. Thus the initial encoding is one of content, not structure. However, after specifying such a book, he may wish to specify several subgoals that are necessary in order that that book can be handled. These may include several general goals described in GOAL. These in turn are entered into the system and may be further broken down by encoded structure modules (or content modules, yet again). Typically at the broader (or "higher level") knowledge domains, the modules will all be structure ones. At the lowest levels (extremely specific topics), it will tend to be primarily content modules. However, this is not a requirement.

Summary

In general, the encoder must decide whether the area-part of the goal, in context of the total goal, is best encoded by a structure or content module. (At the highest levels, he may be requested to employ only structure modules). In the case of such structure modules, he will be asked to define the primary subareas that comprise the domain of interest, given the time limitations, etc. These subtopics that define the curriculum for the student are expressed as subgoals to be accomplished. If the encoder then handles each subgoal in turn, which he may be called upon to do, he cycles again through this process. At such time as he feels that a particular educational resource precisely fits the needs of a sub-...-sub-goal under consideration, a content module is encoded. In encoding such a content module, he must specify the preliminary prerequisites so that the student can handle this material. This is given as subgoals that must be fulfilled or met before there is a reasonable expectation that the material can be effectively studied and mastered. In general the encoder should strive to choose structure modules over content modules since this makes the area being developed less dependent upon specific educational resources. This point can be further appreciated by contrasting the encoding of the area "management science" by listing 20 courses that must be taken, versus listing 20 subgoals that must be fulfilled, each in turn requiring, say, 5 sub-sub-goals, each one of which is fulfillable by either some chapter, some short series of lectures (perhaps on video tape), journal articles, or other educational media. Of course since content modules can have prerequisites that lead to other structure or content modules, the curriculum will vary with both components. A hypothetical curriculum is given in figure 2, as it evolved for the particular student's goal as shown. In the curriculum, the various kinds of modules are outlined. The terminal nodes in the figure represent each point at which the curriculum matches the current status or capability of the student. Thus these are all the points at which he may begin his study. Because of space limitations, associated mastery material is only shown for a couple of modules. However, each module would have such a mastery section.



- Figure 2 -

Hypothetical Curriculum resulting from Expansion
of the goal:" Linear Programming/Intro.," C-8, 6-7, 3 weeks

Further Specifications in Encoding: Mastery and Time

Mastery. When encoding a module, the expert is asked to specify some evaluation or verification procedure that will enable the student to confirm his mastery at the specified level in the goal. Such a determination of educational resources or activities that adequately reflect the accomplishment of the appropriate mastery level involves a double consideration. The first and obvious one is what materials or tasks constitute a "test" or evaluation. The second and implicit consideration concerns what is to be tested. In the case of content modules, this second consideration is clear. The test is to concern the material referenced in the module itself. Left to be specified is the activity that constitutes a test or evaluation of the required mastery level for that module. In the case of structure modules, the material to be tested is partially obscured by the fact that the module is composed of sub-goals. However the mastery of the module is not of the subgoals; when they are encoded, the mastery of each one is left to the encoder of each subgoal. Rather the mastery involves mastery of the "node" at which the encoder is working. Thus the evaluation of mastery (at the specified level) requires evaluation of accomplishment at the main node itself.

For structure modules, it is sometimes the case that what the student is to get out or derive from the material contained in the subgoals is strictly greater than the sum of the parts. That is, there may be an integrating activity that unifies the separate subgoals which is an understanding over and beyond the content of each of the subgoals. It is often such an integration that constitutes the goal of the module itself. Hence in those cases where the integration of the subgoals does not occur during the pursuit of the subgoals, such integrating activity will occur as the student attempts to demonstrate and evaluate his mastery of the node itself. Such integrating activity or material or resources are specified in the mastery section of the encoding form.

The first portion of the specification of the mastery section may include a reference to some education resource that attempts to integrate and coordinate the disparate subgoals that the encoder specified. Then, after such material is recorded, the encoder specifies some task or test or procedure whereby the student may evaluate his mastery of the node with which he is currently involved. On the other hand, if in doing the subgoals, the student will achieve the overview or unification as he progresses through the subgoal list, such additional materials need not be assigned in the first part of the mastery portion for the module. The choice and decision is up to the encoder.

Thus the mastery portion is the section of the encoding format that contains not only descriptions of evaluative procedures, but also provides the opportunity (and in some cases the necessity) to assign explicit educational resources or activities that operate at the level of the goal itself (rather than any subgoal level). By way of analogy, recall that on final exams, professors will sometimes describe some short, comprehensive situation, involving an integrative component among the sub-issues the student has studied. Then questions are asked about it. This is only partially analogous to the mastery section, of structure modules. Since we do not have the same time and space limitation, (nor, often the same motivation), we shall replace the final exam's briefly described situation of integration. In its place are references to something more akin to the last section of a book's chapter that attempts to pull all the sections of the chapter together. Associated with this assigned integrating activity are the evaluation procedures (or integrative final exam questions in our analogy). In this respect, of course, both content and structure modules are alike. As an example, we take the (partial) goal "programming AND statistics/intro".

Our structuring could take the forms as given in Figure 3(a). Here the intersection of these two areas is achieved in each of the subgoals. Mastery of the top goal requires only the union of the mastery of the subgoals. However, we might have taken the approach found in Figure 3(b). The student would have the building blocks from each of the two sub-domains; however no activity yet would integrate the two separate domains. Thus, in the mastery portion of the top node (original structure module), we would have to specify the educational resources that included seeing how programming is applied to statistics. Then demands would be made on the student to show his competence at some mastery level.

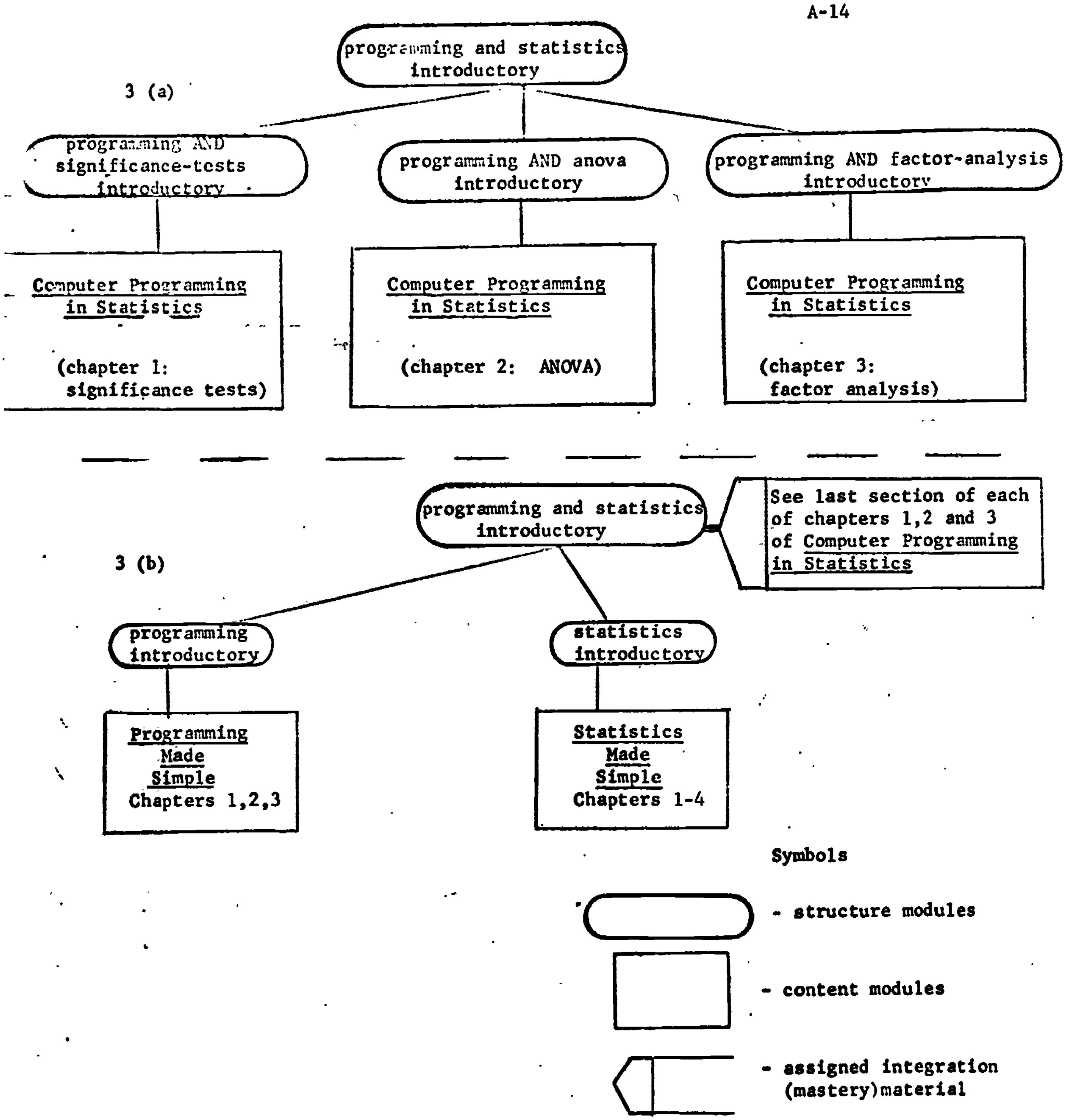


Figure 3

Similar curricula with and without integrative material

Time Recall that the student is assumed to be working full time exclusively on the particular goal being encoded (or each subgoal in turn). The estimate of time needed to complete a module includes the time needed to complete any integrative assignments given in the mastery section. Below are some approximations that relate university courses and terms to such full time effort.

a) A 3 credit course per semester is approximately equivalent to 4 weeks at full-time (or one month)

b) A 2 semester course equals 2 months.

c) A 4 credit course equals 5 1/2 weeks or 1 1/3 months.

d) A mini course at G.S.I.A. is equivalent to 2 1/2 weeks.

3) Six weeks of a normal term in a 3 credit course is equivalent to 1 1/2 weeks.

Some Other Summaries

Level: introductory - 1
intermediate - 2
advanced - 3

Mastery: Level 1 (Pass)
Level 2 (C)
Level 3 (B)
Level 4 (A)

Motive: 1 - lowest
⋮
9 - highest

GOAL: Area-part/Level, Mastery-Range, Motive-range, Time-Range.

Example: Computer-Science/Introductory, C-B, 6-7, 3-4 weeks.

- or -

"identically":

Computer-Science 1, 2-3, 6-7, 3-4 weeks.

(You may wish to tear this page off to keep as reference)

CONTENT MODULE ENCODING INSTRUCTIONS

0. Introduction

The purpose of these instructions is to aid you in filling out the encoding form. You will find both new information as well as summaries of some information that was in the Encoding Guide, which we assume that you have already read. In particular, we assume that you are familiar with a GOAL statement and understand the purpose of the prerequisite lists in content modules as well as the distinction between content structure modules. Checking each number and subsection of the encoding form with these instructions should facilitate your encoding task. In the upper left hand portion of each form, you will find a place to number (in any fashion) each module as well as give your name; in the right hand corner, you may refer to an earlier module, which will imply that this module is identical to the one so referenced, except for those sections that you fill in (which will override the sections in the referenced module). This is a "ditto" box, for your convenience, when you find you are encoding the same thing, or same section, over and over. The rest of the guide is self-explanatory.

1. Module NAME:

   a. An identifying name

      Choose a very simple name in natural language format to describe this module. (.g., Wilde's introductory OR book, "The Brain" film, Samuelson's economics book.)

   b. Abstract

      In a few sentences, give a brief description of the module.

   c. Condense the abstract above into one single, terse sentence.

   d. Area-part of GOAL

      Translate the above sentence into an area-part of a goal (in GOAL language). Review GOAL in the "Description of Your Task" document if necessary. Do not include a specification on Level; that will be treated separately.

   e. Semantics

      For <u>each</u> word (w) used in (d) above, give three lists. The first list (A_1) contains all words which imply w, or from which w is an immediate subpart or subset. In a knowledge domain, it is the category just above the category that contains w. The decision of A_1 is based on the use of w in (c), and hence, in turn on the content and context of the module being encoded.

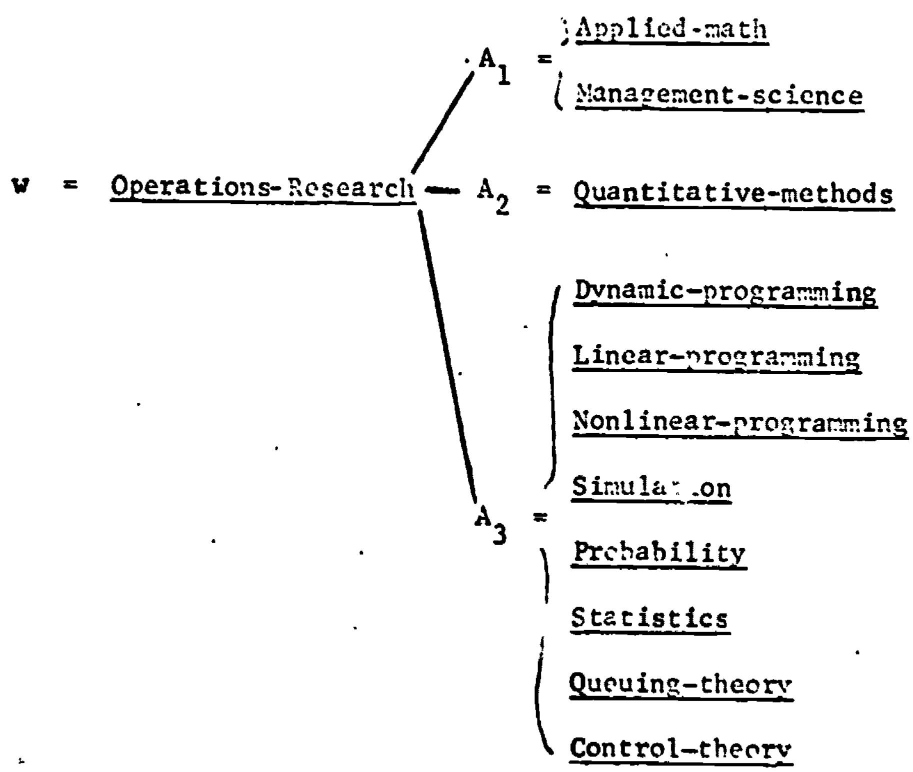
      Then give list A_2. This contains all words equivalent to or synonymous with w. It is those words on an equal footing with w, the same category as w.

      Finally encode A_3 which are those words (i.e., categories) that are subsets of or follow from or are derivable from w. It is the category(s) just below the one represented by w.

Tend to limit the number of entries of each A_i to approximately three or the total (of all members of all the A_i's) to nine. This rule is not fixed.

If necessary, re-arrange space (s) to fit large lists, by boxing in more lines where needed. If more space is needed for other words, use the back of the first page. There is a box to check on the first page if you do this. The same is true for the subgoal list on the second page.

For example, if operations-research had appeared in section (d), then section (e) might have:



Note the use of the hyphen. This is to emphasize that such word pairs as "operations research" and "complex variables" are really to be considered single terms. The composite word (e.g., complex-variables) constitutes a single term W. For very general words (e.g., management-science, philosophy), there may be no logical upper neighborhood. In such cases, the word itself is the only member of the A_1 set. For example, the A_1 set for "management-science" contains only the term "management science"; the A_1 set for "philosophy" contains only the word "philosophy".

2. Level

Choose whether this module is to be considered an introductory, intermediate, or advanced one. Your choice will depend on your purpose in encoding it. Other criteria can affect your choice. For example, the same module may be encoded introductory or intermediate depending on the length of time spent on the module. Abbreviations for levels are: introductory = 1, intermediate = 2, advanced = 3. If you wish, you may specify a range.

3. Time

The remarks in "level" above hold here too analogously. In giving the time, specify the range. The first box allows for a number and a measurement (i.e., hours, days, etc.) as does the second box which gives the upper range. Try to use the same measurements in giving the lower and upper ranges (e.g., "1 day - 3 days" rather than "8 hours - 3 days"). This is not a fixed rule.

4. Media

Check off the media in which this module is encoded. If "other" applies, place a message in quotes that tells how the material is encoded.

5. Motive

Give two numbers which describes the range of motivation, ranked from one to nine, within which this material is suitable for use. One is lowest and nine is highest. Thus, if the material is suited for only the most serious students, one might choose the range 8-9. A very simple "popularization" may be more suited to students with little seriousness of purpose (i.e., 1-3). Material may be appropriate for extremely broad (e.g., "1-9") or narrow ranges (e.g., "9-9", "1-1", "5-5").

6. Access

Check the box that indicates how one may get the module for use. If the box "other" is checked, give a message in quotes that explains how one can access the material.

7. Citation

a. Formal description

A formal (bibliographic-like) citation of the module is required.

b. Evaluation

A number reflecting a measure of the quality of this module (from one as the lowest, rated "quite poor" to ten the highest, rated "excellent"). Naturally, the goal that this module is to serve will directly affect the choice here.

8. Prerequisites

a. Education level needed

Choose the number corresponding to the highest level needed before one can "handle" this module. This is often an estimate of "sophistication" needed.

b. Prerequisites needed

Give a complete GOAL statement that describes the prerequisites. This goal described is the one that the student need's to fulfill to be able to accept this module. There may be several goals necessary; list each separately. If one goal obviously and necessarily precedes another, make an attempt to list it first. This is not a requirement.

Note that you are to estimate the suggested time, motive, and mastery ranges that suitably describe the subgoals as you intend them. The total time of all the subgoals should not exceed the total time alloted for this module in part 3 above. By specifying ranges rather than exact times, you will eliminate the need for constant checking of the total time or petty number-juggling. Also there may be some correlation between the motive of the module itself and the kinds of motives to be expected in the pursuit of the subgoals. Of course, there may be no relations, exceptions, etc. There is no fixed rule involved. You may use a range for each parameter, and hence, can include and exclude whatever you wish.

c. Aptitudes

If any of the standard tests listed are pertinent, give the minimum scores needed or expected in the respective boxes. If another test is required, give its name and minimum score in the extra space, as provided (in quotes).

d. Attitudes

If any particular attitudes are required for the module, either tests and minimum scores may be listed or quoted material entered.

e. Other

Give in quotes any prerequisite that must be met that could not be encoded above. This message will be returned to the user as an inquiry; a yes/no response from him will determine whether he meets the condition. A "no" will cause the module to be rejected as unsuitable.

9. Mastery

a. There are four mastery levels, analgous to a student demonstrating competence in some course with either a grade of "pass" (level 1), a grade of "C" (level 2), a grade of "B" (level 3), or a grade of "A" (level 4). Specify the level (or range of levels if this applies), for which this module is suited. Make an effort to achieve agreement between the goal of this module and this level indicated (or range specified).

b. Specify evaluation or testing procedures that would adequately reflect a competence at the mastery level (s) specified. This portion too will be quoted. Be clear and complete.

Module Number [ ]
(plus your
name ____________________

Module Encoding Form
Content Modules

This module is the same as
module ____________
in sections ____________

1. Module NAME

a. Identifying Name ____________________

b. Abstract: ____________________

c. In natural language: ____________________
(a sentence)

d. Given in terms of the AREA-part of the GOAL language:

e. Semantics:

w_1 = __________: A_1 = __________ A_2 = __________ A_3 = __________

w_2 = __________: A_1 = __________ A_2 = __________ A_3 = __________

w. = __________: A_1 = __________ A_2 = __________ A_3 = __________

Check here if other words
are on back [ ]

2. LEVEL: [ ] (a number or range where introductory = 1, intermediate = 2, advanced = 3)

3. TIME (for completion):

minimum [ ] maximum [ ]

4. MEDIA (in which module is encoded):

books	course	film	audio	video-tape	computer	journal	seminar	consultation

other: [ ]

5. MOTIVE:

material suitable for students with motives ranging from to (range [1,9]

6. ACCESS:

Hunt library	Science Library	bookstore	audio/visual room

other: [ ]

7. CITATION:

a. Formal description of material: ______

______

______

b. Evaluation (Score from 1-10, where 1 = quite poor, ..., 10 = excellent):

[ ]

8. PREREQUISITES:

a. Educational level needed for module

high school	college	graduate
0	1 | 2 | 3 | 4	5 | 6 | 7 | 8 | 9

b. PREREQUISITE LIST:

	Area/Level	Mastery (range 1-4)	Motive (range 1-9)	Time range
1.				
2.				
3.				
4.				
5.				
6.				
7.				

check here if other prerequisites are on the back ☐

c. Aptitudes:

GRE	SAT	IQ	ATGSB	OTHER:

d. Attitudes: ______________________________

e. Other (any material of important nature allowed, to be answered "yes" or "no" by the student):

9. MASTERY

a. Mastery Level [ ] (a number or range between 1 and 4)

b. Mastery Tests

Appendix C

STRUCTURE MODULE ENCODING INSTRUCTIONS

0. Introduction

The purpose of these instructions is to aid you in filling out the encoding form. You will find both new information as well as summaries of some information that was in the Encoding Guide, which we assume that you have already read. In particular, we assume that you are familiar with a GOAL statement, understand the purpose of the subgoal lists in structure modules as well as the distinction between structure and content modules, and are familiar with the intent of the mastery section (giving integrative information when necessary as well as testing procedures). Checking each number and subsection of the encoding form with these instructions should facilitate your encoding task. In the upper left hand portion of each form, you will find a place to number (in any fashion) each module as well as give your name; in the right hand corner, you may refer to an earlier module, which will imply that this module is identical to the one so referenced, except for those sections that you fill in (which will override the sections in the referenced module). This is a "ditto" box, for your convenience, when you find you are encoding the same thing, or same section, over and over. The rest of the guide is self-explanatory.

1. Module Name

   a. Choose any convenient simple name, in natural language, for referring to this module. (This may be the same as part (b) or (c)).

   b. In natural language, give the purpose or goal of the module; this may often be simply the goal statement of the module, part (c), or even part (a) again.

   c. Give the goal that this module is to satisfy in terms of the area-part of the GOAL language. Again, this could be the same as part (a) of (b).

   d. Semantics

      For each word, or hyphenated terms (w) used in (c) above, give three lists. The first list (A_1) contains all words which imply w, or from which w is an immediate subpart or subset. In a knowledge domain, it is the category just above the category that contains w. The decision of A_1 is based on the encoder's intended use of w, and hence, is based on the content and context of the module being encoded.

      Then give list A_2. This contains all words equivalent to or synonomous with w. It is those words on an equal footing with w, the same category as w.

      Finally encode A_3 which are those words (i.e., categories) that are subsets of or follow from or are derivable from w. It is the category(s) just below the one represented by w.

      Tend to limit the number of entries of each A_i to approximately three or the total (of all members of all the A_i's) to nine. This rule is not fixed.

If necessary, re-arrange space (s) to fit large lists, by boxing in more lines where needed. If more space is needed for other words, use the back of the first page. There is a box to check on the first page if you do this. The same is true for the subgoal list on the second page.

For example, if operations-research had appeared in section (c), then section (d) might have:

A_1 = Applied-math, Management-science

w = Operations-Research — A_2 = Quantitative-methods

A_3 = Dynamic-programming, Linear-programming, Nonlinear-programming, Simulation, Probability, Statistics, Queuing-theory, Control-theory

Note the use of the hyphen. This is to emphasize that such word pairs as "operations research" and "complex variables" are really to be considered single terms. The composite word (e.g., complex-variables) constitutes a single term W. For very general words (e.g., management-science, philosophy), there may be no logical upper neighborhood. In such cases, the word itself is the only member of the A_1 set. For example, the A_1 set for "management-science" contains only the term "management-science"; the A_1 set for "philosophy" contains only the word "philosophy".

2. Choose whether this module is to be considered an introductory, intermediate, or advanced one. Your choice will depend on your purpose in encoding it. Other criteria can affect your choice. For example, the same module may be encoded introductory or intermediate depending on the length of time spent on the module. Abbreviations for levels are: introductory = 1, intermediate = 2, advanced = 3. If you wish, you may specify a range.

3. Motive

Give two numbers which describes the range of motivation, ranked from one to nine, within which this goal is suitable for use. One is lowest and nine is highest. Thus, if the goal is suited for only the <u>most serious</u> student, one might choose the range 8-9. A very simple approach may be more suited to students with little seriousness of purpose. (i.e., 1-3). The goal may be appropriate for extremely broad (e.g.,"1-9") or narrow ranges (e.g., "9-9", "1-1", "5-5").

4. Subgoal List

Give the various subgoals whose accomplishment will satisfy the goal of this module. Note that you are to estimate the suggested time, motive, and mastery ranges that suitably describe the subgoals as you intend them. The total time of all the subgoals should not exceed the total time alloted for this module in part 3 above. By specifying ranges rather than exact times you will eliminate the need for constant checking of the total time or petty number-juggling. Also there may be some correlation between the motive of the module itself and the kinds of motives to be expected in the pursuit of the subgoals. Of course, there may be no relations, exceptions, etc. There is no fixed rule involved.

You may use a range for each parameter, and hence, can include and exclude whatever you wish. If some subgoals would best be started after the completion of other subgoals, put these dependent subgoals after the more "preliminary" ones. When estimating time for completion of some module, assume the student is working full-time exclusively on that module. It is his only task to be pursued throughout the whole work-day.

The estimate of the time to complete a module includes the time needed to do any integrative assignments given in the mastery section. To convert "semester courses" into full-time efforts (plus other equivalences) use the following guide:

a. One 3-hour course = 4 weeks or 1 month of full-time effort

b. One 4-hour course = 5 1/2 weeks or 1 1/3 months

c. Six weeks of a course = 1 1/2 weeks

d. A mini-course = 2 1/2 weeks

5. Prerequisites

a. Education level needed

Choose the number corresponding to the highest level needed before one can "handle" this module. This is often an estimate of "sophistication" needed.

b. Aptitudes

If any of the standard tests listed are pertinent, give the minimum scores needed or expected in the respective boxes. If another test is required, give its name and minimum score in the extra space, as provided (in quotes). You may use values or percentiles.

c. Attitudes

If any particular attitudes are required for the module, a reference to the test and minimum scores may be listed or a quoted message entered.

d. Other

Give (in quotes) any prerequisite that must be met that could not be encoded above. This message will be returned to the user as an inquiry; a yes/no response from him will determine whether he meets the condition. A "no" will cause the module to be rejected as unsuitable.

6. Mastery

a. There are four mastery levels, analogous to a student demonstrating competence in some course with either a grade of "pass" (level 1), a grade of "C" (level 2), a grade of "B" (level 3), or a grade of "A" (level 4). Specify the level (or range of levels if this applies), for which this module is suited. Make an effort to achieve agreement between the goal of this module and the mastery level you specify.

b. Give suitable educational resources or activities that may be necessary to integrate (or summarize or consolidate) the collection of subgoals specified in part 4. Thus this material is at the level of the goal itself, not any sub goal. If you do not feel this is needed or is already achieved in the accomplishment of the subgoals, you may omit this section. If an entry is made in this section, it will be outputted to the student exactly as it appears (it is merely quoted). Be clear, complete, and give concise references where necessary. This is the only information the student will have.

c. Based on part (b) above as well as the subgoals themselves, specify evaluation or testing procedures that would adequately reflect a competence at the mastery level (s) specified. This portion too will be quoted. Be clear and complete. The tests evaluate mastery of the goal rather than just the subgoals, though there are times when this may be equivalent.

Module Encoding Form:
Structure Modules

Module Number [ ] (plus your name ______)

This module is the same as module ______ in sections ______

1. Module NAME

a. Identifying Name: ______

b. In natural language, give the goal of the module: ______ (a sentence)

c. Give goal in terms of the AREA-part of the GOAL language: ______

d. Semantics:

v_1 = ______ : A_1 = ______ A_2 = ______ A_3 = ______

v_2 = ______ : A_1 = ______ A_2 = ______ A_3 = ______

v_3 = ______ : A_1 = ______ = ______ A_3 = ______

Check here if other words are on back [ ]

2. LEVEL: [ ] (a number or range where introductory = 1, intermediate = 2, advanced = 3)

3. MOTIVE: Completion of module appropriate for students with motives ranging from from [ ] to [ ] (range [1,9]

4. SUBGOAL LIST:

Area/Level	Mastery (range)	Motive (range)	Time (range)
1. ______	______	______	______
2. ______	______	______	______
3. ______	______	______	______
4. ______	______	______	______
5. ______	______	______	______
6. ______	______	______	______
7. ______	______	______	______

Check here if other sub-goals are on back [ ]

5. GENERAL PREREQUISITES:

a. General educational level needed:

high school	college	graduate
0	1 2 3 4	5 6 7 8 9

b. Aptitudes:

GRE	SAT	IQ	ATGSB	Other:

c. Attitudes: ______________________

____

____

d. Other (any material of important nature allowed, to be answered "yes" or "no" by the student):

____

____

6. Mastery

a. Mastery Level [ ] (a number or range between 1 and 4)

b. Mastery Assignment

____

____

____

____

____

____

c. Mastery Tests

____

____

____

____

____

____