CURRICULUM VIEWED AS A BINARY SYSTEM: AN APPROACH TO THE DETERMINATION OF SEQUENCE - A PROJECT REPORT

Thomas R. Renckly
Education Specialist
Curriculum Design Coordinator
U.S. Navy Recruiting Command
Orlando, Florida

Gary Orwig, Ph.D.
Asst. Professor
College of Education
University of Central Florida
Orlando, Florida

ABSTRACT

Determining alternative curriculum sequences is a tedious task involving many individuals and analysis of large amounts of curriculum-related information. Because these tasks are not readily reduceable to mathematical operations, and because educators and curriculum designers are generally not so inclined, computer intervention into this design process has been meager. Nevertheless, the power of the computer to handle vast amounts of information coupled with its high speed manipulation ability makes it an ideal instrument to use in the instructional design process. The project reported herein describes the development and application of a model by which a curriculum may be analyzed to determine alternative instructional sequences based upon curriculum objectives and limiting constraints. The project's primary goal is to ultimately apply the model to the analysis and design of instructional sequences for 16 closely related courses currently under development by the U.S. Navy Recruiting Command.

THE PROJECT

The U.S. Navy Recruiting Command has undertaken the task of developing training programs for 16 individual, yet closely related jobs (or billets) within the duty called recruiting. Although development of training programs is not new to the military, the approach to this particular curriculum design problem is. Due to the close interrelationship between and among each of the 16 courses under development, there is heavy reliance upon instructional sequence. For example, several competencies have been identified which overlap in many of the 16 courses. Because such overlaps parallel real-life recruiting practices, they were not avoided. It is educationally sound practice to structure student learning experiences so as to simulate reality as much as possible. Some educational psychologists believe this produces maximum learning transfer.

However, the payback comes in the form of an increased demand on curriculum sequencing. A non-sequitor or ill-sequenced curriculum can damage the realism of learning experience. It can also reduce the student's ability to internalize the concepts presented. The student may appear to perform satisfactorily in the school environment but become disoriented in trying to perform a similar task in the real-world environment. Thus the need for well-sequenced instruction.

The *classical* approach to determining acceptable instructional sequences has characteristically been human intuition. Such an approach is time-consuming in that it seldom produces an adequate sequence on the first attempt. Additionally, considerable work is involved with each iteration. In this current project, intuition is simply not sufficient for the task of aligning 16 courses into a unified sequence.

In any curriculum design problem, there are a myriad of variables which may dramatically affect the ultimate instructional sequence. However, a model exists in the literature which is capable of dealing with complex systems of interacting variables.(1) This model, known as Interpretive Structural Modeling (ISM), has been successfully applied to the sequencing of process elements in a number of design projects in the fields of engineering, agriculture as well as a host of other complex scientific and social problems. (2) Unfortunately, ISM has seen limited use in the field of education as a tool for planning and design. In fact, this writer has found only one such use. And this has been accomplished primarily by Sato and his colleages in Japan. (3,4) It is the intent of this project to adapt ISM to the instructional sequencing problem and build upon the work that has already been done in this area with the hope that successful development here may spawn more educational uses of ISM in this country.

As the initial project report, this paper will present some basic theory underlying the ISM concept as well as a method which shows great promise in assisting the curriculum designer in determining appropriate alternative instructional sequences.

COMPLEXITY IN THE DESIGN PROCESS

The instructional systems approach, or any systematic approach to instructional design for that matter, is anchored in mathematical modeling. It has long been recognized that a systems approach to instructional development is patterned after the scientific method(5) which is in itself a modeling approach. (6) The question then arises as to why the design of instruction is not treated by a mathematical approach to approximating the shape and scope of a curriculum! In their text, Programmed Learning in Perspective, the authors allude to the mathematical character of curriculum. They describe a quasi-mathematical technique (termed the matrix technique) which is useful in determining optimum unit sequencing within programmed instructional material. (7) Davies further generalized this procedure, demonstrating its utility in optimizing presentation sequences for objectives of an entire course of instruction. (8) The logical extension of this work leads one to believe there may be a method by which a complex curriculum composed of disjointed competencies might be alternatively sequenced.

Successful instructional design models call for some sort of determination of sequence at some time during the design process. Often this is achieved through construction of objective trees (or hierarchies). In fact, instruction in the building of such hierarchies is often in great detail (9) -- testimony to its importance in the instructional design process. To anyone familiar with such a task, it is immediately obvious that instructional hierarchies are complex structures *not only to build, but also to interpret.* The casual observer is often unable to visualize the many possible sequencing strategies from the maze of lines displayed. Such insight requires a knowledge of the course content and at least some grounding in basic learning psychology. Yet, even if this prior knowledge is assumed, the task of choosing an appropriate sequence from all the possible sequences displayed on the hierarchy is still not easy. Mathematical modeling and operations research provide some interesting algorithms, however, which demonstrate the potential to assist in solving complex instructional sequencing problems.

In their paper Unified Program Planning, Hill and Warfield describe a method for reducing complex systems of elements (in our case, objectives) into a matrix which describes their mutual relationships. (10) They call this a *self-interaction matrix* because it contains information relating to the interaction of each element with itself and the others in the system. The authors define such a matrix as containing enough information to construct an objectives tree.

For this project, their matrix method is used in developing an objectives hierarchy from an initial set of course objectives. The worth of this matrix method is in its ability to produce a hierarchy which actually contains more information than hierarchies developed by other means. As an example of the kinds of information stored, and generally gleaned from typical objectives trees, consider the hierarchy of a hypothetical curriculum containing 15 interrelated objectives as shown in Figure 1 below.

Several *bits* of information are implicitly stored in this hierarchy. For example, OBJECTIVE 1 appears to be the terminal objective for the curriculum. That is, all other objectives either directly or indirectly terminate at OBJECTIVE 1. Also, OBJECTIVEs 3,8,10,11,12,13,14, and 15 are at base levels with no supporting objectives. Thus, these are ideal starting points for sections or modules of instruction. Yet another *bit* of information available from the hierarchy is implied by the arrows connecting the various objectives. Their pattern indicates the existence of partitions between objective clusters (though such partitions are purely arbitrary). For example, one such partition could be OBJECTIVEs 11, 6, 2, and 1; another, OBJECTIVEs 13, 12, 7, 2, and 1; another, OBJECTIVEs 8, 4, and 1; still another, OBJECTIVEs 15, 14, 9, 5, and 1; etc. Although such partitions are arbitrary, these groupings give some indication of the amount of information potentially stored in an objective hierarchy. All these *bits* of information taken together represent a detailed picture of how each objective interacts with all the rest in this particular hypothetical curriculum.

Yet, a completely different class of interactions exists which also come to bear on a curriculum. This class contains such instruction-related items as resource constraints (money, manpower, and time), student needs, types of learning activities available to students to meet course objectives, types and timing of measurement tests, etc. Each of these has a effect on whether or not a given instructional sequence will work effectively. However, these interactions cannot be stored or displayed on a typical objectives hierarchy, such as that in Figure 1. Even by looking at the hierarchy, it is impossible to discern if such interactions were taken into consideration in the hierarchy's development. Of course, this information could be superimposed onto the hierarchy, however, this could very easily complicate the diagram to the point that interpretation becomes impossible. The reason for this is that there seems to be an upper limit on the amount of

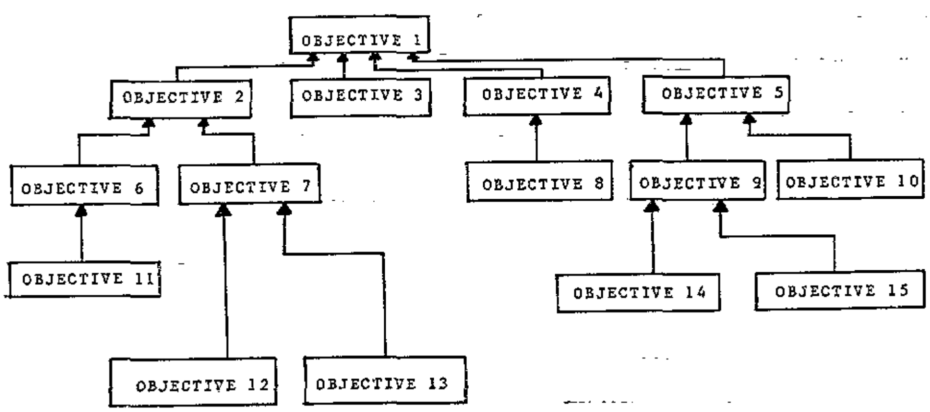


Figure 1. An objectives hierarchy containing 15 interrelated objectives

information that one human can process and operate on at any given time. "Research tentatively shows that the amount of information man is capable of processing is limited, and more data...do not necessarily increase the quality of decisions in the same proportion." (12) It must be made clear at this juncture that the self-interaction matrix is not intended to replace the objectives tree, but only to enhance it. "The self-interaction matrix ...is not as clear as the objectives tree for viewing the relationships among objectives, but it incorporates significant advantages in relating objectives to constraints, alterables and needs" inherent in the instructional system. (13) Thus, in this project, both the matrix and the objectives tree are utilized to their maximum advantages.

In actuality, the curriculum designer, the teaching staff, and the management personnel each recognize a different set of such interactions as mentioned above which impact on the curriculum. Thus, the designer must spend considerable time with teachers to develop an instructional hierarchy which takes into account as many of the ancillary interactions as possible. And when they finally come to an agreement on a reasonable teaching sequence, they may find (to their dismay) that the administration rejects the plan because of some constraining factor neither the designer nor the teachers knew about. Such situations are common and illustrate the need for a model which can contain and process much more curriculum-relevant information than is currently possible. The major requisites of such a model would have to be: convenience, simplicity and utility.

Convenience can be described as the ease of applying the model to the design problem. Simplicity refers to the quantity of information that must be provided by the user for the model's operation. And utility can be expressed as the model's adaptability to a general class of curriculum design problems - from the relatively simple task of sequencing information within a programmed text to the highly complex task of determining the sequence for effective learning in a *spiraled* "K through 12" educational network. ISM, the model used in this project, posesses these primary requisites in varying degrees and is thus a likely candidate for the curriculum design problem.

CHARACTERISTICS OF A BINARY MATRIX

Before detailing the results and current status of the project, we should first clarify the terms used. The literature on the subject is primarily mathematical. For this discussion, the mathematics have been simplified in some places, and eliminated altogether in others. In its place, intuitive arguments have been used. Readers interested in the actual mathematical derivations are referred to the work of Warfield. (14)

A binary matrix is a square array of elements whose values are either 1 or 0. If all the main diagonal elements (from upper left to lower right in the array) are 1s, the matrix is said to be reflexive. Thus, an irreflexive matrix has some 0s on its main diagonal. An irreflexive matrix must be made reflexive in order to be analyzed by the matrix method. Fortunately, this is easily accomplished by adding to the irreflexive matrix an identity matrix. This is also a binary matrix with 1s along the main diagonal and 0s everywhere else.

The rows of a matrix are usually referred to by the letter i, while the columns are usually referred to by the letter j. Every matrix element occupies a position which is at the intersection of a row and column. Thus, any arbitrary element of a matrix can be referred to as the (i,j) element. If a matrix element (i,j) and its "mirrir-image" element (j,i) are the same value (either 1 or 0), then the matrix is said to be symmetric. The degree of symmetry depends upon how many elements (i,j) are matched to their "mirror-images". To illustrate this more clearly, note the mirror-image quality in the binary matrix in Figure 2 on both sides of the main diagonal. For clarity, the zeros have been removed.

	1	2	3	4	5
1	1			1	1
2		1	1		1
3		1	1		
4	1			1	1
5	1	1		1	1

Figure 2. Mirror-Image Symmetry Above and Below the Main Diagonal (dashed)

A binary matrix may have a few assymetric points and still be considered symmetric for purposes of this method if the number of assymetric points are kept to a minimum. In reality, an assymetric matrix yields the best instructional hierarchy. Thus, the degree of assymetry in the matrix determines the richness of the resulting hierarchy. However, this depends upon the nature of the objectives under consideration and the nature of the interactions among objectives - both of which are dependent on the type of curriculum being designed.

TRANSITIVE RELATIONS AND DIRECTED GRAPHS

In determining an appropriate curriculum sequence, considerable thought must be given to how each instructional objective relates to all other objectives in the curriculum. During the so-called "front-end analysis" phase of a design project, relationships between what the student needs and what the curriculum will offer to meet those needs are more likely to be philosophical intuitions than rigorous proofs. The mathematical character of ISM, however, requires a more detailed analysis of such relationships. These relationships are logical rather than mathematical.

Consider the logical relationship among three objectives (a, b, and c) as illustrated in Figure 3. Figure 3A shows that objective a relates to objective b, and that b relates to c. However, objectives a and c are not directly related to one another. Clearly, if objective b were removed from the curriculum, objectives a and c would exist as isolated entities. Such a relation among objectives is called intransitive because there is no direct relation or, or connection, between objectives a and c.

Figure 3B, on the other hand, indicates that

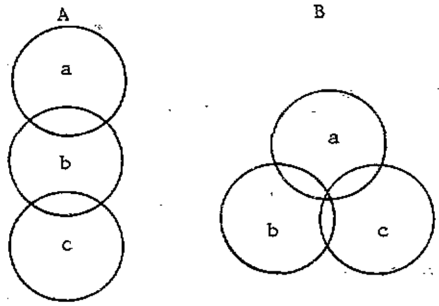


Figure 3. Two Types of Relationships Among Objectives a, b and c

all objectives are directly related to each other. If any one of them is removed, the remaining two are still linked together through a binding relationship. A transitive relation is one in which each objective relates, or is somehow linked to the others in the group.

Though we have used the term "relation" numerous times, we have not yet clearly defined it. A relation is a phrase or term that shows how two or more elements (or objectives) interconnect, or link, to one another. Whether or not a relation is transitive depends not so much on what relation is used, as on the situation in which it is used.

For example, consider the relation "is contained within". If a "is contained within" b, and if b "is contained within" c, then it follows that a "is contained within" c. We can visualize this relation in Figure 4. Any objectives a, b

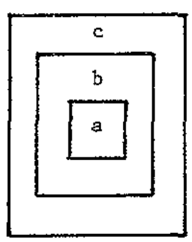


Figure 4. Visualization of the relation *is contained within*

and c for which this relation holds true is considered a transitive set of objectives. It must be borne in mind, however, that even though the *relation* is transitive, not all *objectives* will suit it. If one particular relation is not transitive across an entire set of objectives under consideration, a relation that does apply must be found. Each new relation chosen, of course, must be similarly tested to insure transitivity within the entire objective set.

Some relations are intransitive in all but the most specific of situations. For example, Warfield has reported that the relation "obeys" fails the transitivity test (15): if a "obeys" b, and if b "obeys" c, a may not necessarily "obey" c. In fact, most relations are situation specific. They must be carefully considered in the context of the entire objective set.

Once a transitive relation has been identified, it remains to be discovered how the relation specifically affects each pair of objectives. Does, for example, the relation link objective a to objective b, or vice versa? A simple example should serve to illustrate this point. Consider the transitive relation *depends upon prior accomplishment of*. If objective a *depends upon prior accomplishment of* objective b, then clearly, b cannot possibly *depend upon prior accomplishment of* objective a. In addition to illustrating assymetry, this example also illustrates the concept of directability. In the above example, an arrow could be drawn between objectives a and b with the arrowhead pointing toward objective a to show that a *depends upon prior accomplishment of* b.

If all such directed relations between objectives are considered, a picture of the interactions can be obtained. Such a picture is known as a directed graph. Warfield has shown that any directed graph or *digraph* possesses an associated binary matrix (16). A given binary matrix, however, may produce a number of alternative digraphs. Any one of them could be used as an objectives hierarchy to describe the interrelationships among instructional objectives. The binary matrix needed to produce the digraph is called the reachability matrix. If transitive and assymetric, this matrix can be manipulated to produce a digraph (otherwise known as an objectives hierarchy). The procedure, described by Warfield, requires the formation of tables consisting of various arrangements of objectives (17). The actual procedure followed for this project will be described in greater detail in the next section. This cursory overview of the underlying theory supporting the matrix method will suffice for our purposes here.

THE PROJECT'S METHOD

The process of generating a digraph from a set of curriculum objectives is a straightforward approach composed of the following steps:

1. Identify the objectives of the curriculum.
2. Determine a transitive relation which applies to the objectives in the context of the instructional situation.
3. Place objective relations into a matrix format - termed a self-interaction matrix.
4. Manipulate the matrix into a suitable form - termed a reachability matrix.
5. Re-order the rows and columns of the reachability matrix and partition it to reflect hierarchical levels - termed a modified reachability matrix.
6. Compute a hierarchy (or digraph) from the modified reachability matrix

The curriculum design project described in this paper follows this six step process for generating hierarchies and determining instructional sequences. Since the approach is both complex and time consuming, computer algorithms have been designed to perform most of this work. The remainder of this paper details the process followed in the Navy curriculum project.

Step 1

After the front-end analysis had been completed for the 16 courses under development, a listing of tasks required for training were identified. And from these, a series of learning objectives were developed for each course. One course was used for the pilot study in this project.

Step 2

The transitive relation *is necessary to accomplish* was agreed upon by the subject matter specialists, the curriculum design staff and the approving board for curriculum development. This relation was used in the analysis of the relationships between every possible pair of objectives. Since 18 objectives were originally identified for training in the pilot course, 18 x 18 (=324) distinct objective pairs were analyzed via the agreed upon relation.

Step 3

For each of the 324 objective pairs, a 1 was placed into the corresponding cell of a matrix, if the relation was true. If, however, the relation was false for a particular pair, a 0 was placed in the appropriate matrix cell. The resulting matrix required approximately four manhours to accomplish. The self-interaction matrix which resulted is shown in Figure 5.

Objective Number

Objective Number	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
1	1	0	0	0	0	1	1	0	0	0	1	0	0	1	1	0	1	0
2	0	1	0	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0
3	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
4	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0
5	0	0	1	1	1	0	0	0	0	1	0	0	1	0	0	0	0	0
6	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
7	0	0	0	0	1	0	1	1	0	0	1	0	0	1	1	0	1	1
8	0	0	0	0	1	0	1	1	0	1	1	0	0	1	0	0	0	0
9	0	0	1	0	1	0	1	1	1	0	0	0	0	1	0	0	0	0
10	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	1	0
11	0	0	0	0	1	0	1	0	0	0	1	1	0	1	0	0	1	0
12	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0	0	0
13	0	0	1	1	1	0	0	0	1	1	0	0	1	0	0	0	0	0
14	0	0	0	0	1	0	1	0	0	0	0	0	1	1	0	0	1	1
15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
16	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	1	0	0
17	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	0
18	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1

Figure 5. A Self-Interaction Matrix for the pilot Course

Warfield describes an algorithm with which a computer can be programmed to accomplish this data entry step with reduced effort on the part of the user.(18) Currently, the algorithm is being modified for use in this project, but was not used for the pilot project. After creation of the self-interaction matrix of Figure 5, it was loaded into a BASIC language microprocessor via a prompting routine developed by Orwig. The flowchart of this routine is shown in Figure 6.

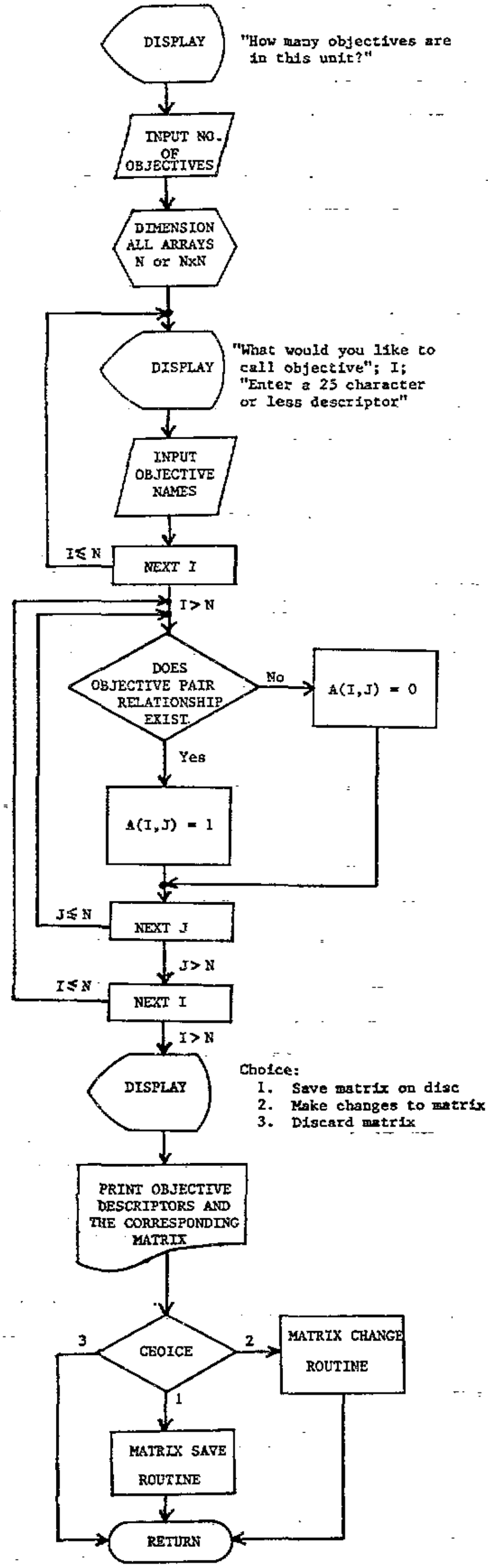


Figure 6. A Prompting Routine for Matrix Data Entry

Step 4

The self-interaction matrix of Figure 5 must be manipulated into a reachability matrix before further analysis can be performed. This manipulation involves raising the self-interaction matrix (M) to successive powers (squared, cubed, etc. by Boolean multiplication) until the following equality is met: $M^n = M^{n+1}$. An algorithm used to accomplish this multiplication process is presented in flowchart form in Figure 7. (A flowchart of the entire computer program developed by the authors appears in the Appendix.) *According to theory, if there are N objectives in* the matrix, the reachability matrix will be derived in *N-1* or less iterations. (19) The self-interaction matrix for the pilot course (Figure 8) was converted to reachability form in four iterations. In other words, the matrix of Figure 8 multiplies out in four iterations to form the reachability matrix in Figure 9, which satisfies the equality: $M^3 = M^4$.

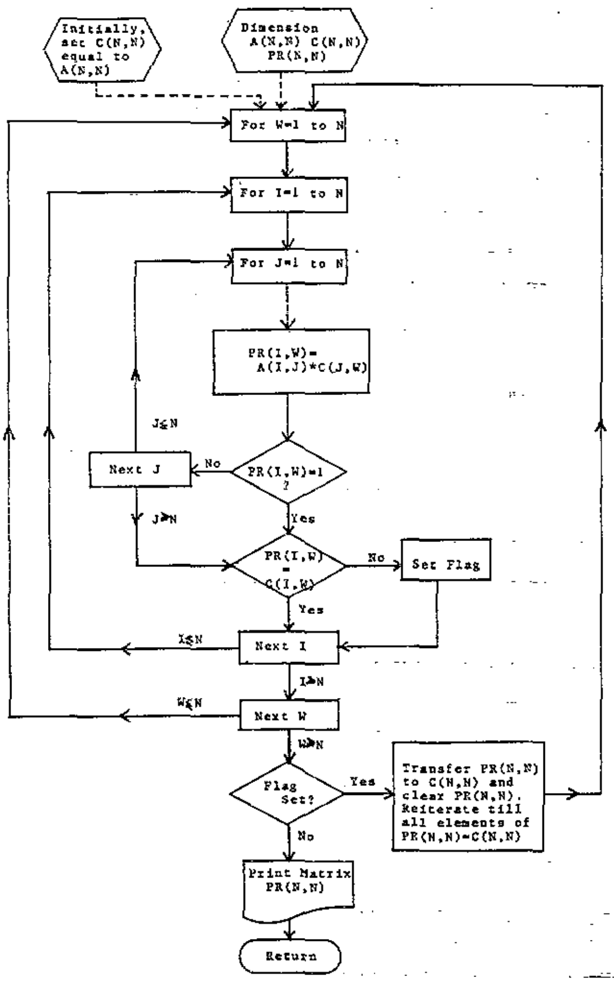


Figure 7. An Algorithm To Convert A Self-Interaction Matrix A(N,N) Into A Reachability Matrix PR(N,N)

Objective Number

Objective Number	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
1	1	0	0	0	0	1	1	0	0	0	1	0	0	1	1	0	1	0
2	0	1	0	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0
3	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
4	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0
5	0	0	1	1	1	0	0	0	0	1	0	0	1	0	0	0	0	0
6	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
7	0	0	0	0	1	0	1	1	0	0	1	0	0	1	1	0	1	1
8	0	0	0	0	1	0	1	1	0	1	1	0	0	1	0	0	0	0
9	0	0	1	0	1	0	1	1	1	0	0	0	0	1	0	0	0	0
10	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	1	0
11	0	0	0	0	1	0	1	0	0	0	1	1	0	1	0	0	1	0
12	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0	0	0
13	0	0	1	1	1	0	0	0	1	1	0	0	1	0	0	0	0	0
14	0	0	0	0	1	0	1	0	0	0	0	0	1	1	0	0	1	1
15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
16	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	1	0	0
17	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	0
18	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1

Figure 8. The Self-Interaction Matrix of Figure 5

Objective Number

Objective Number	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1
2	0	1	0	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0
3	0	0	1	1	1	0	1	1	1	1	1	1	1	1	1	0	1	1
4	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0
5	0	0	1	1	1	0	1	1	1	1	1	1	1	1	1	0	1	1
6	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
7	0	0	1	1	1	0	1	1	1	1	1	1	1	1	1	0	1	1
8	0	0	1	1	1	0	1	1	1	1	1	1	1	1	1	0	1	1
9	0	0	1	1	1	0	1	1	1	1	1	1	1	1	1	0	1	1
10	0	0	1	1	1	0	1	1	1	1	1	1	1	1	1	0	1	1
11	0	0	1	1	1	0	1	1	1	1	1	1	1	1	1	0	1	1
12	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0	0	0
13	0	0	1	1	1	0	1	1	1	1	1	1	1	1	1	0	1	1
14	0	0	1	1	1	0	1	1	1	1	1	1	1	1	1	0	1	1
15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
16	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	1	0	0
17	0	0	1	1	1	0	1	1	1	1	1	1	1	1	1	0	1	1
18	0	0	1	1	1	0	1	1	1	1	1	1	1	1	1	0	1	1

Figure 9. The Reachability Matrix Derived From the Matrix of Figure 8

Step 5

The purpose of this step is to partition the reachability matrix into submatrices which reflect the levels within the instructional hierarchy. The resulting partitioned matrix will be the reachability matrix modified by row and column interchanges. To determine the eventual order of this interchange, a table is created which contains a reachability set, an antecedent set and the product (or intersection) of both sets.

The reachability set for element 1 is found by inspecting row 1 of the reachability matrix (Figure 9). Every 1 in row 1 corresponds to a column index, and every such column index will be in the reachability set of element 1. To find the antecedent set of element 1, inspect column 1. To every entry of 1 in column 1, there is a corresponding row index; and the set of such row indices is the antecedent set of column 1. Each row and column is similarly considered in turn thus producing a table of reachability and antecedent sets for each row of the matrix.

In Figure 9, the row and column indices (1-18) are used to identify the respective elements of the reachability and antecedent sets. Table 1 is constructed from Figure 9 by inspection.

Table 1. A Reachability Table

ROW INDEX(S)	REACHABILITY SET R(S)	ANTECEDENT SET A(S)	SET PRODUCT R(S) ∩ A(S)
1	1 3 4 5 6 7 8 9 10 11 12 13 14 15 17 18	1	1
2	2 4 15	2	2
3	3 4 5 7 8 9 10 11 12 13 14 15 17 18	1 3 5 7 8 9 10 11 13 14 17 18	3 5 7 8 9 10 11 13 14 17 18
4	4 15	1 2 3 4 5 7 8 9 10 11 13 14	4
5	3 4 5 7 8 9 10 11 12 13 14 15 17 18	1 3 5 7 8 9 10 11 13 14 17 18	3 5 7 8 9 10 11 13 14 17 18
6	6	1 6	6
7	3 4 5 7 8 9 10 11 12 13 14 15 17 18	1 3 5 7 8 9 10 11 13 14 17 18	3 5 7 8 9 10 11 13 14 17 18
8	3 4 5 7 8 9 10 11 12 13 14 15 17 18	1 3 5 7 8 9 10 11 13 14 17 18	3 5 7 8 9 10 11 13 14 17 18
9	3 4 5 7 8 9 10 11 12 13 14 15 17 18	1 3 5 7 8 9 10 11 13 14 17 18	3 5 7 8 9 10 11 13 14 17 18
10	3 4 5 7 8 9 10 11 12 13 14 15 17 18	1 3 5 7 8 9 10 11 13 14 17 18	3 5 7 8 9 10 11 13 14 17 18
11	3 4 5 7 8 9 10 11 12 13 14 15 17 18	1 3 5 7 8 9 10 11 13 14 17 18	3 5 7 8 9 10 11 13 14 17 18
12	12 15	1 3 5 7 8 9 10 11 12 13 14 16 17 18	12
13	3 4 5 7 8 9 10 11 12 13 14 15 17 18	1 3 5 7 8 9 10 11 13 14 17 18	3 5 7 8 9 10 11 13 14 17 18
14	3 4 5 7 8 9 10 11 12 13 14 15 17 18	1 3 5 7 8 9 10 11 13 14 17 18	3 5 7 8 9 10 11 13 14 17 18
15	15	1 2 3 4 5 7 8 9 10 11 12 13 14 15 16 17 18	15
16	12 15 16	16	16
17	3 4 5 7 8 9 10 11 12 13 14 15 17 18	1 3 5 7 8 9 10 11 13 14 17 18	3 5 7 8 9 10 11 13 14 17 18
18	3 4 5 7 8 9 10 11 12 13 14 15 17 18	1 3 5 7 8 9 10 11 13 14 17 18	3 5 7 8 9 10 11 13 14 17 18

From Table 1, it is immediately apparent that the only rows for which the set product equals the reachability set are rows 6 and 15. These two rows are therefore removed from the table along with all references to numbers 6 and 15 everywhere else in the table. Thus, rows 6 and 15 from the reachability matrix (Figure 9) become the first two rows of the modified reachability matrix. These two rows will be considered the top level in the instructional hierarchy (or digraph).

Ordinarily, the references to rows 6 and 15 can simply be erased from the table, and the next iteration begun. For the purpose of illustration in this paper, however, each new (reduced) table will be enumerated.

Removal of all 6s and 15s results in the reduced form of Table 2. This time, the

Table 2. Reduced Table - Level 1 Removed

ROW INDEX(S)	REACHABILITY SET R(S)	ANTECEDENT SET A(S)	SET PRODUCT R(S) ∩ A(S)
1	1 3 4 5 7 8 9 10 11 12 13 14 17 18	1	1
2	2 4	2	2
3	3 4 5 7 8 9 10 11 12 13 14 17 18	1 3 5 7 8 9 10 11 13 14 17 18	3 5 7 8 9 10 11 13 14 17 18
4	4	1 2 3 4 5 7 8 9 10 11 13 14	4
5	3 4 5 7 8 9 10 11 12 13 14 15 17 18	1 3 5 7 8 9 10 11 13 14 17 18	3 5 7 8 9 10 11 13 14 17 18
7	3 4 5 7 8 9 10 11 12 13 14 17 18	1 3 5 7 8 9 10 11 13 14 17 18	3 5 7 8 9 10 11 13 14 17 18
8	3 4 5 7 8 9 10 11 12 13 14 17 18	1 3 5 7 8 9 10 11 13 14 17 18	3 5 7 8 9 10 11 13 14 17 18
9	3 4 5 7 8 9 10 11 12 13 14 17 18	1 3 5 7 8 9 10 11 13 14 17 18	3 5 7 8 9 10 11 13 14 17 18
10	3 4 5 7 8 9 10 11 12 13 14 17 18	1 3 5 7 8 9 10 11 13 14 17 18	3 5 7 8 9 10 11 13 14 17 18
11	3 4 5 7 8 9 10 11 12 13 14 17 18	1 3 5 7 8 9 10 11 13 14 17 18	3 5 7 8 9 10 11 13 14 17 18
12	12	1 3 5 7 8 9 10 11 12 13 14 16 17 18	12
13	3 4 5 7 8 9 10 11 12 13 14 17 18	1 3 5 7 8 9 10 11 13 14 17 18	3 5 7 8 9 10 11 13 14 17 18
14	3 4 5 7 8 9 10 11 12 13 14 17 18	1 3 5 7 8 9 10 11 13 14 17 18	3 5 7 8 9 10 11 13 14 17 18
16	12 16	16	16
17	3 4 5 7 8 9 10 11 12 13 14 17 18	1 3 5 7 8 9 10 11 13 14 17 18	3 5 7 8 9 10 11 13 14 17 18
18	3 4 5 7 8 9 10 11 12 13 14 17 18	1 3 5 7 8 9 10 11 13 14 17 18	3 5 7 8 9 10 11 13 14 17 18

reachability set R(s) and set product columns match for rows 4 and 12. As before, these rows are removed from the table and the reachability matrix to become the second level in the modified matrix. Again, removing all references to 4 and 12 from the above table results in the formation of Table 3.

Table 3. Reduced Table - Level 2 Removed

ROW INDEX(S)	REACHABILITY SET R(S)	ANTECEDENT SET A(S)	SET PRODUCT R(S) ∩ A(S)
1	1 3 5 7 8 9 10 11 13 14 17 18	1	1
2	2	2	2
3	3 5 7 8 9 10 11 13 14 17 18	1 3 5 7 8 9 10 11 13 14 17 18	3 5 7 8 9 10 11 13 14 17 18
5	3 5 7 8 9 10 11 13 14 17 18	1 3 5 7 8 9 10 11 13 14 17 18	3 5 7 8 9 10 11 13 14 17 18
7	3 5 7 8 9 10 11 13 14 17 18	1 3 5 7 8 9 10 11 13 14 17 18	3 5 7 8 9 10 11 13 14 17 18
8	3 5 7 8 9 10 11 13 14 17 18	1 3 5 7 8 9 10 11 13 14 17 18	3 5 7 8 9 10 11 13 14 17 18
9	3 4 5 7 8 9 10 11 12 13 14 15 17 18	1 3 5 7 8 9 10 11 13 14 17 18	3 5 7 8 9 10 11 13 14 17 18
10	3 5 7 8 9 10 11 13 14 17 18	1 3 5 7 8 9 10 11 13 14 17 18	3 5 7 8 9 10 11 13 14 17 18
11	3 5 7 8 9 10 11 13 14 17 18	1 3 5 7 8 9 10 11 13 14 17 18	3 5 7 8 9 10 11 13 14 17 18
13	3 5 7 8 9 10 11 13 14 17 18	1 3 5 7 8 9 10 11 13 14 17 18	3 5 7 8 9 10 11 13 14 17 18
14	3 5 7 8 9 10 11 13 14 17 18	1 3 5 7 8 9 10 11 13 14 17 18	3 5 7 8 9 10 11 13 14 17 18
16	16	16	16
17	3 5 7 8 9 10 11 13 14 17 18	1 3 5 7 8 9 10 11 13 14 17 18	3 5 7 8 9 10 11 13 14 17 18
18	3 5 7 8 9 10 11 13 14 17 18	1 3 5 7 8 9 10 11 13 14 17 18	3 5 7 8 9 10 11 13 14 17 18

From Table 3, the third level of the modified matrix is shown to be composed of rows 2, 16, 3, 5, 7, 8, 9, 10, 11, 13, 14, 17, and 18. Deleting all these references from Table 3 results in the formation of Table 4. Note that only row 1 remains to make up the fourth and final level of the modified matrix. The resulting modified matrix is shown in Figure 10. The heavy black squares clarify various submatrices which denote the four levels identified from the tables. Note that both row and column designations in the modified matrix have been identically interchanged. This is automatically accomplished by the computer algorithm.

Table 4. Reduced Table - Level 3 Removed

ROW INDEX(S)	REACHABILITY SET R(S)	ANTECEDENT SET A(S)	SET PRODUCT R(S) ∩ A(S)
1	1	1	1

Objective Number

Objective Number	15	6	4	12	2	16	3	5	7	8	9	10	11	13	14	17	18	1	
15	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	Level 1
6	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
4	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	Level 2
12	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
2	1	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	Level 3
16	1	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	
3	1	0	1	1	0	0	1	1	1	1	1	1	1	1	1	1	1	0	
5	1	0	1	1	0	0	1	1	1	1	1	1	1	1	1	1	1	0	
7	1	0	1	1	0	0	1	1	1	1	1	1	1	1	1	1	1	0	
8	1	0	1	1	0	0	1	1	1	1	1	1	1	1	1	1	1	0	
9	1	0	1	1	0	0	1	1	1	1	1	1	1	1	1	1	1	0	
10	1	0	1	1	0	0	1	1	1	1	1	1	1	1	1	1	1	0	
11	1	0	1	1	0	0	1	1	1	1	1	1	1	1	1	1	1	0	
13	1	0	1	1	0	0	1	1	1	1	1	1	1	1	1	1	1	0	
14	1	0	1	1	0	0	1	1	1	1	1	1	1	1	1	1	1	0	
17	1	0	1	1	0	0	1	1	1	1	1	1	1	1	1	1	1	0	
18	1	0	1	1	0	0	1	1	1	1	1	1	1	1	1	1	1	0	
1	1	1	1	1	0	0	1	1	1	1	1	1	1	1	1	1	1	1	Level 4

Figure 10. The Modified Reachability Matrix Containing Four Hierarchical Levels

The dashed lines within the level 3 submatrix identify constituents, or interior links, within that level. The largest of the three constituents is called a universal submatrix because it contains all 1s indicating that each of the associated objectives in that submatrix are mutually reachable to each other. In the literature, this is more commonly known as a *maximal cycle*. The dashed lines to the left of each of the four heavy-lined submatrices outline what we in the project have termed *communication submatrices* which essentially describe how one level communicates with the level above it. These submatrices become useful in determining paths in the eventual digraph.

Step 6

At this step, all required information exists in the modified reachability matrix to compute the digraph. Warfield has noted that a given reachability matrix does not produce a unique digraph. (20) This implies that more than one digraph can be constructed from the reachability matrix of Figure 10. All the digraphs constructed in this project are generalized digraphs which are actually composites of all the possible digraphs contained in the reachability matrix.

The construction method is illustrated using Figure 10 for reference. The following illustration represents the process used to manually construct the digraph from the modified matrix. The computer algorithm accomplishes this entire process in a manner which is *transparent* to the user. The process is presented here for those who wish to develop their own algorithms.

Begin by laying out each of the four levels identified by the heavy-lined submatrices (levels) and starting at the bottom of the matrix. Level 4 contains only row 1. Level 3, the largest level, contains rows 2, 16, 3, 5, 7, 8, 9, 10, 11, 13, 14, 17, and 18. Level 2 contains rows 4 and 12. And level 1, the highest level, contains rows 15 and 6. By referring to the dashed submatrices (communication submatrices) to the left of each level submatrix, connecting paths between the objectives of one level and the objectives of each higher level on the hierarchy can be determined.

For example, the level 4 communication submatrix (bottom row in Figure 10) has the following pattern: (0 0 1 1 1 1 1 1 1 1 1 1 1). This pattern matches the patterns of rows 3, 5, 7, 8, 9, 10, 11, 13, 14, 17, and 18 in the third level. Thus, a connecting path from objective 1 to each of those mentioned above can be drawn on the digraph. Note here that no connecting path exists between objectives 1 and 16 or 1 and 2 because their communication patterns do not match.

On level 3, there are three separate parts (or *constituents*) within the level. One constituent is composed of objective 2; another is composed of objective 16; and the third is composed of objectives 3, 5, 7, 8, 9, 10, 11, 13, 14, 17, and 18. As stated earlier, this third constituent is called a maximal cycle. Thus, interconnection paths can be drawn on the digraph between objectives 3, 5, 7, 8, 9, 10, 11, 13, 14, 17, and 18.

To get from level 3 to level 2, the level 3 communication submatrix (the dashed matrix to the right of the level 3 submatrix) is analyzed gainst the level 2 submatrix. Since level 3, as was shown, possesses three separate and unique constituents, there are three unique communication patterns to consider. For instance, the maximal cycle constituent (the largest within level 3) has a communication pattern of (1 1). This pattern matches the 1s in both row 12 and row 4 of level 2. Thus, connection paths can be drawn on the digraph from any member of the level 3 constituent to each of objective 12 and 4 in level 2. It is suggested here that only one member from level 3 be connected to level 2 since each member of the maximal cycle is already connected to all others in that constituent (by virtue of it being a maximal cycle set). In addition, a single connecting path allows the resulting digraph to appear considerably more simple. However, the choice to do or not to do this is completely arbitrary. Also in level 3, the row 16 communication pattern (0 1) matches only row 12 in level 2, while row 2's communication pattern matches only row 4. Thus, two more connecting paths can be drawn. Continuing in this manner, a complete digraph can be drawn to represent the reachability matrix. The finished digraph is shown in Figure 11.

We should digress here for a moment to make an important point. Tatsuoka contends that a digraph can be constructed merely by analyzing the self-interaction matrix (he terms it the *adjacency matrix*). This writer, however, believes that although Tatsuoka's contention is valid and logically consistent, the adjacency matrix contains only enough information for one unique digraph, whereas, the reachability matrix yields a more generalized digraph. In a manner of speaking, the reachability

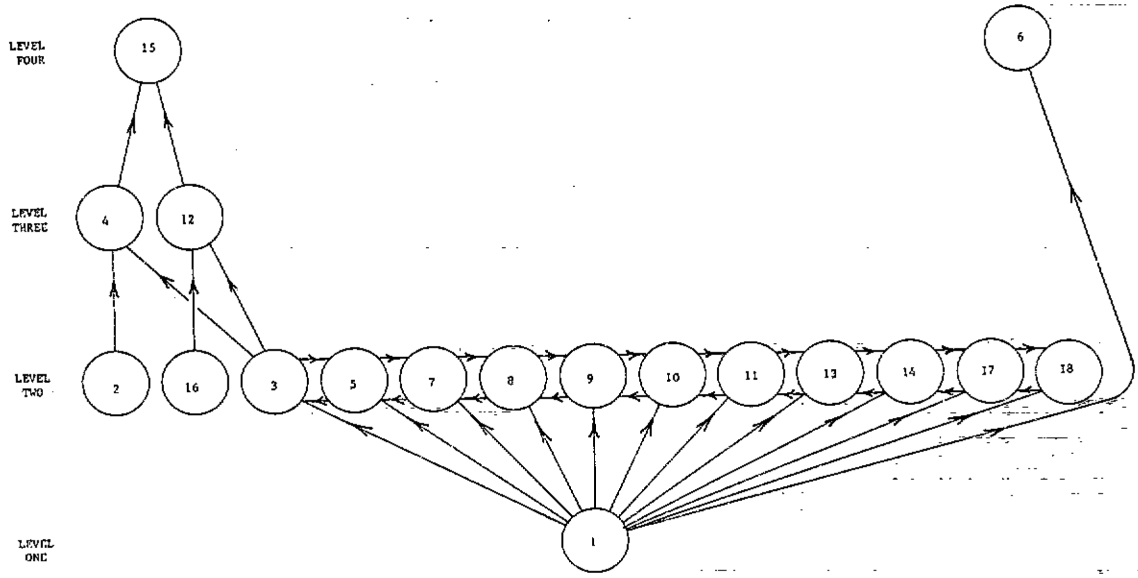


Figure 11. A Digraph for the Pilot Curriculum

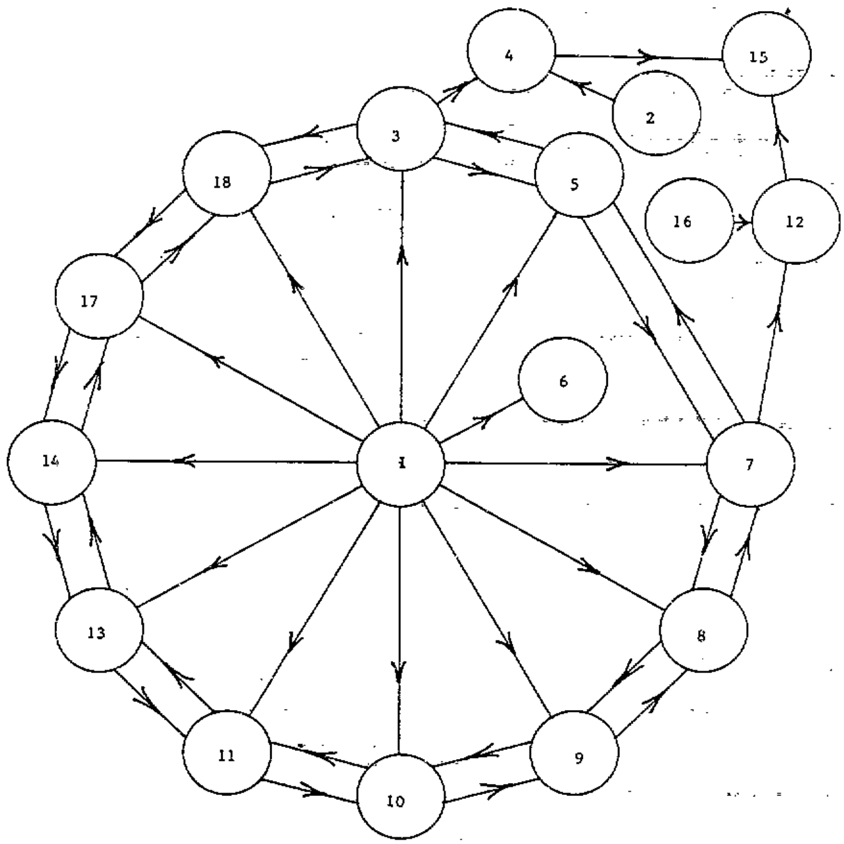


Figure 12. An Alternative Form for the Digraph of Figure 11

matrix is a composite of a family of adjacency matrices. This is intuitively true since the reachability matrix is computed by raising the adjacency matrix to consecutive powers.

This can also be shown mathematically. Take, for example, the number 16. There are two numbers whose consecutive products will equal 16 - they are, of course 2 (2x2x2x2) and 4 (4x4). Both consecutive products result in the same number - 16. However the original numbers 2 and 4 are obviously not the same. This same analogy transfers to the problem of whether to use the adjacency or reachability matrix in determining a suitable instructional digraph.

A digraph computed from the adjacency matrix will undoubtedly be more simplified than one derived from the reachability matrix, although the level of complexity does not begin to become a hinderance until very large numbers of objectives (40 or more) are to be manipulated. In other words, the digraph derived from the reachability matrix will usually contain more paths than that computed from the adjacency matrix.

Each path on the digraph can be thought of as a legitimate transition from one objective to another within the curriculum. Looking at the digraph in this way, one can begin to see that by developing such a transition-laden digraph yields a more fertile data base from which alternative instructional sequences may be derived.

In the pilot project, it was recognized that the digraph of Figure 11 could be redrawn to yield more meaningful information to the curriculum designer. This alternate digraph is shown in Figure 12. This type of digraph is called a *minimum edge representation* of the hierarchy (22).

Note in this figure that the maximal cycle constituent of level 3 is represented by a bi-directional circle interlocked via objective 1. From the viewpoint of the actual course curriculum there is, in fact, a great deal of coherence among objectives 1, 3, 5, 7, 8, 9, 10, 11, 13, 14, 17 and 18. Thus, it is not coincidental that such a pattern has emerged. Note also that objective 6 can only be reached by objective 1. Therefore, any instruction concerning objective 6 must rely on information presented during instruction on objective 1 - if, that is, the students are to see a logical transition from one lesson to the next. Since objectives 1 and 6 appear isolated from the rest, instruction relating to these two objectives could very easily form a module of instruction. Indeed, other modules begin to emerge from the digraph upon closer inspection. It will be left to the reader who gains pleasure from such activity to discover these other modules.

This in and of itself is a remarkable tool for the curriculum designer - to be able to identify "natural" groupings of objectives via mathematical analysis. However, this is merely a fringe benefit of the matrix analysis technique. As the computer analyzes the reachability matrix and its communication patterns, a data base is formed which contains all possible legitimate transitions from any given objective to any other. Once computed, this data base is used for comparison with a user's transition selections. A user can, in fact, experiment with various instructional sequences - transitioning from one objective to another until an entire course is created. By comparing user-selected transitions with the permissible transitions stored in memory, the computer will inform the user if a particular instructional sequence is, or is not, advisable. It will even printout the sequence created by the user in hard copy, if a printer is attached. Figure 13 is an actual, though partial, computer printout of the interactive instructional sequence creation routine.

```
WITH WHICH OBJECTIVE WOULD YOU LIKE TO START THE SEQUENCE?  1

THE FOLLOWING TRANSITIONS ARE ADVISED.  CHOOSE ONE OR ENTER ZERO TO END THE SEQUENCE:

        3 5 7 8 9 10 11 13 14 17 18    ? 7

OK.  1 → 7 →

THE FOLLOWING TRANSITIONS ARE ADVISED.  CHOOSE ONE OR ENTER ZERO TO END THE SEQUENCE:

        3 5 8 9 10 11 13 14 17 18    ? 3

OK.  1 → 7 → 3 →

THE FOLLOWING TRANSITIONS ARE ADVISED.  CHOOSE ONE OR ENTER ZERO TO END THE SEQUENCE:

        4 5 7 8 9 10 11 12 13 14 17 18    ? 4

OK.  1 → 7 → 3 → 4 →

THE FOLLOWING TRANSITIONS ARE ADVISED.  CHOOSE ONE OR ENTER ZERO TO END THE SEQUENCE:

        15    ? 2

THIS OBJECTIVE IS OUT OF SEQUENCE.  DO YOU STILL WANT TO SELECT IT (Y OR N)?  Y

OK.  HOWEVER, IT WILL BE FLAGGED TO REMIND YOU IT'S OUT OF SEQUENCE.

   1 → 7 → 3 → 4 → [2] →

THE FOLLOWING TRANSITIONS ARE ADVISED. CHOOSE ONE OR ENTER ZERO TO END THE SEQUENCE:

        4    ? 4

OK.  1 → 7 → 3 → 4 → [2] → 4 →

THE FOLLOWING TRANSITIONS ARE ADVISED. CHOOSE ONE OR ENTER ZERO TO END THE SEQUENCE:

        15    ?  8

THIS OBJECTIVE IS OUT OF SEQUENCE. DO YOU STILL WANT TO SELECT IT (Y OR N)?  Y

OK.  HOWEVER IT WILL BE FLAGGED TO REMIND YOU IT'S OUT OF SEQUENCE.

   1 → 7 → 3 → 4 → [2] → 4 → [8] →

THE FOLLOWING TRANSITIONS ARE ADVISED.  CHOOSE ONE OR ENTER ZERO TO END THE SEQUENCE:

        3 5 7 9 10 11 13 14 17 18    ?  18

OK.  1 → 7 → 3 → 4 → [2] → 4 → [8] → 18 →

THE FOLLOWING TRANSITIONS ARE ADVISED. CHOOSE ONE OR ENTER ZERO TO END THE SEQUENCE:

        3 5 7 8 9 10 11 13 14 17    ?  16

THIS OBJECTIVE IS OUT OF SEQUENCE. DO YOU STILL WANT TO SELECT IT (Y OR N)?  Y

OK.  HOWEVER IT WILL BE FLAGGED TO REMIND YOU IT'S OUT OF SEQUENCE.

   1 → 7 → 3 → 4 → [2] → 4 → [8] → 18 → [16] →

THE FOLLOWING TRANSITIONS ARE ADVISED. CHOOSE ONE OR ENTER ZERO TO END THE SEQUENCE:

        12    ?  0

OK.  HERE IS THE CURRICULUM SEQUENCE YOU HAVE CREATED:

   1 → 7 → 3 → 4 → [2] → 4 → [8] → 18 → [16]

DO YOU WANT TO CREATE ANOTHER SEQUENCE (Y OR N)?  N

OK. BYE FOR NOW.
```

Figure 13. An Interactive Instructional Sequence Dialog Between User and Computer

LIMITATIONS WITHIN A CURRICULAR SYSTEM

Naturally, the ultimate decision as to how a curriculum is to be arranged rests with the managers or administrators of the curriculum. It has been this writer's experience that a major deficiency of front-end analysis is the inadequate attention paid to the interplay among the numerous internal and external constraints and limitations placed upon a given curriculum. Limitations such as facilities, personnel, time, money, social factors, etc., if not anticipated in advance of establishing a curriculum sequence, could result in the ultimate alteration of an otherwise logical instructional sequence.

It is admittedly a complex task to consider the effects of all possible limitations affecting a curriculum without some means to organize and manipulate very large amounts of data. The project described in this paper has illustrated a method with the power to expand and accomodate the analysis of such limitations -- and thus produce an ultimate curriculum sequence which is sensitive to those limitations. The work on this expansion forms the basis for Phase II of this project planned to be completed later next year.

The ultimate goal of this project is to develop an integrated curriculum for 16 closely related courses. Each course possesses certain characteristic limitations which are either reinforced or overcome by the remaining courses. It is desired that this project will produce a curriculum which will reconcile the majority of those limitations. Such a goal is common to curriculum designs both in the military and civilian sectors of education. In that respect, at least, those of us associated with this project feel a bond with educators in every sector of society.

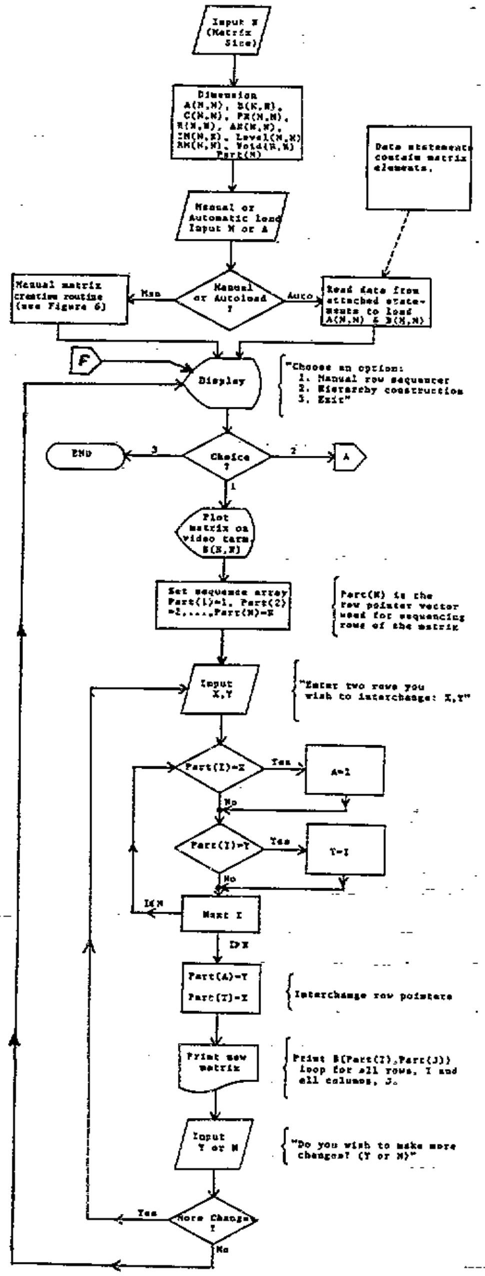
ABOUT THE AUTHORS

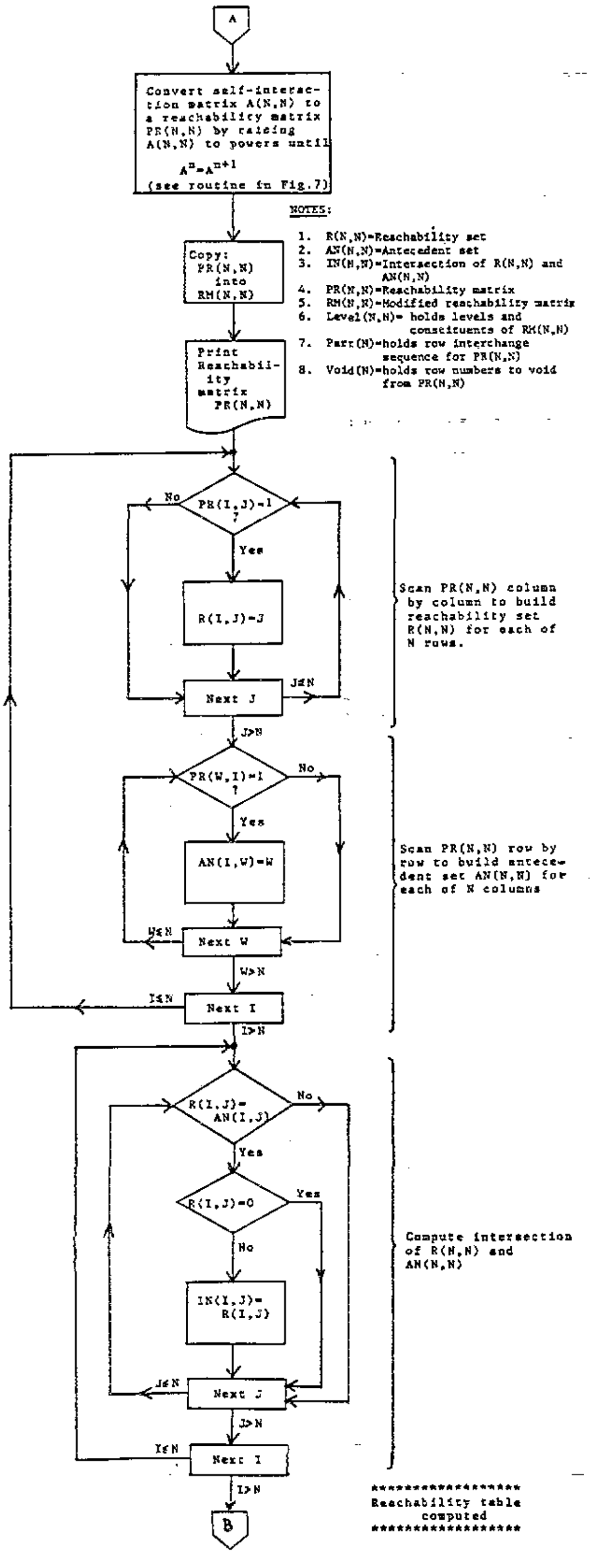
Mr. Thomas R. Renckly, Education Specialist, U.S. Navy Recruiting Command. Coordinator for Curriculum Design at the Navy Recruiting Orientation Unit, Educational Research and Systems Development Department, Orlando, Florida. Currently pursuing a doctorate in Curriculum and Instruction at the University of Florida.

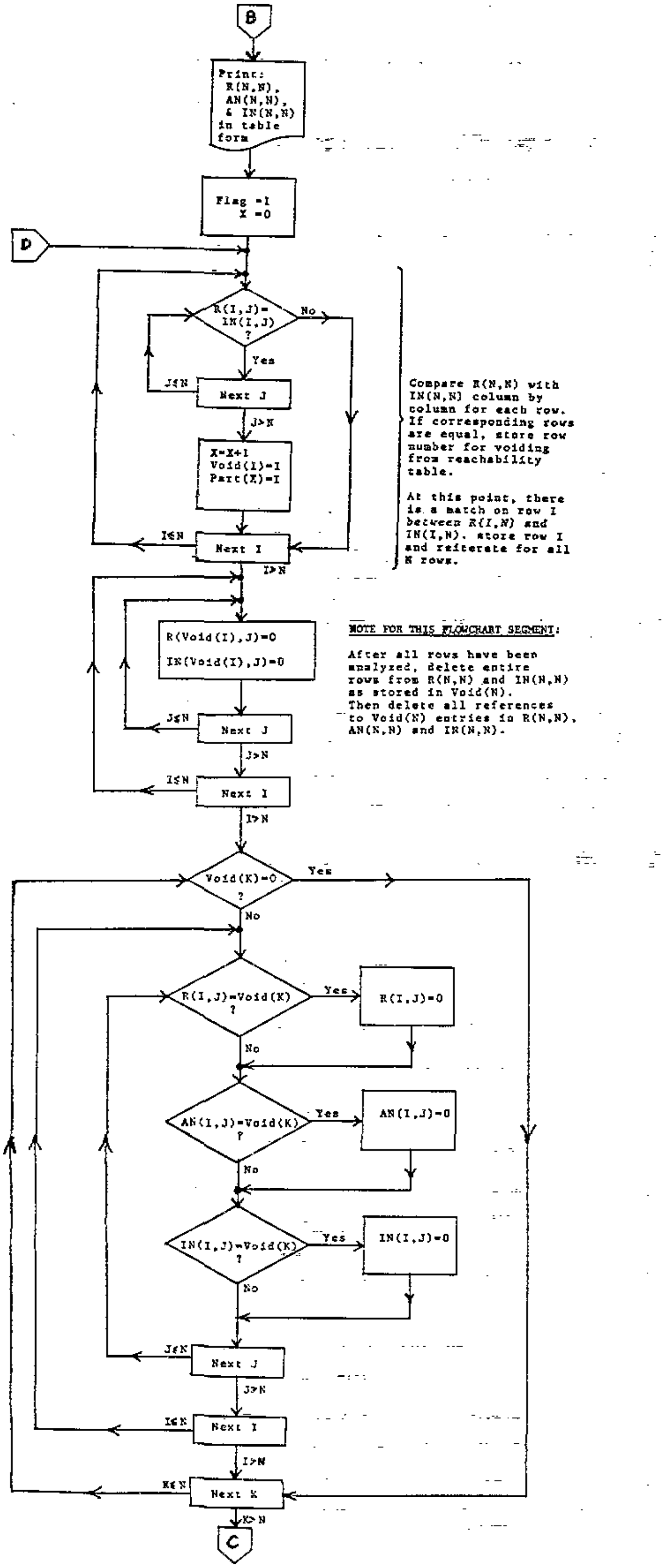
Dr. Gary Orwig, Asst. Professor in Instructional Technology with the College of Education at the University of Central Florida. Teaches courses in computer applications in instructional technology; actively engaged in research concerning interactive video and instructional design.

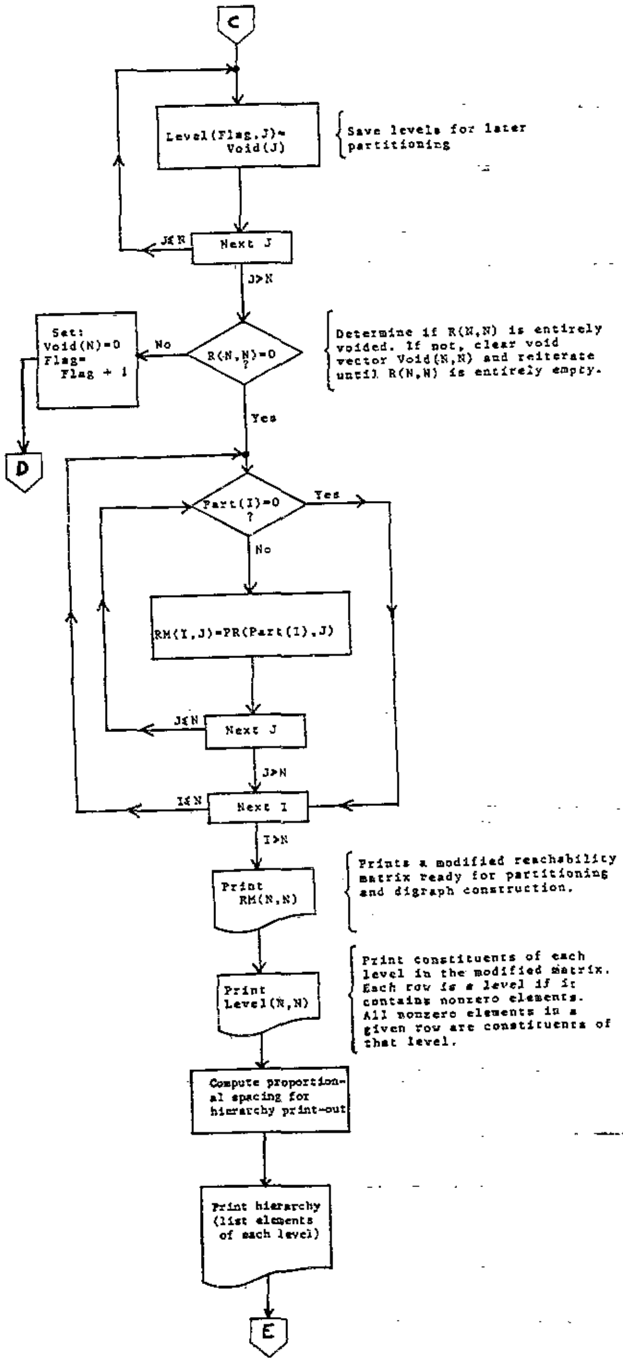
APPENDIX

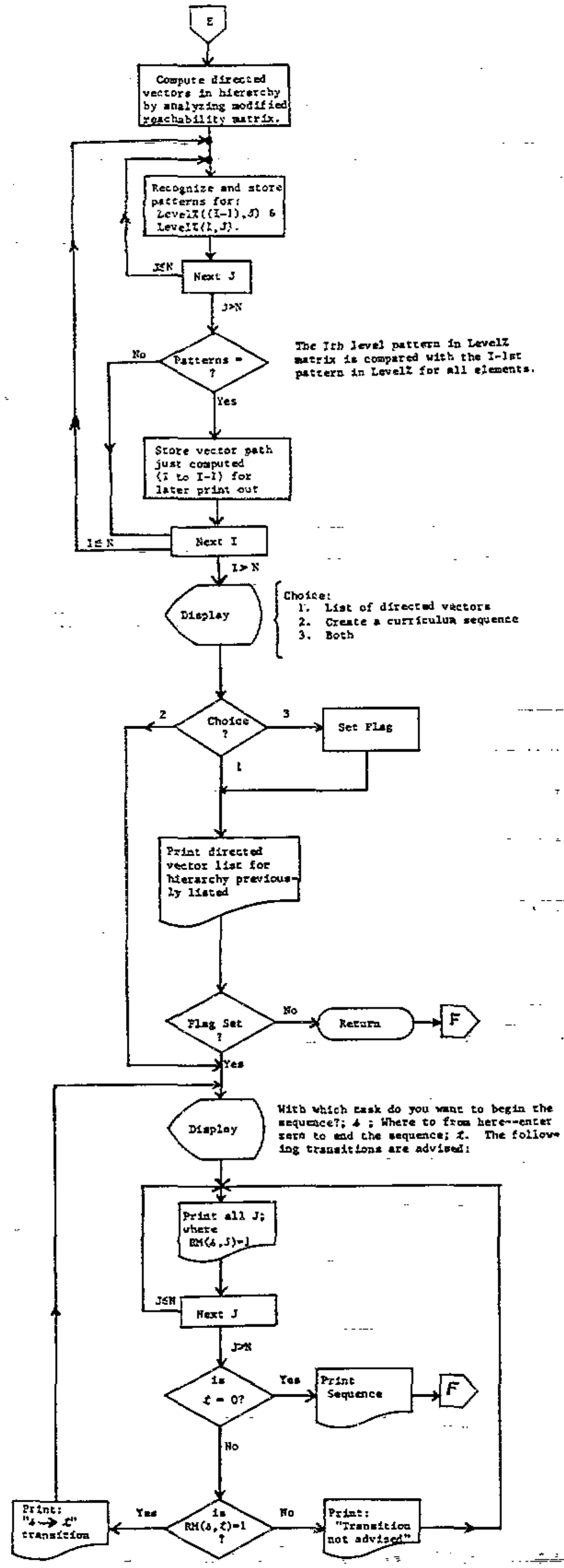
A Detailed Computer Flowchart for Developing A Sequence Digraph From a Set of Curriculum Objectives (continued next 2 pages)




A
Convert self-interaction matrix A(N,N) to a reachability matrix PR(N,N) by raising A(N,N) to powers until $A^n = A^{n+1}$ (see routine in Fig.7)
NOTES:
1. R(N,N)=Reachability set
2. AN(N,N)=Antecedent set
3. IN(N,N)=Intersection of R(N,N) and AN(N,N)
4. PR(N,N)=Reachability matrix
5. RM(N,N)=Modified reachability matrix
6. Level(N,N)= holds levels and constituents of RM(N,N)
7. Part(N)=holds row interchange sequence for PR(N,N)
8. Void(N)=holds row numbers to void from PR(N,N)
Copy: PR(N,N) into RM(N,N)
Print Reachability matrix PR(N,N)
PR(I,J)=1 ?
No
Yes
R(I,J)=J
Next J
J≤N
J>N
Scan PR(N,N) column by column to build reachability set R(N,N) for each of N rows.
PR(W,I)=1 ?
No
Yes
AN(I,W)=W
Next W
W≤N
W>N
Scan PR(N,N) row by row to build antecedent set AN(N,N) for each of N columns
Next I
I≤N
I>N
R(I,J)= AN(I,J)
No
Yes
R(I,J)=0
Yes
No
IN(I,J)= R(I,J)
Compute intersection of R(N,N) and AN(N,N)
Next J
J≤N
J>N
Next I
I≤N
I>N
B
****
Reachability table computed
****


B
Print: R(N,N), AN(N,N), & IN(N,N) in table form
Flag =1
X =0
D
R(I,J)= IN(I,J) ?
No
Yes
Next J
J≤N
J>N
X=X+1
Void(I)=I
Part(X)=I
Next I
I≤N
I>N
Compare R(N,N) with IN(N,N) column by column for each row. If corresponding rows are equal, store row number for voiding from reachability table.
At this point, there is a match on row I between R(I,N) and IN(I,N). store row I and reiterate for all N rows.
R(Void(I),J)=0
IN(Void(I),J)=0
Next J
J≤N
J>N
Next I
I≤N
I>N
NOTE FOR THIS FLOWCHART SEGMENT:
After all rows have been analyzed, delete entire rows from R(N,N) and IN(N,N) as stored in Void(N). Then delete all references to Void(N) entries in R(N,N), AN(N,N) and IN(N,N).
Void(K)=0 ?
Yes
No
R(I,J)=Void(K) ?
Yes
R(I,J)=0
No
AN(I,J)=Void(K) ?
Yes
AN(I,J)=0
No
IN(I,J)=Void(K) ?
Yes
IN(I,J)=0
No
Next J
J≤N
J>N
Next I
I≤N
I>N
Next K
K≤N
K>N
C


C
Level(Flag,J)= Void(J)
Save levels for later partitioning
Next J
J≤N
J>N
Set: Void(N)=0 Flag= Flag + 1
No
R(N,N)=0 ?
Determine if R(N,N) is entirely voided. If not, clear void vector Void(N,N) and reiterate until R(N,N) is entirely empty.
D
Yes
Part(I)=0 ?
Yes
No
RM(I,J)=PR(Part(I),J)
Next J
J≤N
J>N
Next I
I≤N
I>N
Print RM(N,N)
Prints a modified reachability matrix ready for partitioning and digraph construction.
Print Level(N,N)
Print constituents of each level in the modified matrix. Each row is a level if it contains nonzero elements. All nonzero elements in a given row are constituents of that level.
Compute proportional spacing for hierarchy print-out
Print hierarchy (list elements of each level)
E


E
Compute directed vectors in hierarchy by analyzing modified reachability matrix.
Recognize and store patterns for: LevelZ((I-1),J) & LevelZ(I,J).
Next J
J≤N
J>N
Patterns = ?
No
Yes
The Ith level pattern in LevelZ matrix is compared with the I-1st pattern in LevelZ for all elements.
Store vector path just computed (I to I-1) for later print out
Next I
I≤N
I>N
Display
Choice:
1. List of directed vectors
2. Create a curriculum sequence
3. Both
Choice ?
2
3
1
Set Flag
Print directed vector list for hierarchy previously listed
Flag Set ?
No
Return
F
Yes
Display
With which task do you want to begin the sequence?; 𝓈 ; Where to from here--enter zero to end the sequence; 𝓉. The following transitions are advised:
Print all J; where RM(𝓈,J)=1
Next J
J≤N
J>N
is 𝓉 = 0?
Yes
Print Sequence
F
No
is RM(𝓈,𝓉)=1 ?
Yes
No
Print: "𝓈→𝓉" transition
Print: "Transition not advised"

BIBLIOGRAPHY

1. Hill, Douglas J. and Warfield, John N. "Unified Program Planning," IEEE Transactions on Systems, Man, and Cybernetics, SMC-2, No. 5, Nov. 1972, pp. 610-621

2. Delp, P., Thesen, A., Motiwalla, J., and Seshadri, N. System Tools for Project Planning, International Development Institute, University of Indiana, Bloomington, Indiana, 1977, pp. 92-103

3. Sato, Takahiro. "Hierarchical Display of Networks of Teaching Elements Using the Interpretive Structural Modeling Method" IECE Transations on Educational Technology, 1978-04, pp. 27-30 (manuscript in Japanese)

4. Sato, Takahiro. "Determination of Hierarchical Networks of Instructional Units Using the Interpretive Structural Modeling Method," Educational Technology Research, No. 3, 1979, pp. 67-75 (In English)

5. Bobbitt, Franklin. How To Make A Curriculum, Boston: Houghton Publishing Co., 1924

6. Nadler, Gerald. Work Design: A Systems Concept, Homewook, Illinois: Richard D. Irwin Inc., (Revised) 1970, pp. 21-47

7. Thomas C.A., Davies, I.K., Openshaw, D., and Bird, J. Programmed Learning in Perspective: A Guide To Program Writing, Chicago: Educational Methods Inc., 1963

8. Davies, Ivor K. Competency Based Learning: Technology, Management and Design, New York: McGraw-Hill, 1973, pp. 92-102

9. Esseff, Peter J. and Mary S. Educational Systems For The Future (Mediated Presentation), 1978, (4th printing)

10. Op. Cit., Hill and Warfield, p 613

11. Op. Cit., Hill and Warfield, p 613

12. Op. Cit., Nadler, p 524

13. Op. Cit., Hill and Warfield, p 614

14. Warfield, John N. Structuring Complex Systems, Columbus: Battelle Institute, Monograph No. 4, April, 1974

15. Warfield, John N. personal communication, February, 1981

16. Op. Cit., Warfield, Chapter 2, p 9

17. Warfield, John N. Societal Systems: Planning, Policy and Complexity, New York: John Wiley & Sons, 1976, pp. 276-284

18. Op. Cit., Warfield, Structuring Complex Systems, Chapter 4, pp. 1-8; Chapter 5, pp. 1-9

19. Tatsuoka, Maurice M. "Recent Psychometric Developments in Japan: Engineers Grapple with Educational Measurement Problems," presented at ONR Contractor's Meeting on Individualized Measurement, Columbia, Missouri, September 19, 1978, p. 6

20. Op. Cit., Warfield, Structuring Complex Systems, Chapter 2, p 9

21. Tatsuoka, Maurice M. personal communications, June 1981; (reply) July 1981

22. Op. Cit., Warfield, Structuring Complex Systems, Chapter 2, p 9