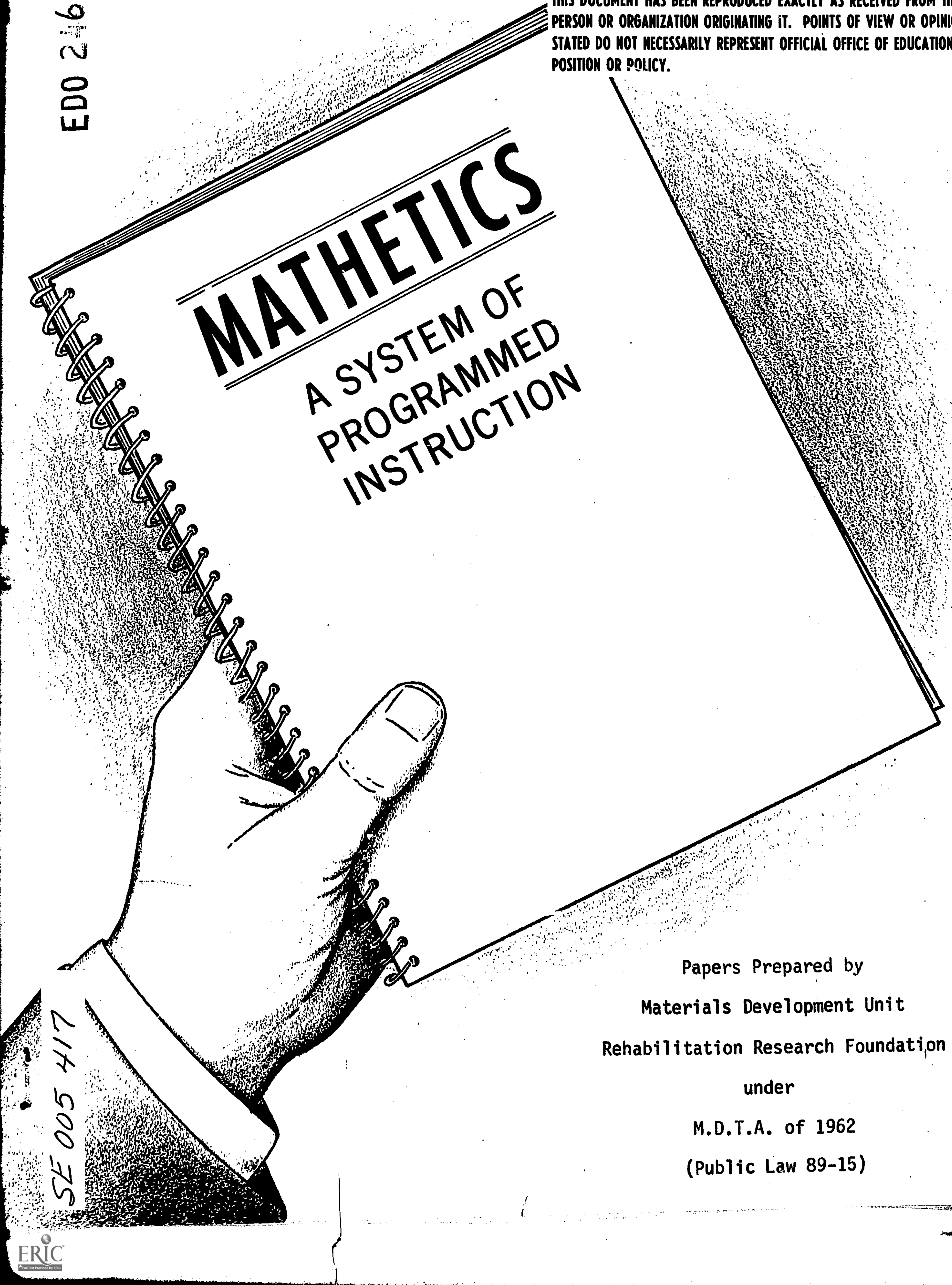
DOCUMENT RESUME

ED 024 601 SE 005 417

Mathetics, a System of Programmed Instruction.
Rehabilitation Research Foundation, Elmore, Ala. Materials Development Unit.
Pub Date Apr 66
Note-61p.
EDRS Price MF-$0.50 HC-$3.15
Descriptors-*Curriculum Development, Industrial Training, Instruction, Program Descriptions, *Programed Instruction, Training
Identifiers-Alabama, Elmore, Materials Development Unit, Rehabilitation Research Foundation

The four papers contained in this document were written by three members of the Materials Development Unit, Rehabilitation Research Foundation, Elmore, Alabama. These papers were selected for joint distribution to present the authors' experiences with the mathetical system in light of current mathetical programing activities. "The Two Meanings of Mathetics" and "Mathetics - The Ugly Duckling Learns to Fly" are an overview of the mathetical system and the techniques employed by the mathetical analyst and writer. "Mathetics in Industrial and Vocational Training" describes the activities of major mathetical programing units in the United States. Finally, "The Development and Production of Mathetical Programs - A Case Study" describes the major production procedures of programing units. (RP)



Papers Prepared by

Materials Development Unit

Rehabilitation Research Foundation

under

M.D.T.A. of 1962

(Public Law 89-15)

INTRODUCTION

The four papers contained herein were written by three members of the Materials Development Unit, Rehabilitation Research Foundation, Elmore, Alabama. They were selected for joint distribution to present our experience with the mathetical system in the light of current mathetical programming activities.

The first three papers were delivered at the National Programmed Learning Conference, Leicestershire, England, in April, 1966. The fourth was presented to the Fourth Annual Convention of the National Society of Programmed Instruction, St. Louis, Missouri, April, 1966.

"The Two Meanings of Mathetics" and "Mathetics: The Ugly Duckling Learns to Fly" are an overview of the mathetical system and the techniques employed by the mathetical analyst and writer. "Mathetics in Industrial and Vocational Training" describes the activities of major mathetical programming units in the United States. "The Development and Production of Mathetical Programs: A Case Study" describes the major production procedures of our programming unit.

Mathetics, by name, is not being practiced by a large number of programmers; however, the techniques that have been utilized by mathetical lesson writers are gradually being adopted by many other writers of programmed materials.

The usual product of the mathetical system is a programmed text, but the practices and procedures of mathetics are believed to be applicable to education in a much broader view.

THE TWO MEANINGS OF MATHETICS[1]

J. H. Harless
Chief Programmer
Rehabilitation Research Foundation
Elmore, Alabama

Introduction

Mathetics has been the subject of controversy since its first description by Thomas F. Gilbert early in 1962. Gilbert then defined mathetics as, "...the systematic application of reinforcement theory to the analysis and reconstruction of those complex behavior repertories usually known as 'subject-matter mastery', 'knowledge', and 'skill'."[2]

If one were to poll the programming world in an attempt to derive a descriptive definition of mathetics, he would discover that virtually nothing of a specific nature is generally known of this system, except that "mathetics teaches everything backwards."

---

[1]This paper is part of a three unit presentation on mathetics. For a further explanation of the characteristics of mathetical lessons, and a survey of their usage in the United States, see Michael T. McGaulley. "Mathetics in Industrial and Vocational Training." Rehabilitation Research Foundation.

For a description of the organization and working procedures of the programming unit of the Rehabilitation Research Foundation, see Samuel J. Cassels, III. "The Development and Production of Mathetical Programs: A Case Study."

[2]Gilbert, Thomas E., "Mathetics: The Technology of Education," Journal of Mathetics, Vol. 1, No. 1, January, 1962, p. 8.

This misconception that mathetical lessons (programs) necessarily and invariably teach retrogressively, stems from one of the more unusual strategies sometimes employed by a mathetical lesson writer. Although this special technique represents only one of the many devices used, "chaining" of the behaviors has become synonomous with mathetics itself. Present-day matheticists employ chaining under special circumstances contingent upon many variables, as we shall see later.

The goal of the proponents of mathetics is to evolve a genuine technology of education by welding the concepts of behavioral science to the effective practices and procedures that have always been utilized by <u>good</u> teachers, and more recently, by good programmers. Therefore, mathetics is not "new" in the usual connotation of the word.

A technology, by definition, is not new in fundamentals, but rather is the systematic application of concepts and principles for new functions or the improvement of the methodology for old ones. Therefore, the word "mathetics" has two distinct meanings: First, in its broader implication, mathetics is a complete <u>training system</u> that guides the trainer to description of mastery, and discovery of training deficiencies of a specific population. The system includes guidelines for the analysis of the skills and knowledges to be learned, and specific strategies for overcoming the deficiencies. This remedy of a training deficiency may take the form of a programmed text or other types of training vehicles such as films, slides, role-playing exercises, etc. Also, to make this system complete, a methodology for validation and curriculum implementation has been evolved. Mathetics is secondly a step-by-step procedure for the construction of the actual programmed lesson. This process, like other good programming methods, is devoted to ensuring that only the exact functions of the mathetical teaching unit (the "exercise") are served.

It has been demonstrated that the mathetical technology is precise enough to allow two matheticists working independently with the same set of objectives to produce programs identical in many important aspects such as size of step, order of steps, level of simulation, cueing conventions, and general page layout.[3]

The System of Analysis

Rarely does a matheticist set out to "program" any particular topic or block of subject matter. A mathetical program is not a textbook or a chapter in programmed form. The duty of the matheticist is to discover the exact training deficiency in a well-defined population.

In order to discover this deficiency, the matheticist's first task is one of description. This is done in the job analysis when the matheticist interviews specialists in a particular domain

---

[3]The general conclusion that mathetics is only an eclectic system is somewhat contradicted by this fact.

for the purpose of listing all the products[4] produced by that job. These products are then described in terms of the requirements they must meet, the steps of performance in deriving them, and the knowledges and behaviors that are prerequisite to their production.

Having completed the job analysis, the matheticist describes the characteristics of the trainees, noting especially the related knowledge, skills, and academic abilities of the students.

The precise training deficiency is then found by listing the differences between the master and the student. This difference is expressed in the form of training objectives.

With the job and population analyses and objectives as guides, the matheticist writes a "prescription" for the training deficiency. The prescription expresses the steps in producing a product in stimulus-response units called "operants."

Essentially linear programmers write a prescription of behavior when they break the behaviors to be taught into the smallest steps. But at this point, the linear programmer in many instances is ready to write his program without any specific attention to the special nature of the behavior to be taught.[5] The matheticist, on the

---

[4]A "product" is the describable and measurable results of performance. Virtually any task can be described in terms of the products produced by that task.

[5]Happily, many programming groups are beginning to pay more than lip-service to analysis at this point.

other hand, spends much time in a detailed behavioral analysis to determine the characteristics of his prescription.

This analysis of the prescription has three overlapping purposes: (1) to discover the learning problems inherent in the behaviors, such as interactive inhibitions and the amount of generalization needing to be taught; (2) to discover learning problems inherent in the intended student population; (3) to determine the optimum sequence for teaching the behaviors called for in the prescription. This rigorous and systematic analysis of the behaviors to be learned by the student is the heart of the mathetical system and is the major rationale for mathetics. The programmer's behavior is guided by this analysis which forces him to examine all aspects of the material he will ultimately teach in the program. The analysis gives him a scientific basis on which to make his selection of teaching strategies.

Using the results of the analysis, the matheticist reexamines his prescription and answers this question: How many of the stimulus-response units (operants) will be presented to the student at one time? In other words, what is the "operant span" of this population? This concept of "operant spans" is a rather radical departure from the usual concept that the teaching units (frames) of programmed instruction should be in extremely small steps.[6] Step sizes designed to meet the requirements of the individual

---

[6]Even though this theorem is undergoing a change in P. I., programs are still written in relatively small steps with little regard to the characteristics of the design population.

population have demonstrated that the boredom resulting from homogeneously small frames has been eliminated in mathetical programs. Parenthetically, the physical bulk of the programs has been drastically reduced for production economy, quicker terminal reinforcement, and time savings for the student.

The matheticist then summarizes his decisions in a detailed lesson plan to serve as a guide for the actual writing of the program. Depending on his findings, the matheticist may decide to employ one or more of several special strategies for increased teaching and learning efficiency.

One of the more interesting and controversial of these special strategies is the process of "chaining." This procedure, which has become erroneously synonomous with mathetics, is the arrangement of the operants so the student is presented the last step of performance first, the next to last step second, and so forth until all operants have been taught to the student.

For example, a mathetical programmer teaching long division may demonstrate to the student how to perform the last operation (subtraction) first; then, he would demonstrate how to multiply, while cueing the student to subtract to complete the operation; finally, he would demonstrate the first step (short division), cue the student to multiply, and release him without cues to subtract to complete the long division again.

By this chaining process the student is constantly completing the operation while adding new knowledge and behavior to his repertory. In theory, chaining allows him to work for and receive the terminal reinforcement many times while learning the steps of the operation. Also, by using the chaining process, the student can relate the particular step he is learning to the entire operation and can see how each part is germane to the final product.

Thus mathetics is first of all a procedure for analysis of the behavior to be taught.

In a second amd more specific meaning, mathetics is a scientific technique of program construction.

<u>Exercise Construction</u>

There are a finite and describable number of functions that each teaching unit (exercise) of mathetics must serve: 1) the student must be put under the control of the discriminative stimulus (or S^D); 2) the student must observe the stimulus by directing his attention to the S^D and by classifying the S^D from all other discriminative stimuli; 3) the student must be given instructions on how and what response to make in the presence of the discriminative stimulus; 4) the student must make the required response in presence of the S^D; 5) the student must place the S^D in context with the entire operation being taught.[7]

---

[7]Gilbert, Thomas F., "Mathetics: II. The Description of Teaching Exercises," Journal of Mathetics, Vol. I., No. 2, April, 1962, pp. 7-56.

Some exercises require additional functions depending on their location in the sequence; a "prompt" (or cueing) of a previously demonstrated exercise and/or the presenting of the S^D of a previously demonstrated and cued exercise for the student to perform without aid is necessary in these exercises.

The matheticist examines the exercises he has written and synthesizes the entire operation into verbal statements in step form. He also attempts to discover similar behaviors already known to the student population. The matheticist uses this archetype behavior as an analytical homology or analogy for the student to use to relate the new skills or knowledge he is to learn in the exercises. With the synthesized steps and archetype, the matheticist constructs "understanding" exercises to precede the "performance" exercises. These "understanding" exercises give the student an overview of the behaviors he is to learn. They are designed to provide him with a "selective looking behavior" in the "performance" exercises.

The matheticist will also spend considerable time writing and rewriting these "understanding" and "performance" exercises to insure that the language is suited for the population, that no irrelevant subject matter is introduced, and that the context of each behavior is always set for the student.

At this point the matheticist has completed only two-thirds of his work. The draft program is checked by a subject-matter and curriculum specialists for content validation. Any changes indicated are made.

The matheticist then begins trying out the program on actual members of the design population. This exacting procedure is performed to validate or correct any decisions made by the matheticist in operant span, sequence, wording, etc.

Matheticists, like other good programmers, take the point of view that a motivated student is seldom at fault when a program fails to teach him; therefore, the student is the central figure in the analytical, design, and tryout phases of the mathetical process. Students' responses, questions, and comments are carefully noted in this tryout process and revisions are made according to the dictates of the <u>student</u>.

When the matheticist is empirically confident that the program has undergone sufficient tryouts and revisions to meet the requirements of the objectives, the program is field-tested on large numbers of the design population under the conditions of intended use.

Summary

Mathetics, therefore, is a complete training system that gives the programmer; (1) a guide for determining <u>what</u> to teach, (2) a basis for making teaching strategy decisions, (3) a detailed procedure for constructing a program.

Mathetics is somewhat eclectic in nature, but is unique in application, if not in principles. All inclusively, mathetics is a step toward a technology of education.

MATHETICS IN INDUSTRIAL AND VOCATIONAL TRAINING

Michael T. McGaulley, LL.B.
Rehabilitation Research Foundation
Elmore, Alabama

This paper is part of a three unit presentation on mathetics.[1]

In describing the effect of mathetics on vocational and industrial training, I shall (1) make mention of the major mathetical programming units and the lessons they have produced, (2) point out the significant characteristics of mathetical lessons, and (3) discuss some of the difficulties presently being encountered by mathetical programming units.

I. MATHETICAL PROGRAMMING UNITS NOW IN OPERATION

The Rehabilitation Research Foundation is a non-profit corporation conducting a number of experimental projects[2] in education and human development at Draper Correctional Center near Montgomery, Alabama.

One of the projects is a unique school in which all instruction (other than in basic literacy) is accomplished with programmed instruction. Another of the projects is a vocational school for youthful offenders in which courses are offered in barbering, bricklaying, welding, electrical appliance repair, radio-television repair, automobile servicing, and technical writing. Academic deficiencies of the students are remedied by daily sessions with programmed instruction.

---

[1]For an explanation of the mathetical analysis and exercise writing technique, see J. H. Harless. "The Two Meanings of Mathetics," Rehabilitation Research Foundation.

For a description of the organization and working procedures of the programming unit of the Rehabilitation Research Foundation, see Samuel J. Cassels, III. "The Development and Production of Mathetical Programs: A Case Study, "Rehabilitation Research Foundation.

[2]These experimental projects are supported under the Manpower Development and Training Act, contracts #(M)6068-000 (OMAT) and #82-01-07 (HEW), and by the National Institute of Mental Health, Contract #MH00976-04.

A programming unit was set up adjunctive to the vocational school to provide training materials where needed. The availability of enough programs in academic subjects permitted emphasis to be placed on developing programs to assist in the teaching of manual skills and other knowledges needed in the shops. Mathetical lessons were selected as the primary training medium. Writers and staff were hired and operations began in October, 1964.

As this paper is written, six lessons have been successfully field tested: "Recognizing Electrical Circuit Symbols," "Introduction to the Volt Ohm Milliammeter," "Soldering Leads," "Introduction to Electricity, Part I," "Mixing Mortar," and "Cleaning Carburetor Air Cleaners." Nearly 300 students participated in the tryouts.

The largest number of students took "Recognizing Electrical Circuit Symbols." To determine its limits, the lesson was tested with three distinct populations: (A) students with a year or more of experience or training, (B) students with at least two months of related training or experience, (C) those with no related training or experience. Time required to complete this lesson ranged from 45 minutes to 5 hours, 15 minutes, with an average of around 3 hours. Average gain from pretest to posttest (all groups included) was 79%. Average posttest scores were 95%, 96% and 91% for the three groups, respectively.

Another series of field tests is scheduled for February, 1966. "Recognizing Electronic Circuit Symbols," a companion to the electrical symbols lessons, will be tried then. Part II of "Introduction to Electricity" will also be ready, as will a lesson for bricklaying students in establishing the floor level of a building under construction. Three parts of a package for barbering students are nearly ready for tryout: "Preparation of a Customer" and "The Tools and Areas of a Haircut" have been tested on individual students; the third part of the package is an experimental guide to assist beginners as they give their first few haircuts. Illustrations and directions will be printed on a scroll and installed in a device on or near the barber chair. A large package on the estimation of bricklaying materials has been prepared by two young prison inmates, graduates of our one-year course in technical and program writing.

Ultimately, it is expected that these lessons will be used not only in state vocational schools but in other types of training projects as well, including the Federal Job Corps and Youth Opportunity Centers. They will be published by a governmental agency and will be available to industry.

The consultants who advised the choice of mathetics for the programming unit at the Draper Project cited among their reasons the quality of the lessons produced by the Instructive Communications Unit of the U. S. Public Health Service at the Communicable Disease Center, Atlanta, Georgia. This unit was set up in early 1963 on an experimental basis. Results of the first lesson produced were so satisfactory that the unit was soon given a permanent status under a protege of Dr. Gilbert, the originator of mathetics. Two of the lessons published by this unit were cited among the 16 most outstanding in a 1965 survey conducted by Robert Horn of Columbia University.[3] These were "Amebiasis: Laboratory Diagnosis,"

[3]Robert E. Horn. Excellence in Programmed Instruction - Results of a Survey Identifying 16 Outstanding Programs, Programmed Instruction, 1965, IV, 9.

a three volume lesson for a population of doctors, nurses and laboratory technicians, and "Food-Borne Disease Investigation," intended for public health field workers. Two other programs of the unit also received votes in the survey. These were "Insecticide Formulation," intended for crop sprayers and their supervisors, and "Jet Injector Operation," a lesson to be discussed at more length later in this paper. Four other lessons are in advanced tryout stages and will be published within the next few months.

The third major producer of mathetical programs is TECO Instruction, Inc., of Fort Lauderdale, Florida. TECO is a private consulting firm specializing in the preparation of custom programs for industry. One lesson, "Highway Plan Reading," has been taken by over 5000 employees of state highway departments. Another, on the operations of a bank teller, is gaining wide acceptance. One user alone, First National City Bank of New York, has used it with over 600 employees. "Selected Medical Terminology" was written to teach correct spelling, pronunciation, and usage of difficult technical words to a population of hospital secretaries and typists.

The influence of mathetics has also been spread by the efforts of individual matheticists. Some have taken employment in large commercial and governmental programming centers where they have been able to shape the behavior of their colleagues. Matheticists were employed on a consultant basis to assist in the training of the original core of programmers of the U. S. Air Force.

II. CHARACTERISTICS OF MATHETICAL LESSONS

A. Layout and Response Flexibility

Mathetics is not a format system. No attempt is made to achieve uniformity of style or appearance from lesson to lesson or page to page. The governing rule in setting up an exercise is simply to use whatever is best depending on the characteristics of the behavior to be taught and the abilities of the student population. Thus, an "Exercise" (the teaching unit in mathetics) may look much like a linear frame, or it may appear as a double page spread with all the graphic appeal of a good magazine advertisement.

Most often, paper-and-pencil-type responses are called for in mathetical lessons, particularly if the behavior being taught is primarily verbal; however, student responses involving the use of tools or simulator kits are also commonly found in mathetics.

B. Extensive Use of Illustrations and Simulations

A primary function of the complicated mathetical analysis is that of locating and defining those stimuli to which the student should be taught to respond. In a mathetical exercise, we present the student with this particular stimulus (technically, the "discriminative stimulus") and teach him the correct response to be made to it. Since in most cases the discriminative stimulus is something visual, it is most suitable to present it visually to the student in the lesson. Thus it is that you will find that illustrations, representations of the stimulus, are at the center of nearly every demonstration page in mathetical lessons. (Show first and second transparencies)

Less theoretically, illustrations and simulator kits are used because of the assistance they give the student in transferring his knowledge from the learning situation to the job. A picture is worth a thousand words, and the chance to apply what he has learned by practicing the job may be worth a thousand pictures to the student.

In varying degrees of reality, the responses called for in mathetical lessons are simulated performances of the job being taught. Dry-firing with an innoculation injector or soldering gun is simulated performance that approaches very near to the real thing. However, in most situations, a lesser degree of simulation will work just as well. For example, it is likely that a student who has been directed to imagine working with a certain tool, having a photograph or drawing of the tool to guide him, will be able to use the actual tool properly within a very short time of handling it for the first time. Even mere pencil activity such as marking on a drawing at key points or in a certain sequence, may be sufficiently high-grade simulation of job performance that transfer will be insured.

C. Large Teaching Step Size

> In mathetics, "The principle for determining the size of an exercise is <u>not</u> 'break the material into small parts'; the principle is to require in every exercise as much mastery performance as the student can reasonably negotiate.'"[4]

---

[4] Thomas F. Gilbert. Mathetics: The Technology of Education. Journal of Mathetics, 1962, 1, 25.

The first third of a matheticist's time in working on a lesson is spent performing an analysis of the behavior to be taught and making studies of the abilities and characteristics of the design population. Upon the basis of the findings, he estimates the maximum amount of material the students will be able to grasp in each exercise of the lesson. The philosophy in mathetics is to attempt to push the student to take the largest steps possible.[5]

As a practical consequence of this philosophy, whole-page spreads are common in mathetics. Freedom and flexibility come with the larger spreads. There is ample room to use big, clear pictures and plenty of white space. The matheticist can be typographically playful to capture or recapture student attention.

Very often it "makes sense" to the student to be presented with the whole of a job sequence at one time. (Show third transparency) This spread demonstrates the sequence of behavior in administering a shot with the jet injector. Each operation flows on to the next as part of a cycle. The completeness of the cycle can be preserved by teaching in large steps.

The model teaching sequence in mathetics calls for each stimulus-response relationship ("operant") to be presented at least three times: once in a "demonstration," then in a "prompt," when the student is called upon to make a response with some assistance, and finally in a "release"

---

[5] Any over-estimation of student abilities will be caught and corrected in the tryout cycle. If necessary, remedial exercises can be prepared for the less able students in the target population.

when he responds without the help of any hints in the lesson. It is usually heartening to the student to perform the release, particularly as he proceeds further into the lesson and can work more and more of the job in the release. If the programmer follows the three-part model teaching sequence, much of the repetition that students dislike can be avoided.

III. SOME DISADVANTAGES

A. The Cost of the Behavioral Analysis

A mathetical lesson is more than a textbook in program form. The matheticist does not accept the standard teaching sequence as conclusive. What appears to be logical order to the professor may be chaotically confusing to the student. Accordingly, the matheticist starts fresh by going back to the fountainhead, the man who actually does the job. From this subject-matter specialist the matheticist learns exactly how the job is done under field conditions. This information is then broken down in a series of analyses designed to find the true "shape" of the behavior, as well as any teaching problems inherent in the topic, and alternate, potentially more efficient methods of performing the job.

In the course of this analysis, many hitherto unrealized facts are turned up about the job and about the system of which it is a part. This "spin-off" may bring about increased efficiency in job performance as well as savings in training costs.

When the analysis is complete, the matheticist knows exactly what material will be contained in each teaching step. In addition, he will have located most of the learning problems due to competition, or to the multiplicity of variable factors, or to terminology with which the student would not be familiar.

On the average, a full one-third of the matheticist's time is spent in the analysis phase. Some of this time, however, can be regained by increased efficiency in later phases.

B. The "Matheticist Gap"

The cost of training a staff of mathetical programmers is high, for it takes usually from six to eight months for a novice to become competent. During this time, little if any, of the work he turns out is usable. Moreover, the attrition rate among matheticist trainees is high. For some, the mathetical procedure remains forever a mystery. Others cannot develop the knack of expressing their knowledge in terms that can be understood by the student.

It is not yet possible to predict the individuals who will succeed as mathetical programmers. Even those whose backgrounds and test scores parallel those of the best matheticists may turn out to be disappointments. Conversely, we at Rehabilitation Research Foundation discovered our senior staff artist, a man in his fifties with no programming experience, to have an amazing aptitude for mathetical writing. He took over a complicated course in the bricklaying field on which the regular programmers had about given up, tinkered with it in his spare time, and turned out a lesson that we now cite as one of our best.

C. Increased Fabrication Costs

A mathetical lesson that involves the use of a simulator kit will almost certainly be somewhat more costly to manufacture than will be a simple printed lesson. Even if there is no such simulator used, production costs for the mathetical lesson will be greater than for other types of programs because of the heavier use of illustrations. It may be necessary to have a staff artist for every typist/paste-up man. For some lessons, a more expensive printing process and more expensive paper may be required because of special art work.

These, then, are the three major disadvantages of mathetics. All come down in the end to a matter of increased costs. Whether the cost is worth it or not is an individual matter, depending upon the particular circumstances of the user. In industry, the extra costs may be more than offset by the savings in trainee salaries resulting from the greater efficiency of the mathetical lesson. The fabrication costs, amortized over a sufficiently large trainee population, may, in the long run, prove insignificant. Even the problems of finding trained matheticists may be overcome if the work is let out to be done by a programming consultant on a contract basis.

CONCLUSION

Mathetics has now developed into a third force in American industrial programming. The flexibility of the "second generation" programs published within the last year or two indicate that mathetical programming is ideally suited to the training needs of vocational schools and industry where transfer of skills to the actual job situation is critical. Although mathetical lessons are inherently more expensive to produce than are those of other programming techniques, savings resulting from "spin-off" effects and from increased training efficiency may offset the extra cost.

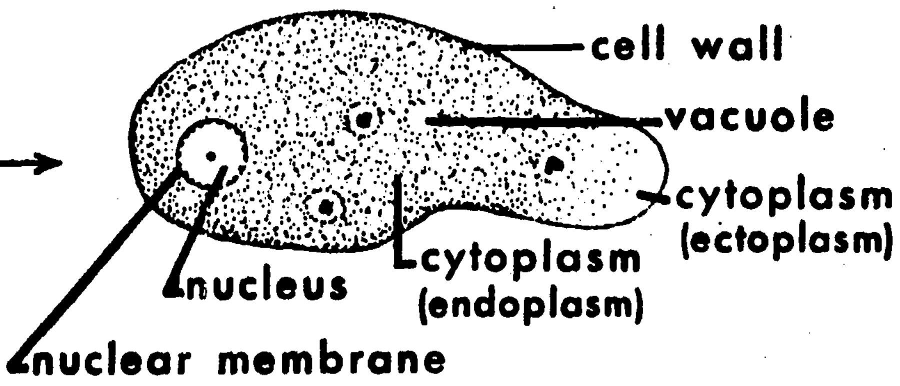
Unit 1

CHARACTERISTICS OF AMEBAE

The CELL is the basic unit of life; each cell contains all of the characteristics necessary to sustain it. These may be classified according to STRUCTURE and FUNCTION.

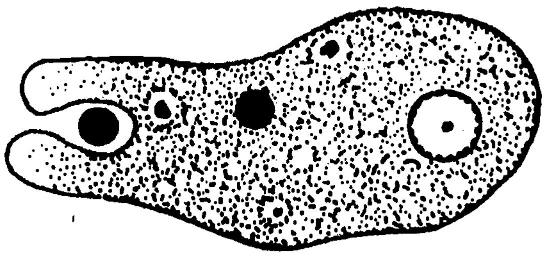
Study carefully the characteristics, drawings, and labels shown below but do not try to memorize:

STRUCTURE characteristics ⟶



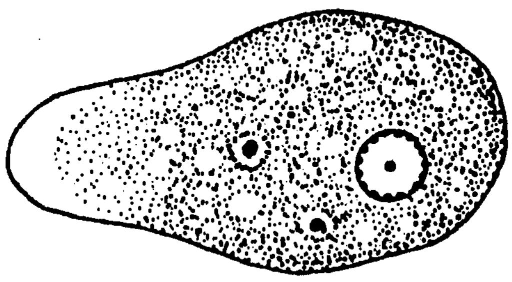
Locate the
1. nucleus
2. nuclear membrane
3. cell wall
4. vacuole
5. cytoplasm
   a. endoplasm
   b. ectoplasm

FUNCTION characteristics ⟶



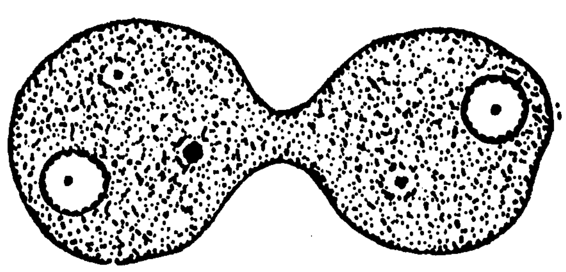
FEEDING

1. FEEDING: note how the pseudopodia engulf the food



MOVEMENT

2. MOVEMENT: note how the cell is pulled forward by the extended pseudopodia



REPRODUCTION

3. REPRODUCTION: note how the cell multiplies by binary fission

Remember:
all cells may be characterized by:
STRUCTURE & FUNCTION

(Adapted from "Amebiasis: Laboratory Diagnosis, Part I," U.S. Department of Health, Education, and Welfare/Public Health Service, Communicable Disease Center, Atlanta, Georgia.)

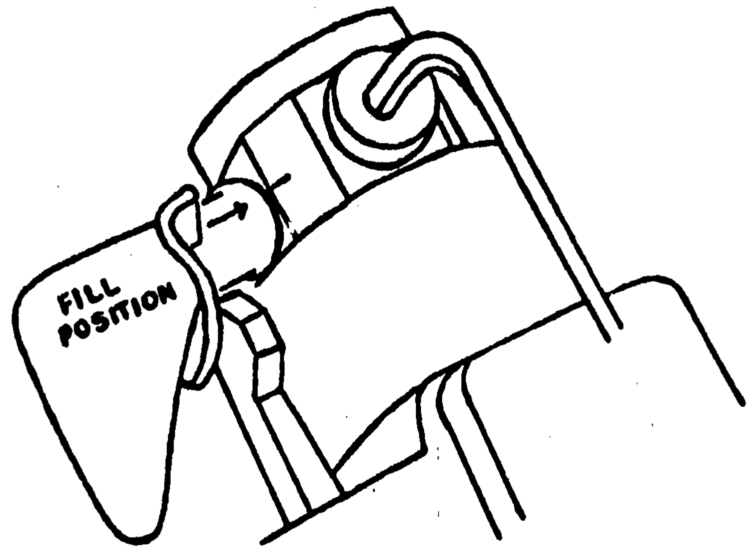
If you already knew the information contained on this page, you may skip to page 7—otherwise continue your study of the above material on the next page.

Unit 2 ADMINISTERING THE INJECTION

The schematic on these two pages shows you how to ***ADMINISTER INJECTION (Step II)***. Study the schematic carefully, then say *aloud* the steps summarized in the middle of the page until you know them. Before leaving these pages you should pick up the injector and actually follow the steps with the machine turned off. Use your own lower arm to see the nozzle imprint (be sure to take off the red protective cap first). Do it several times.

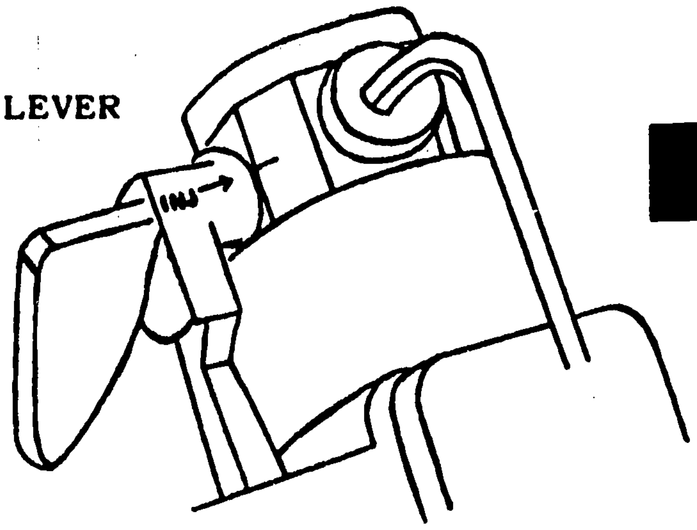
START here and follow the arrows counterclockwise:

1. see that the COCKING LEVER is at "FILL," then



1. COCKING LEVER at "FILL"
2. turn to "INJ"
3. nozzle against arm at 90° angle to the bone
4. support arm, squeeze trigger for 3 seconds
5. COCKING LEVER *immediately* back to "FILL"

2. turn COCKING LEVER to "INJ"



Transparency 2

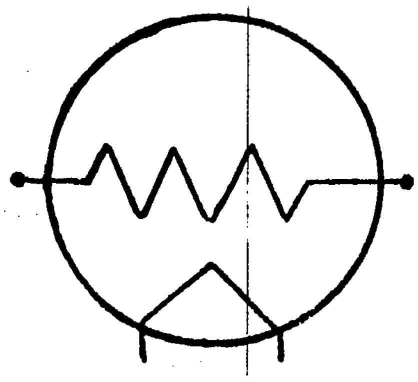
There are

TWO SYMBOLS

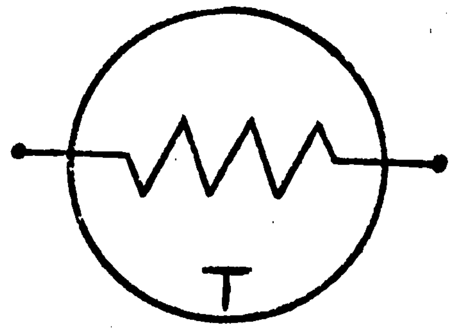
for

THERMISTORS

Notice the spelling.



and



LOOK CLOSELY AT THESE TWO SYMBOLS.

THEY BOTH MEAN THE SAME THING

AND

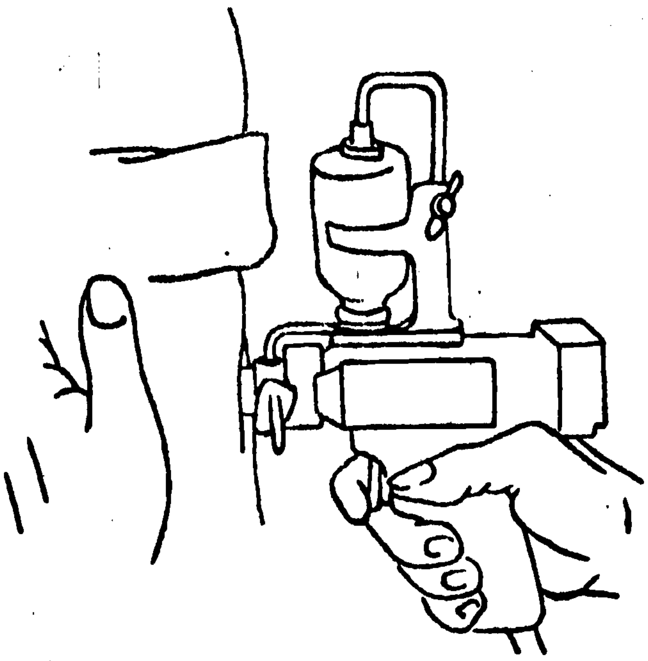
THEY LOOK JUST ALIKE

EXCEPT THAT

SOMETIMES THE SYMBOL IS DRAWN WITH ∩

AND OTHER TIMES

IT IS DRAWN WITH A " T " INSTEAD.

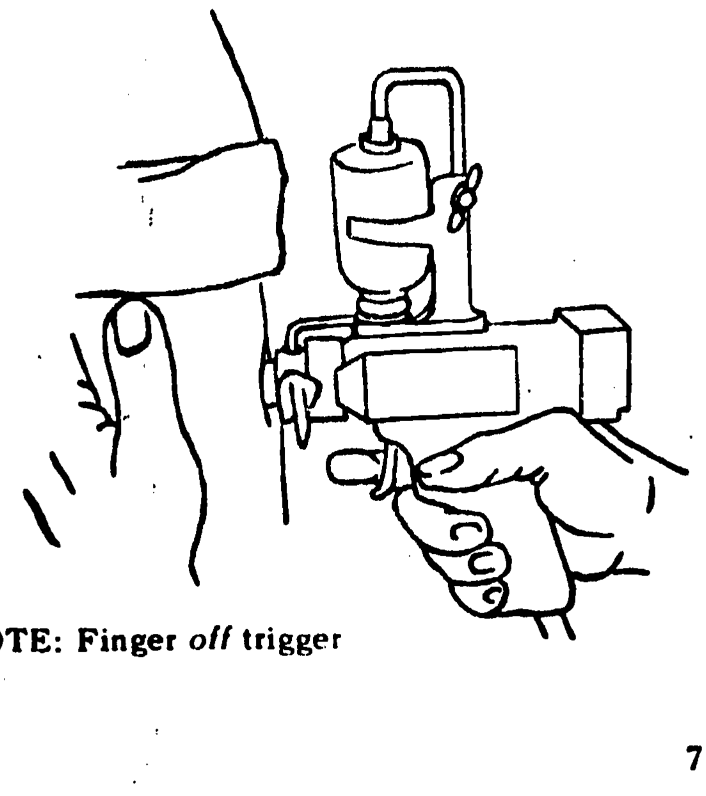


5 turn COCKING LEVER back to "FILL" *immediately* after injection. Leaving on "INJ" too long puts a strain on the machine. NEVER turn on OR off with the lever on "INJ."

NOTE: Finger *on trigger* only after the nozzle is *firmly* seated and the arm is "bunched" or stretched toward the back.

4. squeeze trigger for full THREE SECONDS (count: "One thousand one, one thousand two, one thousand three")

Remember to actually follow the steps with the injector several times before leaving these pages.



3. press nozzle FIRMLY at ½-inch-depth site at 90° angle with the bone toward back of arm; support arm ("bunch" or stretch). Seat the nozzle *firmly* (not on a muscle) so that all points of the nozzle are partially buried. Remember, correct pressure will leave a strong nozzle imprint.

NOTE: Finger *off* trigger

(Adapted from "Jet Injector Operation, Model K3," A Self-Instructional Lesson, U.S. Department of Health, Education, and Welfare/Public Health Service, Communicable Disease Center, Atlanta, Georgia.)

THE DEVELOPMENT AND PRODUCTION OF MATHETICAL PROGRAMS: A CASE HISTORY

Samuel J. Cassels, III
Rehabilitation Research Foundation
Elmore, Alabama

This paper is part of a three unit presentation on mathetics.[1]

In discussing the development and production of mathetical programs by the in-house programming unit of the Rehabilitation Research Foundation of Alabama, this paper will describe the purpose and organization of this unit, present the most useful procedures that have been developed thus far, present some variations of procedures, and make recommendations.

The Rehabilitation Research Foundation of Alabama is a private, non-profit organization that is presently conducting a number of experimental projects in education and human development at the Draper Correctional Center northeast of Montgomery, Alabama.[2] As one of these projects, a vocational school for youthful offenders is producing entrance level workers in the following six trade areas: automobile servicing, barbering, brick-laying, radio and television repair, small electrical appliance repair, and welding. In addition, the following three classes are conducted as an important part of the vocational school project: remedial education, supplementary education, and technical writing.

---

[1]For an explanation of the mathetical analysis and exercise technique, see J. H. Harless. "The Two Meanings of Mathetics," Rehabilitation Research Foundation.

For an explanation of the characteristics of mathetical lessons, and a survey of their usage in the United States, see Michael T. McGaulley. "Mathetics in Industrial and Vocational Training," Rehabilitation Research Foundation.

[2]These experimental projects are supported under the Manpower Development and Training Act, contracts #(M)6068-000 (OMAT) and #82-01-07 (HEW), and by the National Institute of Mental Health, contract #MH00976-04.

The in-house programming unit is called the Materials Development Unit, and it is an experimental and demonstration part of the vocational training project. This unit was established in October of 1964 to develop special programmed materials that would expedite the teaching of the afore-noted six vocational trades. In addition, the unit was to develop similar materials for the courses in remedial education, supplementary education, and technical writing. The unit was to investigate subjects in the programming field; for example, the evaluation and implementation of programs, and research into the methodology of programmed instruction. Finally the unit was to develop special training materials such as instructional wall charts, sequential diagrams, and vocational type visual aids.

The unit presently consists of a chief programmer, two programmers, one program editor, two production assistants, and two artists.

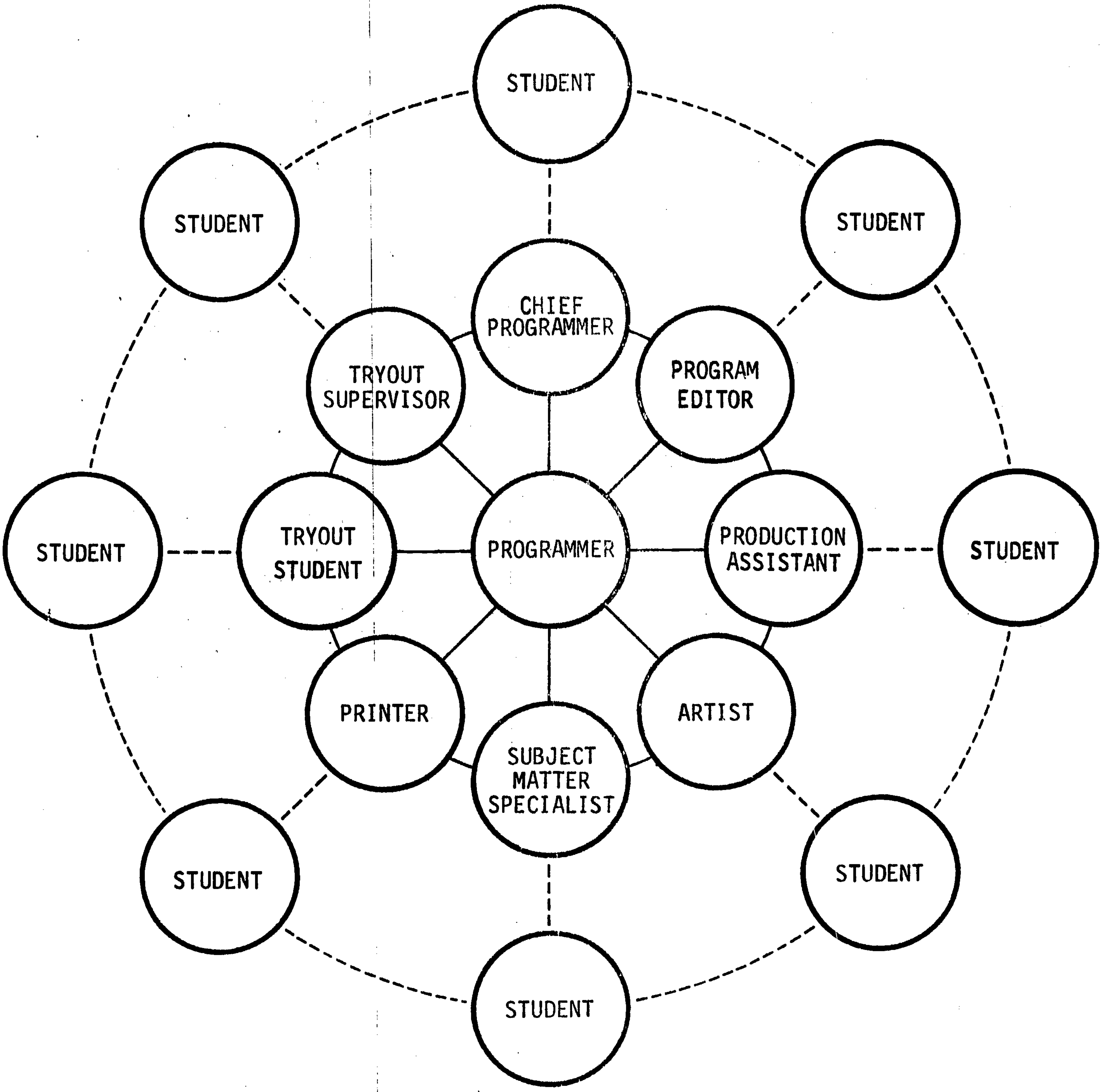
The members of the unit currently perform multiple duties. The chief programmer serves as the instructor of the technical writing class. The programmers serve as revisions associates and tryout supervisors. One programmer supervises the production section. The two production assistants perform multiple tasks including the preparation of all offset-lithographic masters for the unit and for the project. The chief artist serves as the photographic laboratory technician, and the assistant artist serves as the printer for the unit.

The collating and binding of lessons are performed with the aid of the technical writing students. They also assist the unit in the assembly of group test and field test packages.

The programming unit employs a team effort in the development and production of every product. Some of the results of the early individual tryouts made the necessity of a team effort crystal clear. The assumption of multiple duties by each member resulted from a unanimous desire for higher quality and increased efficiency.

The founding theory of the products of the Materials Development Unit is that they must be student oriented and student proved. With this theory as a base, the work of the unit can be best illustrated by the following operations chart.

(Show slide of operations chart)



MATERIALS DEVELOPMENT UNIT

OPERATIONS CHART

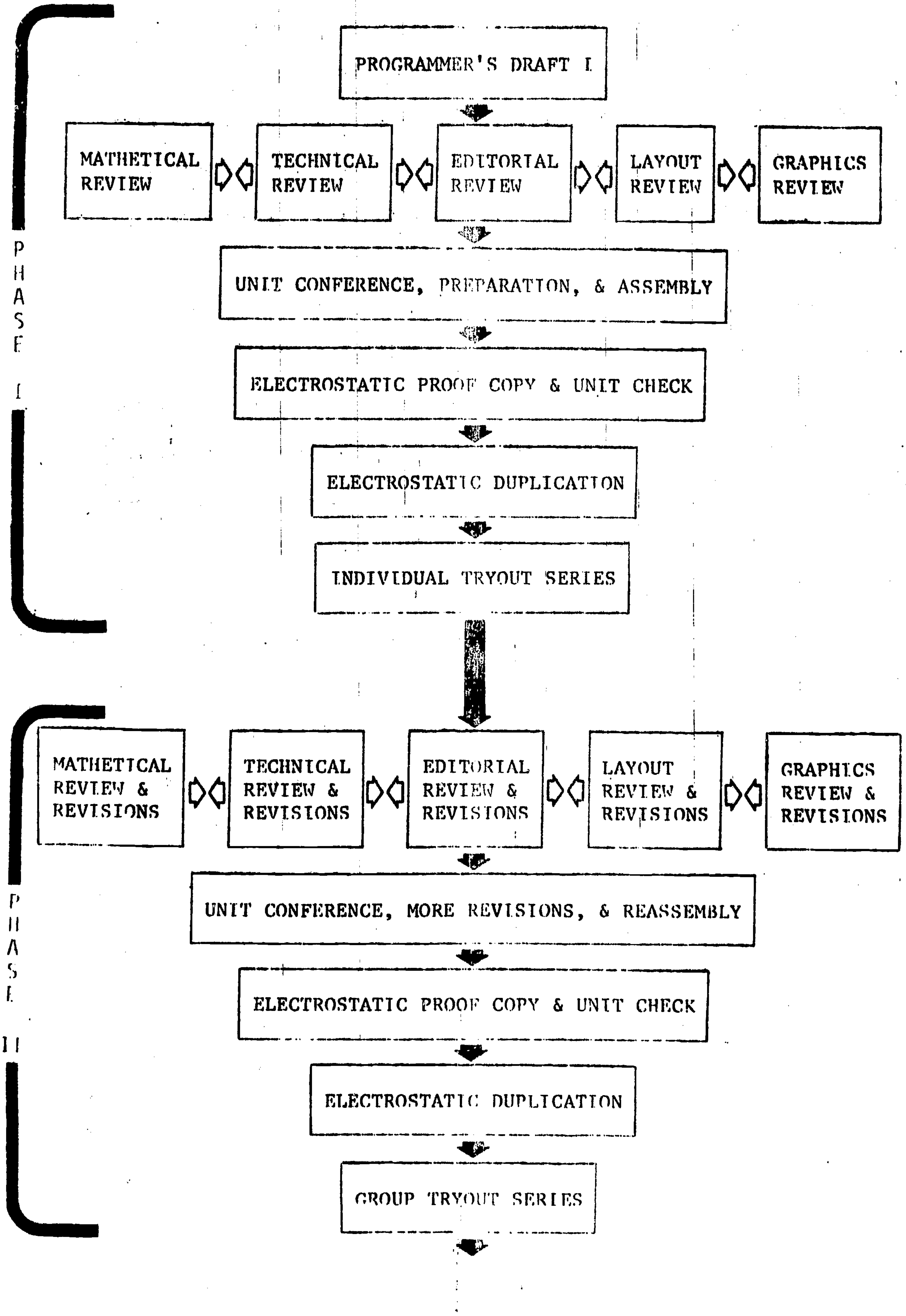
The circular form of the chart illustrates the continuous interaction of the unit members in the creation of an instructive product. The programmer is the center and cornerstone of all programming activities. Since a mathetical lesson is a tutorial medium, the programmer must communicate with the student in every aspect of the lesson. So must the other members of the programming unit.

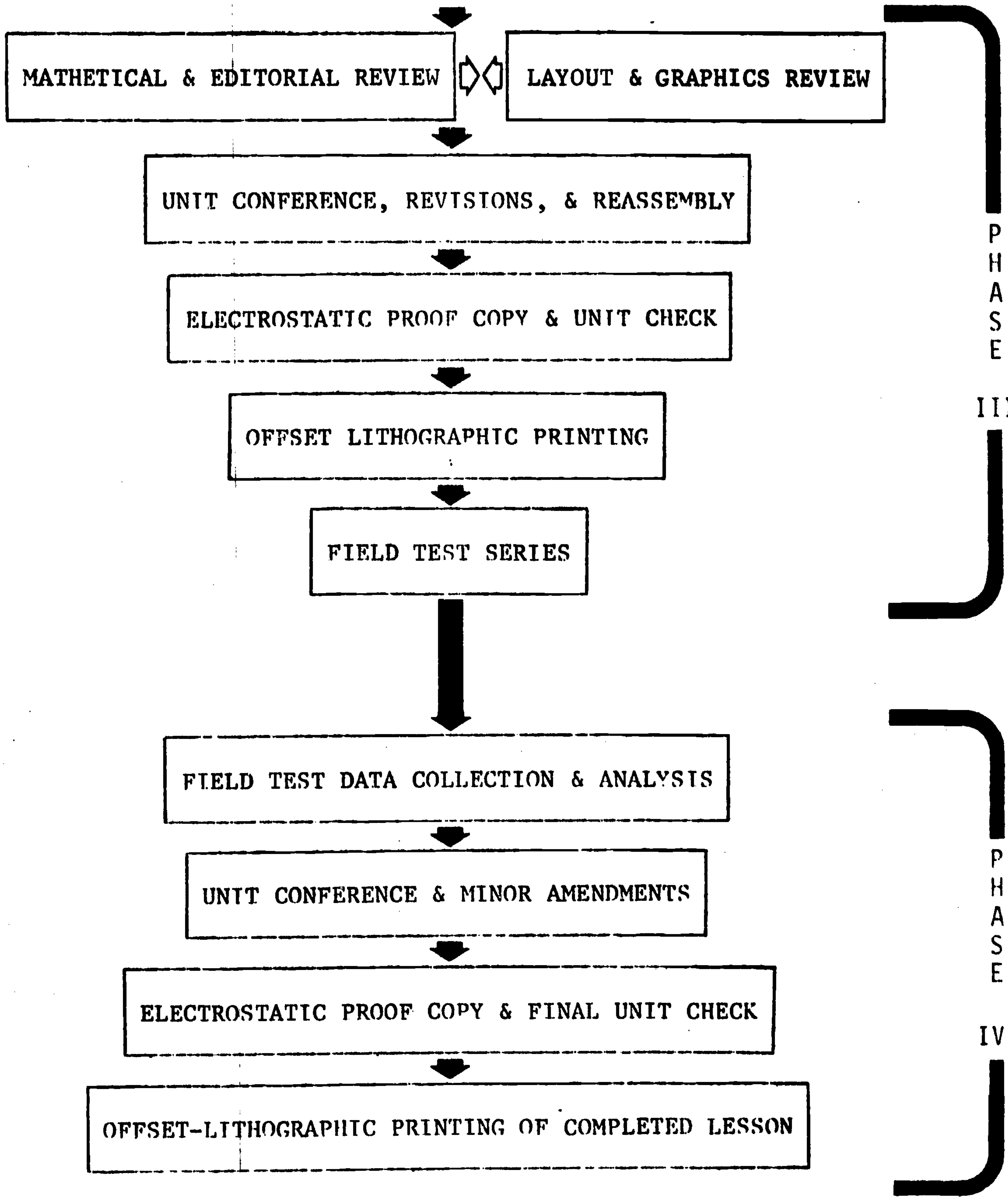
The observer will likely note that the subject matter specialist and the tryout student are included in the unit operations. In most cases, our subject matter specialists are vocational project instructors. They are valuable temporary members of our team because they are skilled in instructive techniques, and are master practitioners of their trades. The tryout students provide important feedback about the unique vocabulary and behavior aspects of the student population.

The unit has evolved a fast route of flow for mathetical lesson development. However, the unit has intentionally avoided the formation of rigid rules in order to maintain a maximum degree of originality. Our development and production process appears to best meet the needs of our type of programming unit. The following general flow chart illustrates our process.

(Show slide of general flow chart)

MATERIALS DEVELOPMENT UNIT
LESSON DEVELOPMENT & PRODUCTION
GENERAL FLOW CHART




MATHETICAL & EDITORIAL REVIEW
LAYOUT & GRAPHICS REVIEW
UNIT CONFERENCE, REVISIONS, & REASSEMBLY
ELECTROSTATIC PROOF COPY & UNIT CHECK
OFFSET LITHOGRAPHIC PRINTING
FIELD TEST SERIES
PHASE III
FIELD TEST DATA COLLECTION & ANALYSIS
UNIT CONFERENCE & MINOR AMENDMENTS
ELECTROSTATIC PROOF COPY & FINAL UNIT CHECK
OFFSET-LITHOGRAPHIC PRINTING OF COMPLETED LESSON
PHASE IV

The programmer's actions from the selection of his subject matter through the completion of his first draft have been discussed by my colleagues, Harless and McGaulley.[3] Thus, it is expedient to begin the discussion of our development and production process at the completion of the programmer's first hand-written draft. For the sake of convenience, this chart is separated into four phases.

The observer will note from the chart that the programmer's first draft is subjected to five review processes in Phase I as follows: mathetical, technical, editorial, layout, and graphics. Usually, these reviews are performed by the chief programmer, subject matter specialist, editor, production assistants, and artists, respectively. It should be noted that embryonic stages of this first draft would have undergone revision treatments by these same reviewers. After the Phase I reviews are completed, a joint conference is held during which suggestions and decisions are made concerning the composition of the first tryout lesson. Immediately following this conference, the production assistants and the artists prepare the lesson for tryout. The duplication of Phase I lessons is usually accomplished by an electrostatic copying machine.[4]

Usually, individual tryouts are conducted in sets of three, with students of low, medium, and high abilities. The review and revisions step of Phase II will determine if the lesson should be phased back for further detailed work and more individual tryouts.

---

[3] J. H. Harless, "The Two Meanings of Mathetics"
Michael T. McGaulley, "Mathetics in Industrial and Vocational Training"

[4] These subjects are discussed in detail in a report now in progress entitled "Shortcuts in the Production of Mathetical Programs," Samuel J. Cassels III, with J. A. Crosby, B. F. Harigel, D. O. Taunton, Jr., and R. R. Truitt.

The Phase II unit conference is a very important step because the first set of detailed decisions are made about revising the lesson. These revisions are carefully and promptly carried out by the staff.

Phase II duplication is usually accomplished by an in-house offset-lithographic press and by utilizing an inexpensive short run electrostatic offset master.[5]

Group tryouts of Phase II are usually conducted with four to eight students testing a lesson simultaneously under simulated classroom conditions. Our unit has tried to develop certain vocational lessons to the point where they can be successfully group tested by an entire vocational class at an opportune time in their course study. Such a tryout not only tests the lesson under excellent classroom conditions, but usually provides a later indication of the lesson's effect on the progress of the class.

At the completion of the Phase II group tryout, an important unit staff conference is held to decide what should be done to prepare the lesson for its first field test. During this conference, every facet of the lesson is discussed, including possible problems in field testing. Preliminary plans are then made, and the field test preparations are begun. When appointments for field tests are confirmed, the necessary number of copies are collated, bound, and packaged.[6]

Field tests are conducted in a manner similar to the group tryouts with the exception that a detailed orientation is given to the students prior to the field test, and that extensive population data is gathered prior to and during the period of the field test.

---

[5] See: "Shortcuts in the Production of Mathetical Programs." Samuel J. Cassels III.

[6] Ibid.

These population data are analyzed at the completion of the field test, and the results assist in the evaluation of the field test results. Ideally, no changes should be made in the field test edition of a lesson. Since few products of man are flawless, minor amendments are provided for before the completed lesson is published. Such minor amendments consist of very small changes in grammar, punctuation, and the like.

Sometimes a field test will reveal unique characteristics about a lesson. Such data should be included as a part of the lesson description. If a field test ever reveals major weaknesses, a detailed review and analysis is undertaken immediately.

The printing process for completed lessons consists of high quality printing work. Therefore, the preparation of all copy for this edition should be as exact as the production assistants and artists can make it.

It should be noted that these general procedures will vary according to the problems raised by each lesson. For this and many other reasons, each lesson should be treated as an individual case by the programming unit. We use the term INDIVIDUALIZED LESSON on every cover to emphasize that each lesson is unique in its design population, in its contents and specifications, and in the way it was created.

The most common procedural variations occur when the first individual tryout results indicate that lessons either communicate very well or virtually fail to communicate at all. In the former case, a preliminary group tryout is usually held immediately to test the reliability of the good news. If the preliminary group tryout is equally successful, indicated revisions are quickly made, and the lesson is retested as soon as possible. Further confirmation accelerates the lesson into a field test. Regretfully, our unit has experienced only a few successfully accelerated lessons to date. Due to the knowledge we gained in 1965, we anticipate achieving substantially more success with accelerated lessons in the near future.

If the first individual tryout fails to communicate with the student, the lesson is phased back to the programmer for reanalysis and rewriting. This ultimately saves valuable time and money. Such a failure cannot be charged to any lack of ability of the programmer, but rather to an overall underestimation of the learning problem for a certain population.

Other common procedural variations concern warranted shortcuts to determine good exercise and lesson design, time saving reassembly shortcuts to obtain immediate results from individual tryout revisions, extended use of quick person-to-person conferences instead of joint conferences, individual decisions instead of group decisions, and accelerating techniques relating to the physical production of lessons.[7]

At the present time, we use the following general guidelines to determine what accelerating measures and shortcuts to take in a lesson's development and production: the initial teaching ability of the first individual tryout, the class status of the tryout population, and the course demand for the lesson. We also rely on strong, but unscientific intuition.

If adequate facilites exist, tailored procedures and accelerating techniques can enable enable mathetical programs to be created for and installed in a variety of vocational training situations in a relatively short time. No attempt is made herein to state a method of

---

[7] The details of these procedures are the subject of a report now in progress entitled "Shortcuts in the Production of Mathetical Programs," Samuel J. Cassels III, with B. F. Harigel, D. O. Taunton, Jr., R. R. Truitt, and J. A. Crosby, Rehabilitation Research Foundation

predicting how much time is consumed in any phase of a mathetical lesson's development because an accurate method does not presently exist to the best of our knowledge.[8] However, a subject of simple sequential behavior of approximately five major steps or less and limited in scope can usually be programmed into an effective mathetical lesson by a unit like ours for a 9th grade population in about 15 working days. This estimate does not consider time for field testing the lesson, perfecting the developed lesson for a specific population, or treating special teaching problems. These factors, together with the other work load of the unit, constitute the major unknown quantities in time prediction.

Our experience in estimating our time needs has been heavily influenced by the inclusion of the technical writing class in certain facets of our operation. Although these young men have contributed much to the effort of our unit, instructional time combined with natural student errors have substantially subtracted from the initial estimate of the class's value to the unit.

As a result of being located within the prison compound, the unit has operated within a variety of negative physical conditions that would not likely exist in any business, industrial, or other "free world" situation.

Our unit has experimented with a wide variety of subject matter and with some of the problems of teaching an inmate population. These efforts have consumed a considerable amount of time that would have been ordinarily devoted to production efforts.

---

[8]A generalized time study, job estimation, and cost analysis is planned by the author for the near future.

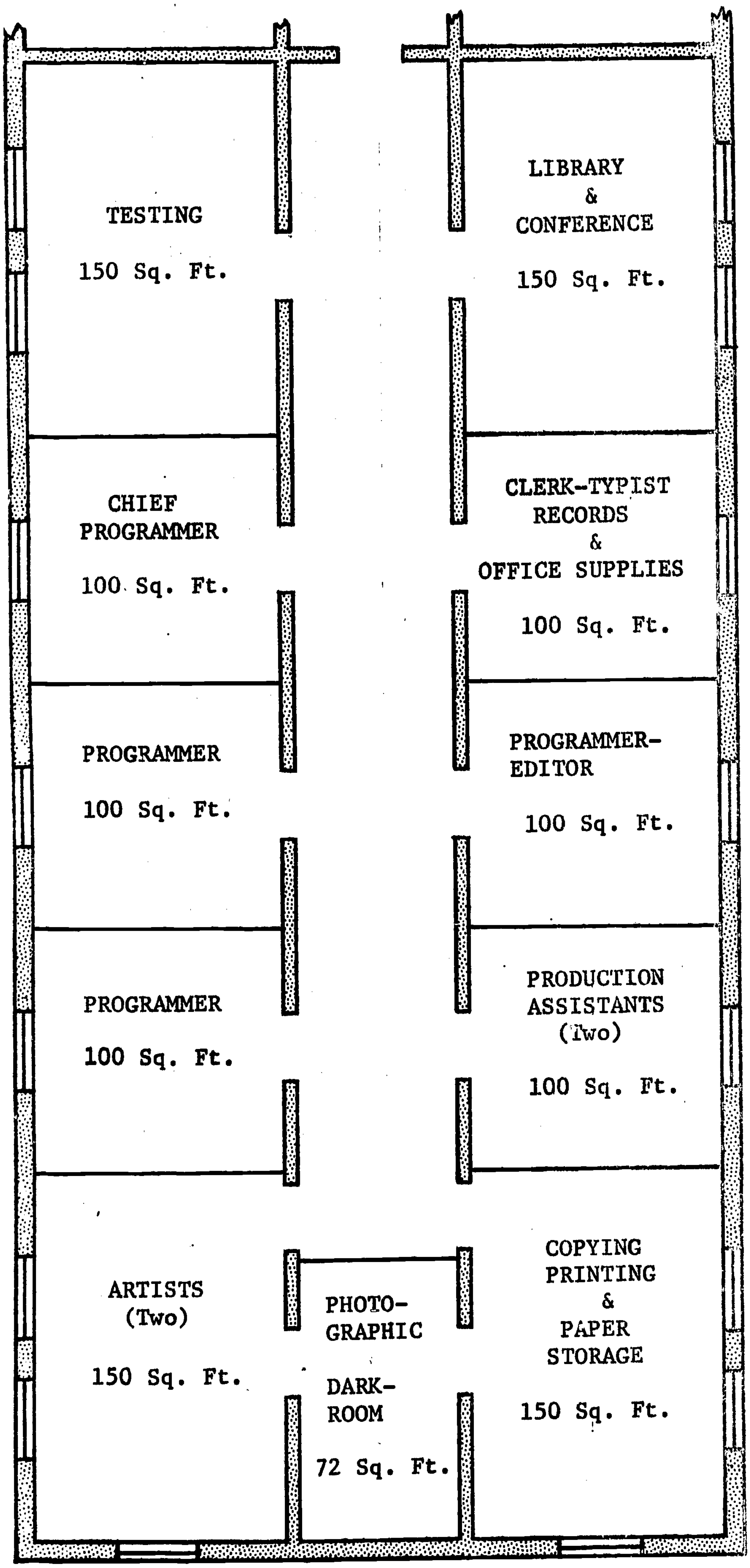
Therefore, we do not feel that an analysis of our time factors would be of substantial value to other units at the present time. We do feel that a development and production period of several weeks is a relatively short time in the field of programming. This does not mean that any mathetical program can be developed and produced in several weeks. It does mean that many subjects can be programmed into a mathetical lesson within a short period.

Mathetical programming units can usually function equally well inside or outside a training body.[9] Since a mathetical lesson should approximate the actual job environment as much as possible, our programming unit is particularly fortunate to be located in the very midst of the six vocational environments which embody most of the subject matter for our current activities. In general, any mathetical programming facility should be able to function efficiently as a separate body as long as such facility does not suffer geographical isolation from the student population, subject matter specialists, and sources of the subject behavior. Our experience has shown that a mathetical program that teaches a vocational subject can be developed by an in-house unit in a relatively short time, with a few staff members, and with a modicum of office space, equipment, and other overhead expenses. Our staff presently occupies some 691 square feet of office space for its entire operation. We have been cramped for space and our efficiency has suffered for this and for similar reasons. Therefore, we recommend that programming units like ours occupy office areas similar to that illustrated by the following diagram.

(Show slide of diagram)

---

[9] For a description of these units in the United States, see Michael T. McGaulley, "Mathetics in Industrial and Vocational Training."



SAMPLE OFFICE SUITE FOR IN-HOUSE PROGRAMMING UNIT

This suite contains 1,272 square feet of usable office space. The hall area is not included.

Note the following characteristics:

1) Can be located in wing of office building;
2) Secluded darkroom formed by closing end of hall;
3) Suite has door for extra privacy.

The size and layout of this suite is presented as a general example of a desirable office area for our type of programming unit. An even more desirable arrangement would include slightly larger offices with connecting doors.

We highly recommend that a programming office suite (and each of its offices) be both private and quiet, and that an interoffice telephone or other communication device be installed in each room. We also recommend that such noise limiting appointments as are financially feasible be installed throughout the suite.

Since programming activities require a maximum degree of concentration by both staff and tryout student, an in-house office suite should be as secluded as possible.

Our process of developing a mathetical lesson or other programmed material has been and is presently geared to vocational subjects, and particularly to those that involve a sequential overt behavior. However, our development process can be readily applied to all subject matter because no step in research, analysis, design, testing, or production has been eliminated.

Because of the accelerating capabilities of our process, and the relative brevity of the mathetical program, lessons that are needed immediately can be developed rapidly and successfully if sufficient staff time and facilities are made available.

The cornerstone of our development process is the ability of an individualized student oriented lesson to be rapidly and accurately student proved. For this cornerstone to exist, salient results of tryouts and field tests must be accurately and readily obtained. Many avenues are open to a mathetical programming unit in preparing tryout editions of an individualized lesson. Because our programs are genuinely individualized and because our unit utilizes accelerating techniques, we elected to produce our first tryout editions in a form that embodies the programmer's concept of the finished product. This is in keeping with our approach to individual tryouts--the individual tryout should serve to confirm the lesson plan, estimates of operant span and exercise design, and all other important aspects of the program. In other words, the research, analysis, and design activities of the programmer should effectively preclude major failures of the first tryout. If a mathetical programmer cannot rightfully expect initial success from his efforts, then his first tryout would be no more than a shot in the dark. This would doom mathetical programming activities to failure at the very outset.

We have discovered that it is not necessary to produce individual tryout editions in a polished physical form. To the contrary, we have determined that even an almost crude edition will prove or disprove the programmer's basic lesson design. Therefore, to confirm the most important aspects of a lesson at the earliest possible time, we have adopted a policy of producing the individual tryout editions without any time consuming finishing touches.

Group tryout editions include many physical improvements, while field test editions incorporate all planned refinements.[10]

We have been able to overcome our natural strong feelings about our first tryout editions and subsequent editions by constantly keeping this fact in mind--the student is always right! The truth of this statement has been proven again and again. And the knowledge of this fact

[10]The details of these production procedures are discussed in a report now in progress entitled, "Shortcuts in the Production of Mathetical Programs," Samuel J. Cassels III.

has enabled every member of our unit to maintain a high degree of objectivity.

Since the programming unit produces each tryout edition according to a detailed set of objectives that we define in behavioral terms, any deviations from the desired behavior will be immediately apparent upon the completion of the lesson by the tryout student. Thus, by taking carefully selected shortcuts in production work and by trying the lesson out as a complete item with definite results under close observation, our lessons can be rapidly and accurately student proved.

With close attention to details, our development process often produces some unusual tryout results. The first set of individual tryouts will often reveal important facts about the student population previously unknown. For example, the previously tested arithmetic abilities of a certain student population showed that the population should be able to negotiate a simple equation with little or no difficulty. A short programmed lesson was developed that utilized a simple equation, after the equation itself was tested for clarity by several students. But when the lesson itself was tried out by a larger number of students, we discovered that the lesson failed to teach certain students who had been "dropouts" from school or who had otherwise failed to gain an adequate education. Most of these students lacked a conventional practice in arithmetic. This proved once again that the ability of these students to perform a prerequisite behavior quickly and accurately was as important as their basic knowledge of the prerequisite behavior. But the unusual thing revealed by this tryout was the appeal of the lesson design to these students, even though these students could not adequately negotiate the behavior. Laboriously and tenaciously, students worked through the lesson with a determination seldom witnessed. Many of the interest stimulation factors in this lesson were due to the efforts of our chief artist who skillfully illustrated the simple equation with three dimensional drawings. Thus, we gathered valuable data about this art work that might have been suppressed by another development process.

The mathetical artist strives to elicit the definite, productive learning response from the student that has been carefully planned for by the programmer. The well drawn mathetical illustration will stimulate the student to accurately imitate that portion of the mastery behavior that is being graphically presented. Even though the aforementioned lesson failed to teach certain students, it was an overwhelming success with students who could negotiate a simple equation.

In addition to unusual tryout results, many types of valuable data are automatically accumulated by the development process. Since each change is carefully considered and agreed upon by the programming staff as a whole, isolated errors in lesson design are most unusual. When errors are made, they are usually pointed out by a prominent deviation in student behavior.

Individual tryouts primarily serve to correct major errors. Group tryouts primarily serve to correct minor errors. Field tests primarily serve to prove the validity of a lesson for a large population and to reveal any special specifications of a lesson. The automatic accumulation of data is pursued through every phase.

In the case where a subject being programmed is of an introductory nature or has few, if any, prerequisites, the data collected by our process can sometimes enable a lesson to be simultaneously produced for slightly different populations. For example, a simple subject matter that is generally applicable to two or more vocational courses can usually be separately tailored into a lesson for each course by utilizing items peculiar to each vocation. Such a procedure allows a programming unit to save valuable development and production time. Of course, if the subject populations have any major differences, it would be necessary to develop individualized lessons for each population.

Our streamlined production process enables us to incorporate revisions without undue loss of time. A rigorous mathetical editing procedure, together with a detailed grammatical editing procedure, provides for scientific and efficient revisions after each tryout until the final field test has been reviewed and approved.[11] The development editions can be validly produced and tested if the physical definition of the printed product is clear and no distracting factors exist such as faded print, ghost images, and lack of opacity. The fine details of the printers art should be saved for the finished product.

The mathetical lessons and other training materials that are being created by our unit are so designed that they can be readily integrated into the existing curricula of our vocational project or those of similar vocational schools. Our development and production process enables us to rapidly tailor programmed lessons for a vocational or related curricula.

As one examines any phase in the creation of a mathetical lesson, the major point to remember is that the procedures followed for a particular lesson are those procedures, however unique, that will best lead to achieving the teaching objectives of the lesson.

---

[11] These procedures will be discussed in detail in a future paper.

We do not claim that our procedures are the ultimate in efficiency. To the contrary, we believe that we have developed an efficient beginning that we can continue to improve in the future.

We welcome the challenge of our future. We have examined our experiences of 1965, and have gained a new determination to solve the problems we have heretofore failed to solve, and to meet more difficult challenges than we have previously met. We are confident that our future efforts can be more effective and that our services can be increased.

In closing, the Materials Development Unit of the Rehabilitation Research Foundation of Alabama extends an invitation to inquiries about details of our activities.

"Mathetics: The Ugly Duckling Learns To Fly"

by J. H. Harless, Chief
Materials Development
Rehabilitation Research Foundation

This paper is intended to be a partial answer to numerous queries concerning mathetics[1] received by the author. These questions can be summarized:

"Whatever happened to mathetics?"

"What is mathetics?"

"What did Gilbert mean when he said ...?"

"What is the difference between mathetics and normal (sic) kinds of programming?"

As is consistent with good practice in mathetics <u>and</u> programmed instruction, an attempt was made to diagnose the exact "training deficiencies" of the "student" population before this paper was written. Over 200 questionnaires were sent to last year's NSPI Convention attendees. Questions pertinent to this report were:

"Have you ever seen a mathetical program?"

"If so, what were the titles?"

---

[1] For additional information on the mathetical technology, see the following papers available from the Rehabilitation Research Foundation, Elmore, Alabama. J. H. Harless, "The Two Meanings of Mathetics"
Michael T. McGaulley, "Mathetics In Industrial And Vocational Training"
Samuel J. Cassels III, "The Development And Production Of Mathetical Programs: A Case Study"

"What are the major characteristics you have observed or heard about the mathetical system?"

"Are there any specific things you would like to know about the mathetical system?"

One hundred and four questionnaires of the 200 were **returned.**

"Have you ever seen a mathetical program?"

Yes:	51
No:	49
Don't know:	4

"If so, what were the titles?"

No answer:	5
Samples:	18
One:	22
Two or more:	20

"What were the major characteristics you have observed or heard about the mathetical system?"

Even though less than half admitted ever having seen a mathetical lesson, all but ten responded to this question.

The most frequent, and usually the <u>only</u>, comment was, "Mathetics teaches backwards."

"Are there any specific things you would like to know about the mathetical system?"

The most frequent responses were:

"A simple explanation of what it is."
"What are the differences in mathetics and other approaches?"
"A guide to writing mathetical frames."

Several things are obvious, if these one hundred and four respondents are representative: A. Very little is known about the mathetical system. B. What is known is a misconception: mathetics is a different format for presenting frames; that is, arranging them in backward order. C. Format alone characterizes programmed instruction of any kind.

What Mathetics Is

Just as it is ridiculous to characterize linear programming as a process of breaking subject matter down into small frames of information, it is even more erroneous to describe the mathetical process as "presenting frames backwards."

The ultimate format of a mathetical lesson is unknown at the beginning of the mathetical process. It is our contention that there are too many variables in the nature of the behaviors being taught, the characteristics of the intended population, and the environmental and curricular setting to be able to state what a mathetical training vehicle will "look like" before a detailed and systematic investigation is undertaken.

This systematic investigation of a training task is the broadest definition of mathetics. The details and implications of this definition are considerable.

Mathetics is a complete training system. It is a step-by-step guide for the lesson writer's behavior to insure that he has considered each element of the training task, has examined and noted numerous facets of the learning theory as related to his particular training task. In short, mathetics is a systematic and documented procedure for looking at behavior to determine the most efficient and effective method for changing behavior.

This "scientific eclecticism" has resulted in a rather rigorous procedure in recent years. Procedures described by Gilbert[2] are still used in part, but subsequent research and trial have dictated revisions, additions, and subtractions. The most glaring example of these changes is the de-emphasis on the chaining of behaviors, probably the most controversial and best known teaching strategy recommended in Gilbert's original treatise.

Although a comprehensive written document still does not exist, the following is the general procedure employed by the mathetical unit of the Rehabilitation Research Foundation.[3]

<u>Occupation Analysis</u>[4]: Given the delimited domain of an occupational title or subject matter area and a general design population, the matheticist, with the aid of a subject matter specialist and references, lists the tasks that make up that domain. In this initial step, the matheticist is interested only in the overt behaviors or the physical products of behavior. (See Appendix A.)

<u>Task Selection</u>: Each task in the list is examined in two phases in the form of a series of selection criteria questions.

---

[2] Gilbert, Thomas E., "Mathetics: The Technology of Education," <u>Journal of Mathetics</u>, Vol. 1, No. 1, January, 1962.

[3] This procedure is <u>similar</u> to the system employed by the Instructive Communications Unit of the Communicable Disease Center, another major producer of mathetically oriented materials.

[4] This procedure was designed for the examination and reconstruction of repertories in the industrial and vocational areas, but is applicable to virtually any domain.

Phase I: A. Can a majority of these students presently perform this task to a minimal level without training?

B. Do adequate training materials exist that "teach" this task as a unit.

If the answer to either of these questions is "yes," the task is eliminated from further consideration.

Phase II: The remaining tasks are listed and the following questions are weighted and asked of each task:

A. Is the instructor unable to teach this task to an acceptable level with only one group demonstration?

B. Is this a genuine training problem?[5]

C. Are there many stimulus generalizations?

D. Can the behaviors be simulated economically?

E. Is the method for performing this task relatively constant?

F. Will training materials on this task have wide use.

G. Is this a basic learning problem?[6]

H. Is there relatively common agreement on the method of performance of this task?

---

[5] That is, does the student need to know, recall, and perform this task; or would a checklist, written instructions, etc. fill the need; or is it a motivational problem?

[6] Or is it contingent on many subskills?

I. Can training material on this task be economically evaluated?

The perfect candidate for a mathetical lesson would have all "yes" answers to these questions. However, the tasks are then listed in priority for treatment.

Task Analysis: Although the task analysis is much too involved a procedure for discussion here, generally the matheticist breaks down the highest priority task into the "products" of the behavior of the task and describes it according to:

A. The criteria of acceptable performance (in terms of time, completion, and accuracy).

B. The small steps of performance of the task.

C. The related information that will facilitate the performance and generalization of the task.

D. Special difficulties experienced by the subject matter specialist in teaching these behaviors to the design population. (See Appendix B.)

Population Analysis: The matheticist discovers all he can about the design population, reading level, math skills (if applicable), prior experience, age range, general intelligence and cultural background.

Training Deficiency Analysis: The task analysis is compared to the population analysis to further define what is to be taught. This analysis should answer the following questions:

A. "Does this task require many sub-lessons (successive approximations to be presented as sub-lessons in a package)?

B. "Will additional diagnostic tests be required to determine the precise deficiences of individual students?"

C. "Will it be possible to prepare one lesson for the task with different 'tracks' for further treatment of individual differences?"

Efficiency Analysis: The order, the possibility of hidden discriminations, the extent of generalization, and omissions of any steps as given by the subject matter specialist are examined. The matheticist does this by performing the behavior himself, noting his own behavior, experimenting with the order, reformulating, testing etc. The matheticist examines references to determine if there is any additional information not gained from the subject matter specialist.

Training Objectives And Criteria Exam: The matheticist uses an approach similar to the well-known Mager system[7], but with special emphasis on describing the restrictions and limitations of the lesson. At this point the matheticist and the subject matter specialist(s) translate the objectives into a mode of evaluation.

Prescription of Behaviors: The behaviors listed in the steps of performance and the pertinent covert behaviors are written

---

[7] Mager, Robert F., Preparing Objectives For Programmed Instruction, 1962, San Francisco: Fearon Publishers, Inc.

in stimulus-response terms. This "behavioral blueprint" expresses the chain and subchains of the behavior and all the discriminations the student must make. The prescription precisely identifies the discriminative stimuli and the responses they occasion. (See Appendix C.)

Generalization Analysis[8]: Each discriminative stimulus is examined to determine the smallest number of instances of the stimulus that should be represented in the lesson to allow for maximum generalization by the student.

Competition-Facilitation Analysis: The matheticist compares each operant of the prescription to every other to determine the interactive characteristics of the behavior. He notes intra and extra lesson competition and facilitations in an effort to determine the most effective order of presentation of the behaviors and to spot special problem areas that will require some additional teaching strategies[9]. (See Appendix D.)

Estimation of Operant Span: On the basis of all foregoing work, the matheticist reconsiders and rewrites the prescription to express the largest step toward mastery the student can take at one time. This is done by combining adjacent operants of the

[8] Detailed procedures for the performance of this and subsequent analysis have been worked out, but are too lengthy for presentation here.

[9] Chaining is an example of a strategy the matheticist may employ on rare occasions. The more common strategies include the use of mediation (a special class of mnemonic), additional prompting exercises, and the maximum use of illustrations.

prescription according to the difficulty of the behavior for the design population, the characteristics of the population, the problems discovered in the analysis of the prescription, and other factors noted by the writer or indicated by the subject matter specialist. This conception of operant spans will be discussed later in the paper.

Lesson Plan: The matheticist lists the operants with their new "spans" in the sequence he has decided to teach them. This lesson plan is written consistent with the philosophy of the "exercise model" whereby each operant is demonstrated to the student in one exercise, prompted to performance in a second, and released to perform the operant in a third without cues.

One exercise may contain a demonstration, a prompt, and a release of three different operants, contingent on the findings of the analysis. In any event, all operants are released for performance by the students in their correct sequence at least once.

Lesson Construction: Following a rather precise set of guidelines, the writer constructs each exercise called for by the lesson plan: represents the discriminative stimulus (S^D); locates the response locus and writes instruction on the performance of the response (S^I); draws attention to the discriminative stimulus (S^A); classifies the discriminative stimulus (S^C); sets the exercises in context with the rest of the behaviors being taught (stimulus complex); allows the student to perform the response in presence of the demonstration (S^F). As mentioned, the exercise may call for two additional functions: a prompt of a previously demonstrated behavior (S^P), and/or the uncued production of a previously demonstrated and prompted operant (S^L). (See Appendix E.)

The remainder of the mathetical process is very similar to the try-out and revision cycle used by conscientious programmers. Each draft of the lesson is tried out on representatives of the design group and revised until the writer is reasonably confident that the lesson achieves the training objectives.

The lesson is field tested on large numbers of students under the actual conditions of intended use.

What Is Different About Mathetics?

Formerly, there was meaning in contrasting mathetics with programmed instruction, but today the "differences" are isolated to a few concepts and practices.

Perhaps a more correct title for this paper would have been "Programmed Instruction: The Ugly Duckling Is Beginning To Fly." This would surely be the case if one were to examine the professed practices and procedures of many of the progressive investigators in programmed instruction and weld their techniques into a precise system.

Whether this progress is a result of interactive influence of mathetics and programmed instruction or independent growth is academic and germane only to ego-needs. The happy fact is that some programmers are re-examining, revamping, and progressing.

However, an examination of the literature, promotional pieces, and a majority of even the most recent programs reveals a still faithful adherence to these characteristics of programmed instruction:

1. Small steps
2. Active responding
3. Immediate confirmation
4. Self-pacing
5. Small error rate
6. Logical sequence

Some authors add one more:

7. Operationally defined objectives

This practice continues despite experimental evidence to the contrary on the validity of some of these items. This practice continues in face of few linear programs and almost all mathetical lessons which have been validated and which demonstrate large step sizes, require large amounts of covert responding[10], give little or no immediate confirmation, and are not in normal-order performance sequence.

In spite of "progressiveness" on the part of many linear programmers, mathetical lessons still exhibit some different characteristics on the face, and a vast number in the part of the iceburg

[10]Most programmers equate "active" responding with "overt" responding.

that the student doesn't see - the rigorous system of analysis employed by mathetical lesson writers.

The major observable difference is the use of the "exercise." The mathetical exercise usually encompasses a double-page spread, but may be several pages long. There is no meaning in a comparison of a frame to an exercise. An exercise represents the largest amount of behavior a student can absorb in one demonstration. One exercise may take scores of frames (in a linear program) to teach, or one frame may indicate behaviors contained in several exercises.

The second observable difference is in the method of confirmation. A mathetical lesson is usually accompanied by an answer book, but not all responses are confirmed. Usually the student is encouraged to check his answers in early exercises of the lesson, and whenever the nature of the responses warrants it. A mathetical lesson recently produced by the Rehabilitation Research Foundation had no confirmation in any form for one edition and a complete answer book for another. No remarkable difference in post-test performance was seen between the two field test groups except that the average time for the answer book group was longer.

Absent, of course, in mathetical lessons are frames of any size; gone, therefore, is the rigidity of format and the boredom of endless blocks of type and blanks so characteristic of linear programs. The matheticist makes a special effort to simulate the behaviors graphically, especially the discriminative stimuli; therefore, the mathetical lesson is usually highly illustrated.

Much time is spent in the design and layout of an exercise to make it attractive and interesting as well as to serve precise functions.

One of the most pleasing differences to the student is that mathetical lessons are characteristically small in bulk. This is due to the increased operant span and the philosophy of greatly delimited domain of the original endeavor.

We have been talking about what lessons and programs look like. The most important consideration, however, is the systematic approach the matheticist takes and the precise attention paid to each analysis. Many programmers profess to perform "behavioral analysis;" usually, a close examination reveals that even the most sophisticated are merely listing generalizations and discriminations. Unfortunately, most programmers begin to write frames as soon as objectives are written and profess to "analyze" as they go along.

The effort that it takes to become proficient in this analytical procedure is self-evident. Whether any programmer, or matheticist for that matter, is willing to expend this extra energy is a factor involved in a larger question than is germane here.

Summary

Mathetics, therefore, is a training system that provides for the trainer: (1) a guide for determining what to teach, (2) a basis for making teaching strategy decisions, (3) a detailed procedure for constructing a lesson.

No matheticist professes to have "the answer." No matheticist feels that he is unique in the universe. Mathetics is constantly undergoing change and will continue to do so until a genuine technology of education is achieved.