DOCUMENT RESUME

ED 053 935 24a SE 012 105

AUTHOR Ferguson, Richard L.; Tse-Chi, Hsu

TITLE The Application of Item Generators for Individualizing Mathematics Testing and Instruction.

INSTITUTION Pittsburgh Univ., Pa. Learning Research and Development Center.

SPONS AGENCY Office of Education (DHEW), Washington, D.C. Bureau of Research.

BUREAU NO BR-5-0253

PUB DATE May 71

NOTE 21p.

EDRS PRICE EDRS Price MF-$0.65 HC-$3.29

DESCRIPTORS *Computer Assisted Instruction, Educational Technology, Educational Testing, *Individualized Instruction, Instructional Media, *Mathematics Education, Programed Materials, *Test Construction, *Testing

ABSTRACT

Described is a procedure for utilizing a computer to generate domain-referenced tests in mathematics. The procedure can be adapted for use in testing and instructional programs in either an on-line or off-line mode. It requires specification of the objectives of interest in behavioral terms and grouping them into sets that share a common content. Addition, multiplication, and fractions are examples of possible groupings. To implement the procedure, one of the sets of objectives resulting from the grouping process is selected, and item forms representative of the behaviors implied by each objective in the set are specified. Then an item generator is developed that facilitates the construction of items representative of all item forms so identified. Given an on-line computer capability, the authors describe how it is possible to use the proposed item generator for assisting measurement and instruction in an individualized mathematics program. (Author/JG)

BR 5-0253
PA 24
SE

UNIVERSITY OF PITTSBURGH - LEARNING R&D CENTER

1971/14

THE APPLICATION OF ITEM GENERATORS FOR INDIVIDUALIZING MATHEMATICS TESTING AND INSTRUCTION

RICHARD L. FERGUSON AND TSE-CHI HSU

THE APPLICATION OF ITEM GENERATORS FOR INDIVIDUALIZING MATHEMATICS TESTING AND INSTRUCTION

Richard L. Ferguson and Tse-chi Hsu

Learning Research and Development Center
University of Pittsburgh

May, 1971

The research reported herein was supported by the National Science Foundation and by the Learning Research and Development Center supported in part as a research and development center by funds from the U.S. Office of Education, Department of Health, Education, and Welfare. The opinions expressed in this publication do not necessarily reflect the position or policy of the Office of Education and no official endorsement should be inferred.

ABSTRACT

THE APPLICATION OF ITEM GENERATORS FOR INDIVIDUALIZING MATHEMATICS TESTING AND INSTRUCTION

Richard L. Ferguson and Tse-chi Hsu

A description is provided for a procedure to utilize a computer to generate domain-referenced tests. The procedure can be adapted for use in testing and instructional programs in either an on-line or off-line mode. It requires specification of the objectives of interest in behavioral terms and grouping them into sets that share a common content. Addition, multiplication, and fractions are examples of possible groupings. To implement the procedure, one of the sets of objectives resulting from the grouping process is selected and item forms representative of the behaviors implied by each objective in the set are specified. Then an item generator is developed that facilitates the construction of items representative of all item forms so identified.

Given an on-line computer capability, the authors describe how it is possible to use the proposed item generator for assisting measurement and instruction in an individualized mathematics program. Such an endeavor is currently underway at the Learning Research and Development Center at the University of Pittsburgh as a component of a project sponsored by the National Science Foundation for providing computer assistance for education in individualized schools.

THE APPLICATION OF ITEM GENERATORS FOR INDIVIDUALIZING MATHEMATICS TESTING AND INSTRUCTION

Richard L. Ferguson and Tse-chi Hsu

Learning Research and Development Center
University of Pittsburgh

Computer Assistance in Measurement

The first application of computer technology to testing, although recommended by Smith as early as 1963, can be found with the emergence of computer-assisted instruction (CAI). CAI relies heavily upon testing, more precisely, response assessments, to provide information for branching decisions. Without these, most CAI would fail to realize deep levels of individualization.

In addition to performing a useful role in CAI, computer-assisted testing (CAT) appears to have a great potential for roles that are exclusively oriented toward measurement. The use of the computer for administration of both norm-referenced tests and domain-referenced tests is an illustration of potentially profitable applications of computer technology to the improvement of measurement procedures.

CAT has some attributes that are difficult to match with conventional paper and pencil tests; it provides quick feedback and allows flexibility in application. Further, paper and pencil tests are inefficient in testing extreme cases because they are usually designed to conform to the median ability of the group to be tested.

The branching capability of CAT makes possible the presentation of items tailored to the ability of the examinee, therefore identifying examinees with extreme abilities more efficiently (Linn, 1969).

A relatively untapped but potentially significant use of the computer for testing purposes involves the measurement of cognitive processes. A better understanding of these processes seems likely to result in new techniques for measuring cognitive functioning (Green, 1969).

The most frequent applications of the computer in measurement have been of the following type:

Computer-Administered Tests. Tests are constructed and then stored in the computer. Items are presented one by one, either at standard teletype or cathode ray tube (CRT) consoles. Although decision and branching functions may be incorporated into this model of testing, it remains similar to fixed length paper and pencil measurement in the sense that the test items are fixed. That is, repeated administrations of the instrument would yield exactly the same test. Aside from requiring a large portion of the computer's memory for storing test items, the amount of computer time expended for this type of application is difficult to justify when examining its advantages over conventional paper and pencil tests.

Computer-Assembled Tests. A large item pool for a particular content area is constructed and stored on tapes or disks. Test constructors then specify criteria required to yield a stratified

sampling from the pool. This type of application may be of particular advantage to test users who do not have sufficient time to construct their own tests. An additional advantage of such tests is that they can be constructed so as to satisfy precisely defined test criteria. As might be expected, the manner of storage and retrieval of desired items is the point of concern in this application (Forbes, 1970).

Computer-Constructed Tests. Necessary information and logics for generating test items are programmed for residence in the computer. The computer then constructs items according to specified parameters. This type of procedure has been used in sentence completion (Anastasio, et al., 1969), spelling (Fremer and Anastasio, 1969) and mathematics (Ferguson, 1970). The procedure features the use of the mechanism of concern in this paper: a routine that permits item construction according to user specification. This routine is called an item generator. One advantage of using item generators is that they do not require large amounts of computer memory; that is, access to a small amount of computer memory is likely to be sufficient for generating any item in the domain of items for which the item generator was programmed. In addition, item generators do not artificially restrict the size of the item pool from which the test constructor can sample. The latter observation reflects the fact that, on a test employing item generators, all items in the specified domain have a non-zero probability of presentation to the examinee. This is not the case for conventional paper and pencil tests or for computer-assembled or computer-administered tests which fix the particular items from the given domain that can be presented.

A good item generator for computer testing should have the following attributes:

(a) requires a minimum amount of the computer's memory,
(b) generates items quickly and efficiently,
(c) permits generation of many different forms of items using the same generator, and
(d) produces items with specifications as precise as the user requires.

Obviously, the nature of the content to be tested is a major factor in determining the characteristics of an item generator. It is much easier to build an efficient generator for mathematics than for reading or the social sciences. Regardless, it is not the intent of this paper to discuss the nature of the content for which item generators are constructed. Rather, our concern is with the development of procedures for building item generators for use in individualized education programs.

Hively, Patterson, and Page (1968) and Osburn (1969) have recommended an approach which features item form analysis for building item generators. The content is analyzed, item forms are specified, and generation rules are devised that permit the random generation of items representative of each item form. This approach is often referred to as domain-referenced testing.

If a single item generator is constructed so as to satisfy the requirements for generating items representative of only <u>one</u> item form, a test that requires measurement of a large variety of item forms may demand large amounts of computer memory to operate. Rather than using <u>an</u> item form as the basis for constructing <u>an</u> item generator, this

paper proposes to unite item forms that share a similar content into a cluster and then construct a general item generator capable of producing an item that is an element of the domain of any item form found in that cluster. The programming task is likely to be more arduous, but the approach offers greater efficiency whenever it is necessary to generate items that are not defined by a single item form. A multiplication generator will be described for the purpose of demonstrating how a generator based on clusters of item forms differs from one derived from a single item form. The discussion will include a description of how such a generator might effectively be used in a program of individualized education.

Testing in a Program of Individualized Instruction: A Frame of Reference

The Learning Research and Development Center at the University of Pittsburgh is concerned with the development of model school environments that have the capability to adapt to individual differences among students in ways that maximize educational outcomes. One element of this developmental effort is the Individually Prescribed Instruction (IPI) mathematics program of the Instructional Design and Evaluation project, the curriculum of which is defined by over 400 behavioral objectives. The objectives are grouped to form units that share a common content and difficulty level. For example, multiplication is a content area that is comprised of six units, each comprised of objectives of varying complexity.

Testing plays an important role in determining the instructional activities for individual students. In this context, measurement exists

to facilitate instructional decision making. For initial decisions, placement tests are used to determine the units of the curriculum for which the student has not achieved proficiency. Then, unit pre-tests and posttests are used to identify the skills that a student possesses within a given unit. Curriculum Embedded Tests (CETs) measure a student's proficiency in a specific skill. Each of the tests described serves to provide information that is then used to formulate an instructional plan for the student.

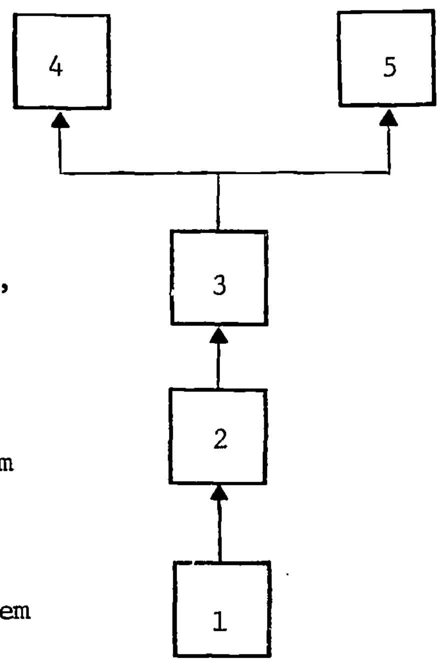
The structure inherent to the mathematics curriculum makes plausible the assumption that the skills that define the curriculum can be linked together in an order that reveals the prerequisite relationships among those objectives. At a less molecular level, it should be possible to specify the structure for specific units of IPI mathematics. Figure 1 provides a list of the behavioral objectives for the level F multiplication unit. It is accompanied by a graphic representation of a hierarchy for those objectives.

For this five skill unit, objectives 1, 2, and 3 are pre-requisite to skills 4 and 5; that is, proficiency is required in the former set before it can be attained in the latter. Also, proficiency in skill 3 implies proficiency in skills 1 and 2. Lack of proficiency in skill 2 implies the same state for skills 3, 4, and 5. Skills 4 and 5 are placed at the same level in Figure 1 to indicate that the order of instruction or testing for these two skills is arbitrary.

Figure 1

Objectives for Level F Multiplication Unit
and Their Prerequisite Relationships

1. Given a two-digit number times a two-digit number, the student multiplies using the standard algorithm.
2. Given a three-digit number times a two-digit number, the student multiplies using the standard algorithm.
3. Given a whole number and a mixed decimal to hundredths as factors, the student multiplies. LIMIT: Whole number part <100.
4. Given the product of two pure decimals ≤.99, the student shows the equivalent in fractional form and converts product to decimal notation, compares answers for check.
5. Given a multiple step word problem requiring multiplication skills mastered to this point, the student solves. (≤3 steps)



For purposes of discussion, let us assume that placement testing has ascertained that a student should begin his study with the level F multiplication unit. A pretest would be used to identify the unit skills for which proficiency is yet to be realized; CETs would be used to assess the effectiveness of subsequent instruction for each of these skills; and, posttests (equivalent forms of the pretests) would confirm the acquisition of unit skills after all instruction had been completed. A demonstration of how the item generator described earlier can be effectively used will focus upon

the construction of tests that perform the function of pretests and posttests in IPI mathematics.

The Structure of the Pretest/Posttest Model

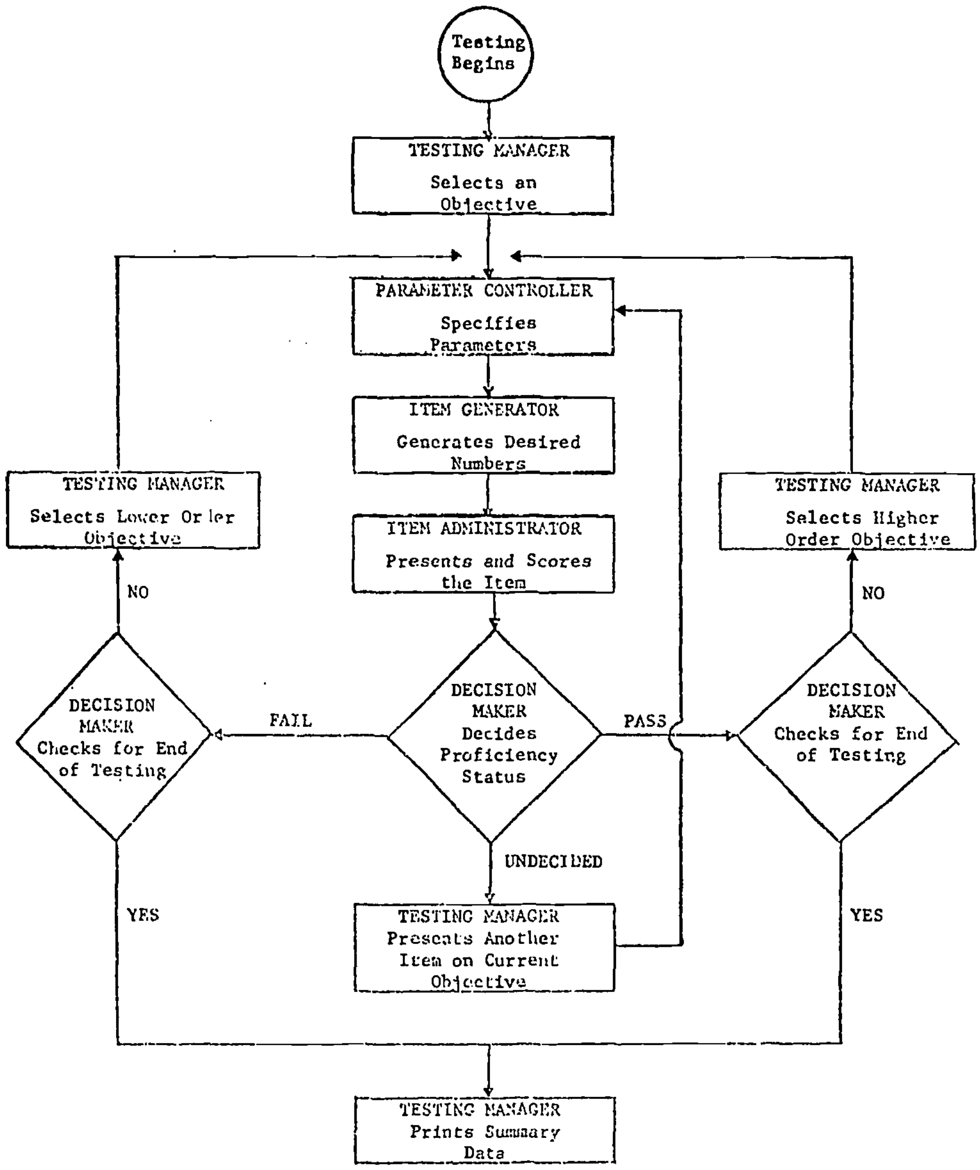
The function of a pretest or posttest is to ascertain the proficiency status of the examinee for each objective. With the existence of a hierarchy for the five skills and a branching rule adapted to it, such a profile can be obtained without testing all of the skills. A previous study has demonstrated that branched testing after this fashion can substantially reduce the time required to obtain the unit profile (Ferguson, 1970).

The test model that is used for the development of computer-assisted branched pretests and posttests is described in Figure 2. Notice that it is comprised of five components: (1) TESTING MANAGER, (2) PARAMETER CONTROLLER, (3) ITEM GENERATOR, (4) ITEM ADMINISTRATOR, and (5) DECISION MAKER. A brief description of each of the components is provided below. A detailed example using the IPI unit described earlier will follow.

TESTING MANAGER. This component controls the sequence in which objectives are tested; that is, determines which skills will be tested and in what order. The criteria used for branching include (1) the student's proficiency status on the objective currently being tested, (2) the level at which he achieved or failed to achieve that status, and (3) the structure of the unit being tested. The MANAGER also controls item presentation. That is, it assures that testing on an objective will continue until a proficiency decision can be reached at specified levels of confidence. Its final function is to summarize response data generated during testing for output to the student and teacher.

Figure 2

Execution Model for Pretests and Posttests Using Item-Cluster Generators



PARAMETER CONTROLLER. Given that many item forms may be required to adequately test a single objective, it follows that the number of item forms required for a particular unit test will be quite large. Consequently the PARAMETER CONTROLLER specifies the

values of the parameters that are required for generating items of a precise form. For a particular objective, it controls both the forms used for item generation and the frequency with which they are used. Its role will become clearer later in the paper.

ITEM GENERATOR. Assuming that the CONTROLLER has fixed the values of the parameter in preparation for generation of an item, the GENERATOR processes the assigned values and generates the numbers required for constructing the desired item.

ITEM ADMINISTRATOR. Once the numbers required for the construction of a specific item have been generated, the ADMINISTRATOR presents the item to the examinee according to specified format and then scores his response.

DECISION MAKER. After the examinee's response to a specific item has been processed, the DECISION MAKER combines this newly obtained data with information generated by the examinee's prior responses to items testing the same objective. Prior to testing, the test builder specifies his levels of tolerance for Type I and Type II classification errors. He also selects the proficiency criteria he will use to determine cut off points for arriving at a decision as to the examinee's competency on a particular skill. A description of how this was accomplished in a previous study is reported by Ferguson (1971). Incorporating all of the information described, the DECISION MAKER determines whether the examinee does or does not have proficiency in the skill or whether another item must be generated and processed prior to reaching a proficiency decision.

To further clarify the general test model just described, it will be studied within the context of its application to the level F multiplication unit exhibited in Figure 1. Construction of the test begins with a detailed analysis of each of the five objectives. The product of this effort is a set of item forms that, when applied, yield a set of items representative of all behaviors defined by each of the objectives in the unit.

It is often the case that a large number of item forms must be identified and tested if the behavior described by a given objective is to be thoroughly measured. For example, when Objective 1 is analyzed, it yields a large number of item forms. Table 1 contains three sample item forms representative of some of the behaviors implied by the objective. When applied, the individual item forms associated with the objective should be capable of generating items that exhaust the entire domain of items for the objective. In addition, a single item form should produce unique items, that is, items not duplicated by other item forms for the same objective. An examination of Table 1 reveals that each item form will yield items that are unique to the domain of Objective 1. Of course, since the three item forms presented are but a small subset of the total set necessary to define the objective, all of the items that could be generated by applying these forms would fall far short of exhausting the item domain for the objective.

When using this procedure for test construction, test constructors must face the problem of determining the level of specificity

Table 1

Examples of Item Forms for Objective
One of the Level F Multiplication Unit

Sample Item	General Form	Generation Rules[1]	General Description
43 $\underline{\text{x}22}$	A $\underline{\text{x}B}$	1. $A = a_1a_2$; $B = b_1b_2$ 2. Check: $a_2 \cdot b_2 < 10$ 3. Check: $a_1 \cdot b_2 < 10$ 4. Check: $a_2 \cdot b_1 < 10$ 5. Check: $a_1 \cdot b_1 < 10$	No Carries
27 $\underline{\text{x}13}$	A $\underline{\text{x}B}$	1. $A = a_1a_2$; $B = b_1b_2$ 2. Check: $a_2 \cdot b_2 \geq 10$ 3. Check: $a_1 \cdot b_2 < 10$ 4. Check: $a_2 \cdot b_1 < 10$ 5. Check: $a_1 \cdot b_1 < 10$	Single Carry to Tens' Place
67 $\underline{\text{x}12}$	A $\underline{\text{x}B}$	1. $A = a_1a_2$; $B = b_1b_2$ 2. Check: $a_2 \cdot b_2 \geq 10$ 3. Check: $a_1 \cdot b_2 \geq 10$ 4. Check: $a_2 \cdot b_1 < 10$ 5. Check: $a_1 \cdot b_1 < 10$	Single Carry to Tens' and Hundreds' Place

of the item forms for an objective. For example, whether or not 25 x 85.42 and 52 x 85.42 should be the output of two different item forms so that samples of both forms of items are included when Objective 3 is tested, is a problem that must be resolved to the

[1] Capital letters represent numerals whereas small letters represent digits. All digits, a and b, were sampled from $U = \{1,2,\ldots, 9\}$. This notation is in keeping with that proposed by Hively et al., (1968).

satisfaction of curriculum specialists and testing experts. Its solution is likely to be achieved as a consequence of experience with intuitive choices that are made in situations to which the procedure has been applied. For example, if such test construction procedures are used to develop instruments for a program of individualized instruction, experience with the tests and their success or failure at providing adequate diagnostic information as input for making instructional decisions will provide input as to whether it is or is not necessary to re-define the specified item forms in the interest of improving the quality of the information generated by the test. Thus, in such a setting, the problem reduces to determining which item forms should be included during testing and what weight particular item forms should receive; that is, whether a particular item form be used more often than another, and in what order the presentation should take place. All of this is accomplished prior to test construction and should be guided by whatever information is at hand for the test constructor.

After the item forms defining unit behaviors have been identified, the next step in test construction is the specification of the parameters that, when supplied to the ITEM GENERATOR, will produce the two factors, multiplier and multiplicand, for a particular item. Some of the parameters specified for both multiplier and multiplicand are: (1) number of digits, (2) sign, (3) decimal point placement, and (4) zero placement. Another parameter permits one to specify the place(s) within the problem to which carrying occurs.

The final step in test construction is the fitting of the branching process to the unit hierarchy. This is accomplished with the assumption that, for the typical examinee, the amount of branching required during the course of testing can be reduced if measurement begins with some objective in the middle of the hierarchy. For the unit of interest, each examinee is initially tested on Objective 3 and then branched to a lower or higher order objective in accordance with the decision resultant from analysis of cumulative item responses for the objective. If, according to specified criteria, the examinee failed to evidence proficiency in Objective 3, he might be branched to Objective 1 or Objective 2, in accordance with his response pattern. If he demonstrated proficiency in Objective 3 he would be branched for testing first on Objective 4 and then on Objective 5. Both skills would be tested since Objective 3 is prerequisite to each and neither is prerequisite to the other. The TESTING MANAGER controls the branching in accordance with the information that it receives from the DECISION MAKER.

To summarize, the TESTING MANAGER initiates testing with an item on Objective 3. The particular item presented to the examinee is constructed by the ITEM GENERATOR in accordance with parameters specified by the test builder by way of the PARAMETER CONTROLLER. Options for the values of the latter variables are determined prior to test construction by analyzing the item forms required to test the stated behavior. After each item is presented and scored by the ITEM ADMINISTRATOR, the DECISION MAKER determines whether or not the examinee's proficiency status can be declared.

If a decision cannot be made, the TESTING MANAGER calls for the generation of another item on the same objective. The next item and any following it are chosen so as to guarantee representativeness of the item forms used in testing the objective. If a decision can be made, the TESTING MANAGER assumes control and branches to test another objective. When all necessary testing has been completed, the MANAGER summarizes the student's performance and presents a list of the objectives that he has not yet acquired. It further matches the items answered incorrectly with the item forms that generated them and presents the teacher with a detailed list of available instructional resources designed to teach the objective that corresponds to the item forms for which errors were recorded.

An essential feature of this test model is that it makes possible the modification or updating of tests with relative ease. Any of the five components of the model can be revised independently without affecting the others. In other words, changes in curriculum materials, proficiency criteria, or objectives will not necessitate a complete re-programming of the test model or the tests constructed by applying it. Only minor modifications are likely to be required.

In the interest of investigating the benefits that may accrue from applying the full resources of a small computer to activities directed at making it an integral part of the operation of an individualized school, the LRDC, with support provided by the National Science Foundation, has undertaken a five year study that calls for computer assistance for testing, instruction, and classroom management. Now in its second year, current project activities include the construction of pretests and posttests like those described.

For a given unit, nearly every child takes a pretest and at least one posttest. If multiple posttests are required, it becomes necessary for children to repeat tests that they have taken previously. With the computer testing procedure just described, no two administrations of the test produce exactly the same test. Since the test items are generated representatively and the numbers used to build each item are generated randomly, all tests are unique but equivalent.

Additional Benefits of the Item Generator

The procedure for item generation just described and the context in which it was discussed, facilitating pretesting and posttesting, suggest several other significant functions to which it may be applied. First, such a procedure would facilitate the generation of problem pages in unique but equivalent form and in any combination desired by a student or teacher. Such assistance could relieve a real burden that falls on the shoulders of teachers in an individualized classroom, the collection and management of materials that assist instruction.

A second possible role for the item generator is as an agent for generating and collecting data needed to determine how precisely objectives and/or item forms need to be specified. A procedure that makes possible the rapid generation of an item representative of any of a tremendous number of item forms encourages investigation of the relationships among these forms.

Finally, the use of the item generator itself aids in the refinement of the curriculum and associated instructional materials

because it demands a thorough examination of the relationships among the objectives, how they are taught, and what is tested.

REFERENCES

Anastasio, Ernest J., Marcotte, Donald M., and Fremer, John. Computer-Assisted Item Writing-II (Sentence Completion Items), Test Development Memorandum 69-1. Princeton, New Jersey: Educational Testing Service, 1969.

Ferguson, Richard L. A Model for Computer-Assisted Criterion-Referenced Measurement. Education, 1970, 81.

Ferguson, Richard L. Computer Assistance for Individualizing Measurement. Pittsburgh: Learning Research and Development Center, University of Pittsburgh, 1971.

Forbes, Dean W. Implications of Computer-Based Test Development for a Total Testing Program. Paper presented at the annual meeting of NCME, Minneapolis, Minnesota, 1970.

Fremer, John and Anastasio, Ernest J. Computer Assisted Item Writing-I (Spelling Items). Journal of Educational Measurement, 1969, 6.

Green, Bert F., Jr. Comments on Tailored Testing. Paper presented at the Conference on Computer-Based Instruction, Learning, Testing, and Guidance. Austin, Texas, 1969.

Hively, Wells, II, Patterson, Harry L., and Page, Sarah H. A "Universe-Defined" System of Arithmetic Achievement Tests. Journal of Educational Measurement, 1968, 5.

Linn, Robert L. Computer-Assisted Testing. Symposium on the Computer Utility--Implications for Higher Education. Manchester, New Hampshire, 1969.

Osburn, Bart. Comments on Computer-Assisted Testing. In Harman, Harry H. et al., Computer-Assisted Testing. Princeton, New Jersey: Educational Testing Service, 1969.

Smith, Robert E. Examination by Computer. Computers in Behavioral Science, 1963, 8.