DOCUMENT RESUME

ED 111 358 IR 002 406

AUTHOR	Olympia, P. L., Jr.
TITLE	Repetitive Domain-Referenced Testing Using Computers: the TITA System.
INSTITUTION	Washington Technical Inst., Washington, D.C.
PUB DATE	Jun 75
NOTE	9p.; Paper presented at the Conference on Computers in the Undergraduate Curricula (6th, Fort Worth, Texas, June 16-18, 1975); Related document is IR 002 399
AVAILABLE FROM	Entire Proceedings; Ted Sjoerdsma, Treasurer, CCUC, 124B (Lindquist Center for Measurement, Iowa City, Iowa 52242 ($10.00; Checks payable to the University of Iowa)
EDRS PRICE	MF-$0.76 HC-$1.58 Plus Postage
DESCRIPTORS	*Computer Assisted Instruction; Computer Programs; Criterion Referenced Tests; Individualized Instruction; Multiple Choice Tests; *Test Construction; Testing
IDENTIFIERS	Computer Based Test Development; Computer Managed Instruction; Domain Referenced Testing; *Totally Interactive Testing and Analysis System

ABSTRACT

The TITA (Totally Interactive Testing and Analysis) System algorithm for the repetitive construction of domain-referenced tests utilizes a compact data bank, is highly portable, is useful in any discipline, requires modest computer hardware, and does not present a security problem. Clusters of related keyphrases, statement phrases, and distractors form minipools from which the computer generates items for a domain-referenced unit of instruction. Test items can take the form of multiple-choice, true-false, matching, and fill-in questions. A random number generator produces data for test items requiring numerical solutions, and the correct answer is computed from a coded formula so computational subroutines are not required for each test item. This component of computer managed instruction allows the instructor to key related items in the data minipool to learning resources and to code the resources themselves for inclusion in the data bank. Use of this system for elementary, secondary, or undergraduate courses can facilitate instructional management and result in positive effects on student morale. (CH)

REPETITIVE DOMAIN-REFERENCED TESTING USING COMPUTERS:

THE TITA SYSTEM

by

P. L. OLYMPIA, JR., Ph.D.

Associate Professor
Allied Health Science Department
Washington Technical Institute
4100 Connecticut Ave., N.W.
Washington, D.C. 20008

Presented before the 1975 Conference on Computers in the Undergraduate Curricula (CCUC/6), 16-18 June 1975, Texas Christian University, Forth Worth, Texas.

INTRODUCTION

Testing, perhaps more than any other single educational activity, traditionally has been viewed as a "necessary evil". In general, students view it with anxiety while many teachers, at all levels of instruction, consider writing and grading examinations a tedious clerical task that takes away valuable time which otherwise could be used for more rewarding activities, e.g., individual student counselling. That testing is necessary in education is at best a mute question since it provides one of the most straightforward means of quantifying a student's achievement. That it is, at the same time, viewed with undeserved trauma bespeaks its own shortcomings.

Most students' anxieties over an examination arise from fear that they "do not know what will be asked", as though the test is a guessing game where one's chances of success depend as much on luck as one's mastery of the subject matter. Removing the "guessing" out of examinations is an important feature of domain-referenced (as opposed to norm-referenced) testing where test items are confined to predefined behavioral objectives and cover only the degrees of mastery, e.g., knowledge, comprehension, etc. in Bloom's taxonomy[1], expected of the student. It is no accident that students in PSI (personalized system of instruction) courses possess a healthy attitude toward examinations; the mastery model requires that a student be given only domain-referenced examinations for (in principle) as often and as long as it takes for him to gain complete mastery of a course unit. Failed examinations carry no penalty except for lost time. Unfortunately, an instructional strategy, such as PSI, that attempts to individualize instruction entails a considerable amount of time and effort for course management. It is precisely within the province of instructional management that the computer is making its most dramatic impact upon education in recent years. Central to any successful computer-managed instruction (CMI) system is the capability to generate repeatable domain-referenced examinations. While computerized test construction has become increasingly popular in recent years[2] many of the techniques in use today are either modest modifications of the simple item-retrieval system or else involve the construction of a massive data bank and/or a complex set of highly specialized subroutines.

At Washington Tech our CMI strategy centers upon a system called Totally Iterative Testing and Analysis (TITA) which incorporates a versatile algorithm for the computer generation of repeatable examinations. The algorithm is superior to many others in current use. The algorithm's attractiveness is fourfold: (1) it utilizes an extremely compact data bank; (2) it is highly portable; it is useful for any discipline since it does not depend heavily on the compact data bank; (3) it requires modest computer memory or hardware, and (4) it does not present a security problem of any kind. Since the procedure was first reported[3] we have made significant strides toward increasing its flexibility.

This paper describes the improved technique for generating repeatable examinations and provides an overview of our CMI strategy where the test generation procedure plays a key role.

TEST GENERATION ALGORITHM: GROSS FEATURES

The versatility of the algorithm essentially results from the fact that the data bank does not contain the full test questions but only their parts so that the computer has the freedom to assemble these parts to produce a logical test question in a completely random fashion. In order to see this more clearly, consider a multiple choice question as essentially consisting of two parts, namely, the question stem and the list of choices. The latter includes the correct answer and several distractors. Now, the question stem itself may be divided into two parts: (1) the keyword or keyphrase, and (2) the "statementphrase". Thus, in the simple test question

ALBANY IS THE SEAT OF GOVERNMENT OF
(A) ALABAMA (B) NEW YORK (C) NEW HAMPSHIRE (D) MAINE

the question stem contains the keyword, "Albany", and the statementphrase, "is the seat of government of".

The data bank consists of clusters of related keyphrases, statementphrases and distractors, which, when taken together, constitute a data "mini-pool".

For example, the test item just cited could be formulated from a mini-pool such as:

KEYPHRASE POOL	STATEMENTPHRASE POOL	DISTRACTOR POOL
Albany	is a city located in	New York
Montgomery	is the capital of	Alabama
Augusta	is the seat of government of	New Hampshire
Concord		Maine
Columbia		North Carolina
		South Carolina

It is clear that a multiple-choice question may be formulated from a given mini-pool by selecting one member of the keyphrase pool, joining it with one member of the statementphrase pool and selecting any three (or four) distractors in addition to the correct answer to make up the list of choices. The order in which the distractors and correct answer appear in the list of choices is, of course, always random. Occasionally, depending on the current output of the random number generator, an additional choice, e.g., "(E) NONE OF THESE", may be included. By design, the probability of this being the correct answer (in which case, the other four choices must all be distractors) is extremely small.

There is no limit on the number of pool members; indeed, greater variability of generated test items results from the inclusion of more members. In addition, each pool member may be a single word or a group of words totalling several printed lines so that one is not limited to the formulation of short test items.

The mini-pool cited above is an example of a "simple" mini-pool in that one correct answer is associated with a given keyphrase regardless of which statementphrase is used. This is because the members of the statementphrase pool are analogous to each other. It is instructive to note that our technique is not limited to the creation of simple mini-pools. The repeatability of the computer generated test and the compactness of the data bank arise, in fact, from the use of "complex" mini-pools where the correct answer associated with a given keyphrase depends upon which statementphrase is used. An example of a complex mini-pool is the following:

KEYPHRASE POOL	STATEMENTPHRASE POOL	DISTRACTOR POOL
A solution of 0.001 M H_2SO_4	has a pH of about	2.7
A 0.01N solution of sulfuric acid	has a pOH of approximately	11.3
		3.0
A 0.1N KOH solution		2.0
A 0.001 M HCl solution		11.0
		12.0
		1.7
		12.3

To accomodate test items whose question stem cannot be divided properly into a keyphrase and statementphrase, the data bank may also contain items with only a question stem and a distractor pool as in more traditional approaches.

TRUE/FALSE, COMPLETION AND MATCHING TYPE TEST ITEMS

Additional flexibility is achieved by realizing that a multiple choice question can be recast both as a true/false or fill-in-the-blank question. In the former case, the question is false if any one of the distractors is used to complete the question and true, if the correct answer is used instead. Since the probability of formulating a "true" test item is comparatively small, students are required to either correct the test item or else specify what is wrong with it whenever their answers are "false".

To formulate a fill-in-the-blank question the computer uses one member of the statementphrase pool and joins it with one member of either the keyphrase pool or the distractor pool. The third pool member is not used (except for constructing the key) since the blank is in its place. In other words, a completion test item always presents two possibilities. We can either place the blank where the keyphrase is supposed to go or else where the correct answer (or distractor) is supposed to go. Generally, there is less or no ambiguity in the latter. For example, in the question intended to read as

GROUP VIIIA ELEMENTS ARE CALLED NOBLE OR INERT GASES

the two possibilities are: (1) _______ are called noble or inert gases, and (2) Group VIIIA elements are called ________. The first case is ambiguous in that its blank may be filled correctly by any one of the following: (a) He and Ne (or various combinations of Group VIIIA elements); (b) The chemically unreactive group of elements; (c) Elements with zero valence.

The division of a test item into a keyphrase, statementphrase and distractor portion readily permits its being included in matching-type sets of questions. To accomplish this, an entire mini-pool is gathered and the members of the statementphrase pool are discarded. The test items then involve matching the set of keyphrases with the corresponding set of distractors. It should be noted that members of complex mini-pools are not normally suited for formulation as matching-type test items.

It is clear that our technique allows the computer to present a given test item in any one of several different types using exactly the same compact data bank.

THE DATA BANK

The data bank is arbitrarily divided into sections corresponding to each unit of a course associated with specific course objectives, viz, the domain. Test items in the bank are either "root" or "branch". Each item has a unique question number and a keyphrase. If the item is a root item, it also contains the statementphrase pool and the distractor pool. A branch item is associated with a specific root item and appears in the data bank without either a statementphrase pool or distractor pool. Thus, the branch item is much shorter and simpler than a root item and needs only to point to its associated root item in order to obtain the required pools. Since there is no limit on the number of branch items associated with a given root item, the compactness of the data bank is evident.

NUMERICAL PROBLEMS

The test generator is now also capable of formulating test items requiring numerical solutions where the required data are produced by the random number generator at execution time. This feature enables one to construct truly repeatable examinations while still preserving the compactness of the data bank and the portability of the entire package. The correct answer is computed by the machine from a coded formula so that one does not need to write a separate subroutine for each test item. The data bank entry for this type of test item is different from the others which are formulated from data mini-pools. Here, the question is written in full, except that the pound sign (#) appears where a number is to be randomly generated. For example, a test item may be

MOLES OF AN IDEAL GAS AT # °C AND # ATM OCCUPY A VOLUME (LITERS) OF

In the bank this particular question is followed by six numbers with the correct number of significant digits, which define the upper and lower bounds for the three data. These are, in turn, followed by a coded formula which the program uses to compute the correct answer, e.g., 1*0.082*(2+273.1)/3, where fixed point numbers refer to the internally generated data and floating point numbers are numeric constants which are part of the formula. Test items such as this still may be cast as multiple choice, true/false or fill-in-the-blank.

MANAGING INSTRUCTION WITH THE TEST GENERATOR

The fact that test items in the bank are all domain-referenced, in addition to the clustering of related items in data mini-pools, greatly facilitate the management of instruction. Each mini-pool is keyed to a set of learning resources which a student may consult for more information or in-depth study. The learning resources key may be included in the data bank, although we use a separate file for this purpose. The learning resources are coded with a five-digit number where the first digit identifies the type of learning resource, e.g., film, cassette, CAI module, textbook, and the remaining digits provide specific information about the resource, e.g., pages of a reference book, film number, etc.

A student, learning for mastery, takes a computer-generated test and hands his completed paper to his instructor or proctor who then feeds the student answer sheet to an optical reader and grader program. The grader program, not only scores the test, but also updates the student's course history, performs the required statistics, such as item analysis, and uses the learning resources file to prescribe a program of study. The cycle may be repeated if necessary since the test generator is capable of providing a set of unique but equivalent examinations.

What a CMI of this sort does for the student morale is inestimable. Its success as an instructional strategy is guaranteed by at least two factors: (1) the impairment of student learning brought on by the usual delay between examination time and the time the graded paper is returned to the student is avoided; (2) the unique examination each student gets serves as a valuable study aid. It eliminates student anxiety over examinations, and also accelerates student learning by encouraging students to discuss course materials among themselves.

REFERENCES

1. B.S. Bloom, "Taxonomy of Educational Objectives. The Classification of Educational Goals. Handbook I: Cognitive Domain", Longmans, Green and Company, New York, 1956.
2. See for example, G. Lippey, EDUC. TECH., 13, 10 (1973).
3. P.L. Olympia, Jr., Proceedings of the First National Conference on PSI, 118, 1974 (Georgetown Univ.); EDUC. TECH, 15 (6), 153 (1975).