FROM DOMAIN-REFERENCED CURRICULUM EVALUATION TO SELECTION OF EDUCATIONAL MICROCOMPUTER SOFTWARE

Wells Hively

CSE Report No. 236
1984

Center for the Study of Evaluation
Graduate School of Education
University of California, Los Angeles

The project presented or reported herein was in part performed pursuant to a grant from the National Institute of Education, Department of Education. However, the opinions expressed herein do not necessarily reflect the position or policy of the National Institute of Education, and no official endorsement by the National Institute of Education should be inferred.

Table of Contents

Introduction

Much of my past work has been in the field of domain-referenced testing and curriculum evaluation. Some of that work took place in a happy association with the UCLA Center for the Study of Evaluation, which published one of our contributions to this field as the first of its Monograph Series in Evaluation (Hively, Maxwell, Rabehl, Sension, & Lundin, 1973). Those of you who know this work probably will not be surprised at the approach we are now taking to the selection of educational software: compare, contrast, classify, and try to avoid over-generalization.

Currently, we are concerned with evaluating microcomputer programs that can enhance instruction during the period of schooling when it is easiest to consider the curriculum as a whole: preschool through grade 9. We have formed a publishing company to assemble and transmit information about educational microcomputer software to schools. Our purpose is to help school people more easily find what they need and use it more effectively. Specifically, we want to help teachers answer the following questions:

What kinds of programs are currently being developed?

How can we find good ones?

How can we use them effectively in school?

How can we tie them into the basic school curriculum?

We assume that there are many different types of educationally useful microcomputer programs, each with its own practical purpose, each derived from its own theoretical assumptions and each, therefore, requiring its own unique set of evaluative criteria. We also assume that the lesson-plan settings in which teachers use the programs have at least as much influence on their impact as the characteristics of the programs themselves. Consequently, useful evaluation must take account of both the characteristics of the programs and the ways in which they are used.

The terminology used to classify different types of programs has by no means settled down. To make matters worse, the terms often carry evaluative connotations. Currently, outside the military, "CAI" (computer-assisted instruction) is a low-status term. "Drill and practice" is out. "Simulations" are in. "Learning games" are in. "Computer literacy" is definitely in. But all these terms are operationally hazy. It's important to try to clarify the terms we use to classify programs, because the classifications govern our approach to evaluation: programs are relatively easy to compare and evaluate within the same class, but very difficult to compare across classes. Let me give you examples of the different classes of programs we are encountering, and suggest some more precise nomenclature for them.

Types of Programs

Domains of Practice

A simple and generally useful type of program may be called "Domains of Practice". A good example is a program published by Sunburst Communications called Smart Shopper Marathon. The purpose of the program is to provide practice in rapid arithmetical estimation. The setting is an imaginary supermarket. Each so-called "aisle" represents a different domain of practice. In aisle 1, the student has to rapidly estimate the results of dividing a decimal by a whole number. The student's job is to answer as many problems as possible in a given time, so quickly that detailed calculation is a hindrance: rounding off and estimating is the skill that must be practiced. In aisle 2, the student has to estimate the results of multiplying whole numbers. In aisle 3, subtracting decimals; in aisle 4, comparing fractions; in aisle 5, multiplying whole numbers times decimals.

The problems in each set are generated in random order, and each time you use the program the "aisles" appear in a different order. Therefore, because students are not likely to memorize rote sequences, the program lends itself to repeated practice without boredom. The scores used to judge youngsters' progress are based on a combination of speed and accuracy.

Programs like Smart Shopper Marathon are characterized by items drawn from clearly defined domains of knowledge or skill, a high frequency of opportunities to respond per unit time, and almost total absence of instruction presented by the program itself. The teaching of the constituent skills and the orientation to the problem-solving approach must come from an outside source. The students must obtain guidance in strategies for estimation from their teacher, or from each other, working in a small group. Therefore, programs like this make good vehicles for classroom demonstrations and for small group discussion. They provide external focus and feedback around which classroom activities can be assembled. Increased performance on the domains of practice presented by the computer may become the criterion toward which teacher and students can work together, a welcome change from the teacher's usual job of standard setter and task master.

Tutorials

It is useful to compare programs of the domains-of-practice type with programs of a second type that have historically been called "tutorial". A tutorial program developed by the Minnesota Education Computing Consortium leads up to the geometrical definition of an angle. What the student encounters in programs of this class is a fixed and predetermined sequence of presentations of bits of information interspersed with questions and answers. The term "tutorial" is too broad to clearly denote this type of program. Let's narrow the terminology to "linear tutorial". This is the classic programmed-instruction format which most people associate with the so-called "CAI" that is currently out of fashion. Perhaps one can see the reason why. The frequency of opportunities for students to respond in linear tutorial programs is low in comparison to the rapid-fire opportunities provided in domains of practice. Nearly always, the expository material could be conveyed faster in a book or in a conversation with a teacher or a peer. Perhaps most important, the sequence of "telling and testing" arises totally out of the mind of the designer, with no elbow room given for the idiosyncracies of different learners.

In constructing domains of practice, we are on fairly safe ground because we are creating models of subject matter. The theory and methodology of domain-referenced testing provide a fairly solid foundation for this job. But in linear-tutorial programs we are attempting to model the dynamics of teacher-student interaction without actually allowing dynamic interaction. There is little theory to guide this task, and successful programs of this kind are hard to find. Perhaps artificial-intelligence theory will eventually help us construct truly dynamic tutorial programs of the Socratic or error-analysis type. But good, dynamic tutorial programs have not yet filtered down to the practical level where we encounter them in our survey. For practical purposes, teachers can do much better by putting small groups of children to work on domains-of-practice programs, and letting the teaching arise from class discussion and spontaneous interaction, than by sending individual children off to have learning doled out to them in small droplets by step-by-step, linear-tutorial programs.

Education Games: Extrinsic

An enormously popular third type of program is the education game similar to those seen in video parlors and arcades. Basically, these games are domains-of-practice with several added attractions. This combination may be called extrinsically-motivated practice or extrinsic games. Some of them are a lot of fun.

An example of an extrinsic game is the DLM Company's Alligator Mix. If your answer in the alligator's belly matches the problem in the apple that comes floating in from the left side of the screen, you win by opening the alligator's mouth and swallowing it. If it doesn't, you leave the mouth closed, and it bounces off and spins away. At the beginning, there is just one alligator, at the bottom of the screen, and the apple has to travel a long way, so you have plenty of time to make up your mind. After a string of successes, a new alligator surfaces. The apple doesn't travel as far to get to that alligator, so you have to think faster. When there are four alligators lined up, you really have to hustle. The teacher or student can choose from nine skill levels, which have to do with the velocity of the apple's motion, and three problem ranges which have to do with different domains of practice.

Another example from DLM is called Demolition Division. The tanks all come forward at the same time, shooting at your forts. Your job is to position a 0 underneath the cannon that aims at the most threatening tank. Then change the 0 to a number that corresponds to the questions on the tank, press the space bar, and destroy the tank before it knocks down the wall and destroys your cannon.

What makes these games fun is delicate grading of speeds and levels, freedom to select levels that match entering skill, and richness of alternatives in ways to respond. A whole art and technical literature is growing up in the area, and standard "plot formulas" are rapidly appearing.

Another kind of plot formula for an extrinsic game is demonstrated by Sunburst's Math Mansion. It gets good mileage out of a "dungeons and dragons" theme. The thematic development and the richness of alternatives in Math Mansion trade off against relatively low frequencies of opportunities to respond. We are a long way from knowing, if we ever will know, what are the optimum mixes of such ingredients. But youngsters identify the good examples by their attention and their resultant learning.

Education Games: Intrinsic

A fourth category of program might be called, by way of contrast, intrinsic games. QED Company's Arith-magic program is called "Diffy". The student volunteers a set of four numbers which the computer places at the corner of a square. Then the student goes around the square finding the differences between each pair of numbers. The differences found in the first round then form the corners of a new square for the second round, and the student goes around finding the differences again. This goes on until, eventually, lo and behold, the differences all come out the same. The challenge is to figure out what characteristics of the starting numbers make the differences converge quickly or slowly. The game provides a vehicle for discussion, exploration, and curiosity, and incidentally provides a very high frequency of opportunities for subtraction practice.

Other examples of intrinsic games are Sunburst's Teasers by Tobbs and MECC's Taxman and Bagels. Games like these tap into the whole realm of classic puzzles and brain twisters, some of which lend themselves nicely to computer presentation. As usual, the most frequent examples tend to be in the field of mathematics, but there is no reason why they need to be limited to that field.

Intuition Builders

A fifth promising category of programs is exemplified by two in the Milliken Company's Edufun series: Golf and The Jar Game. Let's call them intuition-building programs. In Golf the problem is to direct the ball from the tee to the green by estimating an angle and a distance using a compass rose for reference, and a given unit of distance. If you lead off, you must estimate distance and direction absolutely (in terms of the compass rose and the unit of distance), but if you shoot second, you can correct your direction by adding or subtracting degrees to the course taken by your opponent's ball, and you can correct your opponent's estimate of distance. The game builds up a nice intuitive judgment of angles and directions.

In a similar vein, The Jar Game builds upon the intuitive statistical notion of drawing beads out of a jar. The young student is then shown diagrams of jars containing different proportions of two kinds of candy pieces. The job is to figure out on which jar of candy a randomly-directed fly will land more often.

There are many other potential examples of geometrical and statistical intuition-building activities that computer experiences could enhance. The ease and speed with which the computer can generate these examples is delightful. We have come a long way from the old days of having children estimate the number of raisins in an average slice of raisin bread by taking apart a loaf of bread and counting the raisins in selected slices.

Simulations

The sixth category is simulation programs. There are so many different kinds of simulations, and they can produce so many different outcomes, that this category will no doubt be subdivided later, but the characteristics that guide subdivision are not yet clear. The MECC Sell Series, built around the famous Sell Lemonade, is an example. The simplest one of the series is called Sell Apples.

When youngsters are turned loose on a program like this, they may learn many different things, depending on the context provided by the teacher. They may learn to read carefully and follow instructions in detail. They may learn to interpret data in tabular form. They may build up an intuition about the relationship beween price and volume of sales. They may learn important habits of record keeping. They may learn to transform data into graphical form and interpret trends. At a deep level, they may learn some important strategies that underlie scientific method, such as choosing extremes of variables and narrowing down to find the maxima and minima. They may even learn something about the cost benefit of seeking truth. None of these things is taught for sure by the program. They depend on the context provided by the teacher and other students. It is particularly obvious, in the case of simulation programs, that validity and usefulness depend as much on the context provided by the teacher and peers as on the programs themselves.

Information Retrieval Programs

MECC also provides a nice example of the seventh category:

information retrieval programs. Nutrition asks you to provide a list of your food intake for one day. Then it gives you a nutritional analysis: how well your day's food intake represents the four basic food groups, how your numbers of calories provided by fat, carbohydrate, and protein compare to the recommended number of calories for a person of your age and stature, and how your intake of iron, calcium, vitamin A, and vitamin C compares to the recommended daily requirements. This is what the computer does best, and its role in this kind of instruction is distinctive. MECC's Nutrition program does not provide a means of accessing or adding to the nutritional data base, but one can easily imagine programs to which teachers and students might add information for foods not currently included, or ask for other kinds of analyses; a nice meeting ground between specific subject matter and computer literacy.

Nutrition is a miniature data base, and elsewhere a wonderful array of useful data bases is becoming available to teachers and students. Compuserve, for example, is a service that enables computer owners to obtain information from many data bases at night--airline schedules, weather reports, and so on. The potential of data bases such as these as vehicles for instruction is tremendous. Answering questions that come up in class by accessing a nutritional data base overnight would be a considerable step up from writing letters to the Department of Health.

Tools and Displays

The eighth and last category is such a large and heterogenous category that it, too, will no doubt soon be subdivided. For now, let us call it "tools and displays". In this category are all the programs that perform helpful calculations, the programs that process words, programs that display graphs of changes in phenomena detected by sensing devices like thermometers, and programs like the famous Logo that offer environments with important properties to be explored. The educational utility of these programs is limited only by imagination and experience. The following is just one example.

A program produced by Spinnaker Software called Delta Drawing is a kind of baby Logo. The commands are easy to understand, and a small child can start making interesting pictures almost immediately. We start with an arrow, move it forward by pressing the D key, and turn it to the right by pressing the R key. We change the color of the line by pressing the C key and then a number corresponding to the color we want. We move it forward again and change its direction and color. We store all the preceeeding steps in a sub-routine which can be repeated. We repeat the sub-routine to generate a kind of rose window. We may fill the spaces on the screen by choosing a color and then pressing Control F. The computer keeps track of all these steps as a string of symbols, and we can switch to text mode from the graphics mode and examine the string, operate on it, and go back to graphics to see the results.

Programs like Delta Drawing offer nice opportunities to explore symmetries and artistic effects. For example, when a line passes beyond the border of the screen, we have a choice of having it wrap around and appear from the opposite border or having it bounce off at an equal angle. The line bounces and bounces again like a billiard ball. It continues bouncing to generate a symmetrical pattern.

It is also possible for a repeated figure to wrap around and then bounce to produce a complicated effect. It goes on bouncing and creates an interesting artistic result made up of a combination of expected and surprising features. There is considerable potential in programs like these in the hands of teachers with sensitivity to some of the relationships between art and mathematics.

The overwhelming impression one gets from watching children work with all the foregoing different types of programs--ranging from open-ended environments, like Delta Drawing, to practice sequences like Smart Shopper Marathon--is that their effectiveness depends at least as much on the classroom context in which they are used as on the properties of the programs themselves. Properties of programs which are drawbacks in one context may be benefits in another, and exciting uses may be totally unanticipated by the people who developed the programs.

A Curriculum Guide

With the foregoing review of types of programs as background, now let me tell you about the product of our work: a curriculum guide for grades 0-9 called Hively's Choice. The target audience is what you might think of as "second wave" educators--not the original enthusiasts, but the experienced and thoughtful mainstream teachers on whom any successful educational innovation depends.

Several characteristics distinguish the guide from other efforts to help teachers evaluate and choose software. First, the guide only contains software that has been found to be particularly outstanding in quality and ease of use. Second, the guide is designed in such a way as to make it as easy as possible for teachers to connect the recommended software to curricula and lesson plans. This is done in several ways. The user may begin by looking at a chart showing where each of the programs fits into general subject matter areas and the grade levels over which it is likely to be useful. Next, the user turns to a set of quick descriptions, organized by subject matter, within grade levels, and arranged so that one can look through them very rapidly so as to maximize chances of discovering unexpected connections to upcoming lesson plans.

The reader who finds something of interest by perusing the quick descriptions may turn to a detailed description of that program. There, the goal is to describe the program in enough detail that one can intelligently decide whether it would really be useful and exactly how it would relate to ongoing curriculum.

A subsection of the detailed description called "Curriculum Connections" includes words and phrases that can be used as cross references to scopes and sequences. "Objectives" briefly describes the kinds of learning which may be enhanced by the program, and a section called "Instructional Examples" gives recommendations about how best to utilize the program in classroom discussion, small group or individual work. "Estimated Engagement Time" helps teachers plan how much time to allow for work on the program by the whole class, small groups, or individuals.

Also in the detailed description section one may find the technical information about the program, the hardware it requires, and the name, address and telephone number of its producer. The rest of the book consists of cross indices by subject matter and topic, and an alphabetical listing to facilitate the location of specific programs.

Our goal is to find rich, engaging and easily usable programs that have solid connections to all the areas of the basic curriculum, preschool through grade 9, and that take all the different forms described in the earlier part of this paper. If you imagine the curriculum as a matrix of subject matter areas by grade levels, some of the cells in the matrix are already getting crowded while others are virtually empty. Over time, our goal is to weed out programs in the crowded cells so as to include only a selection of the most useful and interesting ones, and to seek entries in the empty cells. Each year the guide book will be revised, following the analogy of a European travel guide. Like a travel guide, each edition will be cumulative and self-contained.

Organizationally, this work is done by a small, carefully selected group of contributing editors, who work in schools and work very closely with teachers in training. These editors are chosen to represent areas of the country where thoughtful and interesting work with microcomputers is going on. In their daily work with teachers, the contributing editors keep an eye out for outstanding programs and interesting ways of using them. They forward their reviews to a small central editorial staff that produces the book.

In meetings of this editorial staff, we work to explicate the bases for our selections, to clarify categories of programs and the evaluative criteria applicable to each, to organize observations about effective classroom use of various types of programs, and to identify useful sequences and combinations of programs. This work aims to create a dialogue between good theory from the technical literature and careful observations of classroom use. From these, we are developing, year by year, a progressively more useful, readable, and balanced curriculum guide--one which can contribute both substance and diversity to the curriculum for preschool through the first nine grades.

References

Hively, W., Maxwell, G., Rabehl, G., Sension, D., & Lundin, S. (1973). *Domain-referenced curriculum evaluation*. Los Angeles: UCLA, Center for the Study of Evaluation.