Application of Experimental and Quasi-experimental Research Designs to Educational Software Evaluation

Eugene W. Muller

This article proposes a set of guidelines for the use of experimental and quasi-experimental methods for research (Campbell and Stanley, 1963) in educational software evaluation. The goal of applying these methods is to obtain empirical evidence of student performance, in order to determine if programs are making desired learning effects.

Rationale for This Type of Evaluation

If a program is intended to be educational, it would seem reasonable to conclude that students should *learn* when using it. One of the more sophisticated and comprehensive approaches to the evaluation of educational software is through a form developed by the Educational Products Information Exchange (EPIE) Institute, which examines many facets of a software product, including program content, program intents, program appropriateness for intended users, clarity, fairness and accuracy, graphics, audio, support materials, documentation, user control, feedback, and other aspects. In most cases, the EPIE form calls for the analyst's judgments of a program's effectiveness. While it does provide for some evaluator observations of student performance, *no empirical methods of analyzing program effects are prescribed.* To objectively and accurately determine the effectiveness of a program on student learning, systematic data-gathering procedures need to be planned and conducted, using direct performance data.

A number of examples of this type of empirical approach to software evaluation can be found in the computer-assisted instruction (CAI) literature. Daellenbach, Schoenberger, and Wehrs (1977) compared the effects of CAI to traditional teaching methods on cognitive and affective development through the use of experimental and control groups. Evans (1982) compared retention rates for learners who used CAI versus lecture and workbook based methods. Suppes and Morningstar (1969) compared the performance of students using CAI with control students in mathematics and Russian language instruction.

The chief advantage of these methods is the control over sources of invalidity. If we can study the effects of a program under controlled conditions, we theoretically can conclude with some certainty that any learning effect is a real one (Borg and Gall, 1971). The research designs discussed in this article and their applications to educational evaluation are not new (for a discussion, see Wolf, 1979), but the educational medium of instructional software *is* relatively new. The goal of this article is to develop some generalizations for the empirical evaluation of software, based upon the suitability of each research design to the type of software being evaluated, and on the circumstances under which the evaluation is being conducted.

Role of This Approach

These proposed methods will not act as substitutes for the approach used in the EPIE form, or any similar form. Instead, the information obtained by the implementation of these procedures should act as a complement to a range of information obtained about a software product. Coburn, Kelman, Roberts, Snyder, Watt, and Weiner (1982) suggest that there be four broad areas of concern when evaluating a program: (1) program content—the suitability of materials for the students and the objectives, and the accuracy and significance of the content; (2) pedagogy—the nature of a program's feedback, the program developer's assumptions of learning, and the types of learning modes used; (3) program operation—the control that users have when using the program, the program's quality, and the quality of the documentation; and (4) student outcome—the degree to which students learn what the program intends to teach, and the effectiveness of the program compared to non-computer-assisted instruction in the same area. The research methods described in this article are specifically addressed to this fourth area.

Limitations of This Approach

The limitations of these methods are the same that apply in any educational research study. Experiments in education are frequently carried out to test the effectiveness of materials. This may

Eugene W. Muller is Research Associate/Data Manager for the Second IEA Science Study, Teachers College, Columbia University, New York.

involve the use of a classic single variable design—that is, the manipulation of a single treatment variable to observe its effect on one or more dependent variables. The crucial aspect of these experiments is to maintain control over any confounding variables. However, holding such confounds constant in an educational setting is a more difficult task than in a laboratory setting. Instead, quasi-experimental procedures (Cook and Campbell, 1979) are employed as substitutes. The trade-off in these situations is that while both the experiment and the quasi-experiment are designed to control for some or all of the potential threats to internal validity (i.e., history, maturation, testing, instrumentation, regression, differential selection, mortality, and interactions), one sacrifices the generalizability of the results the more one attempts to control for such influences. More rigorous laboratory control will make the results less transferable to a field application. The goal in educational research is to attain sufficient rigor in order to make the results scientifically acceptable, while at the same time maintaining enough realism to make the results transferable to other educational settings.

Summative Versus Formative Evaluation

These methods would probably be more suitable for a summative evaluation of a software product. Summative evaluations are designed "to provide data for policy decisions about the adoption or discontinuation of the use of an instructional package" (Kandaswamy, 1980). They are conducted after a program's development, and sometimes use hard data, reliable instruments, and representative samples. The results of summative evaluations act as guides for the potential purchasers or users of the instructional materials. Formative evaluations, on the other hand, are conducted by the program developers to determine how programs can be modified, to make them more effective. In most cases, it would seem impractical to conduct a series of experimental studies (which, to perform correctly, would require an extensive amount of planning and preparation) for the formative evaluation of a program.

Types of Software to Be Evaluated

Two types of software will be discussed in this article: tutorial programs and drill/practice programs. It may be difficult to classify a program solely into one category or another, but in general tutorial programs are those in which the computer assumes responsibility for instruction, while drill and practice programs assume some prior instruction in the concept or skill addressed (Heck, Johnson, and Kansky, 1981). The appropriateness of each design to a particular program will depend upon how a program can be classified into one of these categories.

Pretest-Posttest Control Group Design

One approach to software evaluation would be the pretest-posttest control group design, in which learners are randomly assigned to one of two groups: an experimental group receiving treatment, or a control group not receiving treatment. Often the Analysis of Covariance is employed to analyze the results, which adjusts for any pretest differences between the two groups and compares the group means. A significant F-value indicates that the experimental group differed in performance from the non-experimental group. This a true experimental design, and is depicted in the following manner:

R O X O
R O O

where R represents random assignment, O represents a test score, and X represents the educational treatment. Subjects are randomly assigned to the treatment and control groups to cancel out initial differences. This design controls for most of the threats to internal validity (Campbell and Stanley, 1963). It would in most cases be applicable to both tutorial and drill/practice software. It would require that the conditions for subjects within each group are as uniform as possible with the exception of the treatment variable, which in this case would be the software to be evaluated. Possibly, the control group students could spend the treatment time normally allotted to the experimental group students using a non-educational computer game. As an alternative, one could have the control group undergo another type of educational treatment (such as extra individual or group instruction in the area of study covered by the program) to compare the outcomes of both methods.

Single-Group Pretest Posttest Design

What often occurs in the case of evaluation are situations that prohibit the use of randomization, or the creation of a control group entirely. Several alternatives are available. One is to use no comparison group at all, and merely administer the treatment to the entire group of learners, obtaining pretest and posttest measures. This design, known as the single-group pretest posttest design, is depicted in the following manner:

O X O

This design presents a number of difficulties in interpretation, including possible invalidity due to

maturation, history, regression, instrumentation, selection, mortality, and interactions. It is not used in laboratory studies, but is many times the best that an evaluator can implement. To employ such a design, one must eliminate, through logical means, each alternative explanation of the posttest results in order to conclude that the actual change in scores were due to the treatment. If, for example, the material to be learned is specialized and novel, such that the student could not have learned it elsewhere between tests, this would lend support to the hypothesis that any gains in posttest scores were due to the treatment. Drill and practice programs, which usually assume some prior instruction in the skill addressed, will probably not fit well into this category. Most likely, the subject matter covered by such software would be in an area that the student is already undergoing instruction in during his or her regular classes. Attributing the gain in performance to the program would be difficult, if not impossible.

However, if the program is tutorial in nature, and deals with subject matter that is specialized and novel to the student, it may be possible to rule out alternative explanations for the gain in performance, and instead attribute the test score differences solely to the program. This design should be employed with variables that are not likely to change unless some direct action by the researcher is taken to bring about such a change. Furthermore, it is recommended that the interval between pre- and posttesting be kept short to further insure that extraneous variables do not enter into the situation.

Nonequivalent Control Group Design

Another alternative to the problem of the inability to establish a comparable control group is to use what is called the nonequivalent control group design, which is identical to the pretest posttest control group design except that the subjects are not randomly assigned to groups.

O X O
O O

A number of techniques are available to establish the comparability of the two groups. This design is called a quasi-experiment, and one must interpret the results with caution. It probably could be used with both types of software, but again one must be certain that the conditions for both groups are as similar as possible except for the independent variable, which in this case is the program being evaluated. One must be particularly aware of the threat of "intra-session history" (Campbell and Stanley, 1963).

Time Series Design

A third approach available when the creation of a control or comparison group is not possible is the time series design. This is a quasi-experimental design and involves measuring a single group of subjects at periodic intervals, both before and after the introduction of the experimental treatment. If the experimental treatment has any effect, it should be reflected by a change in the test scores after the appearance of the treatment. This design is typically depicted in the following manner:

O O O O X O O O O

It is similar to the single-group pretest posttest design, except that the time series design uses additional measurements. This design rules out possible invalidity due to maturation and testing.

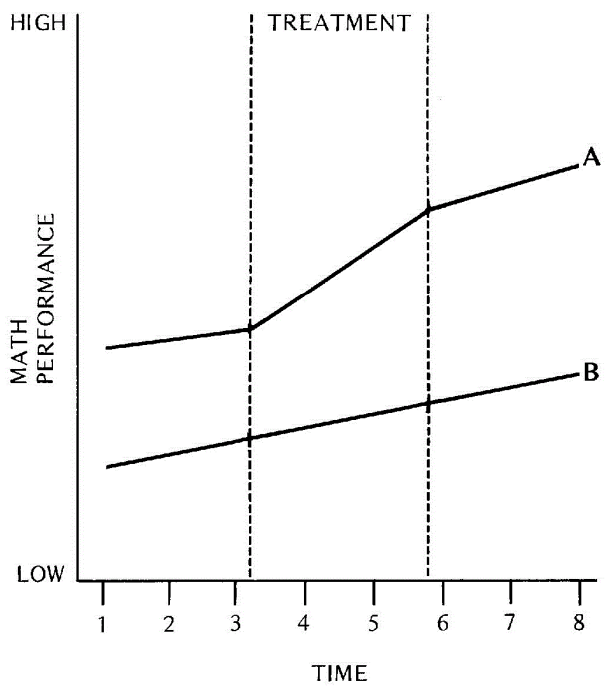
It is suggested that this design be employed when a change in the method of teaching is instituted. This could extend to the use of drill/practice software, which offers a change in a learning method from the traditional form of instruction. Consider a case where a school offers math software to a group of students who are periodically given standardized tests in mathematics. The effectiveness of the introduction of the program could be depicted graphically. Two possible outcomes are presented in Figure 1.

These hypothetical studies use a slight modification of the time series design, where treatment is continued for two observations after the introduction of the program.

Points in time of testing are indicated along the horizontal axis; performance ratings are indicated on the vertical axis. Results that appear similar to line A would suggest that the introduction of the program did have an effect on math performance, indicated by the fact that the level was considerably higher after the treatment was introduced. The results for B, on the other hand, suggest that the program had no effect, since the upward trend in performance was not altered in any way after the program was introduced (time series designs are not just analyzed graphically, but also through statistical analyses).

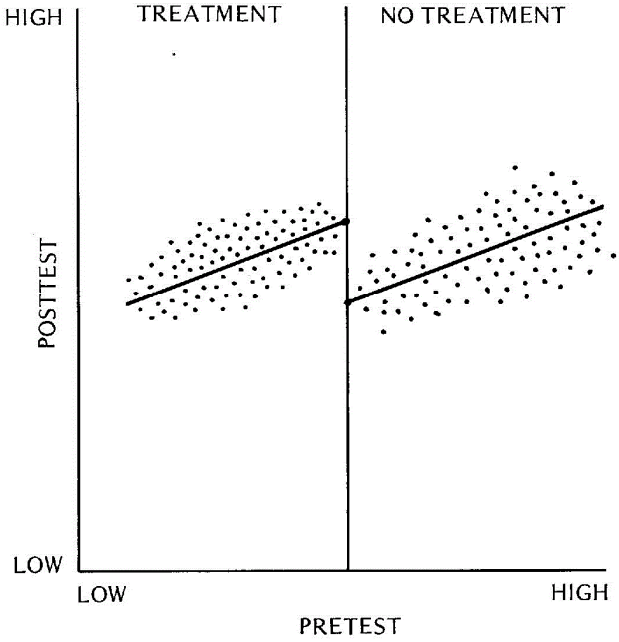
When applying this design to software evaluation, it may be possible that the change in performance is due not to the program but to the "reinforcing" effect of the computer. To control for this, one could have the students regularly use a computer before, during, and after the introduction of the software (here again, a non-instructional game could be used). In this way, any abrupt change in performance could not be attributed merely to the use of a computer, since this condition will remain constant over all mea-

Figure 1



Possible Outcomes of a Time-Series Design: Results depicted in line A suggest that the program had an effect on performance. Results similar to line B would suggest no effect.

Figure 2



Possible Outcome of a Regression-Discontinuity Design: The difference in predicted performance at the cutoff score reflects the treatment effect.

sures, and will be a natural part of the student's environment.

Researchers need be aware of several possible sources of invalidity with this design, including instrumentation and history. This design would appear to be most appropriate for schools that already have and are using computer hardware, and wish to evaluate the effect of a new piece of software once it has been introduced.

Regression Discontinuity Design

Suppose a program to be evaluated is one that is intended for only a *subgroup* of learners. An example of this type of software can be found in Fredriksen, Warner, Gillote, and Weaver (1982), who describe a number of experimental programs designed to improve reading skills. One such program, designed to improve the ability to recognize letter clusters, flashes words on a video screen in rapid succession and has the user determine whether or not a particular cluster is present. Another program addresses the problem of inadequate decoding skills that poor readers often have. These and similar programs do not teach, but offer practice for the purpose of remediation. They are not intended for all students, but only those deficient in a particular skill.

When a program of a remedial nature is to be evaluated, it may not be practical or even ethical to deny treatment to a subgroup of learners for the purpose of creating a comparison group. In such a case, it may be possible to employ the regression discontinuity design as a means by which to compare the treatment group with a nonequivalent group of learners. The design is presented graphically in Figure 2.

All subjects are administered a pretest related to the skill being taught (e.g., reading decoding skills). A cutoff point is selected—all learners below this cutoff score are chosen to undergo treatment by working with the program, while those who score above this cutoff score do not receive treatment. After the period of instruction, a posttest is administered to all groups, and a separate regres-

sion equation of posttest scores on pretest scores is computed for each group. The difference in the predicted scores at treatment and the cutoff point for the non-treatment group is considered to be evidence of the treatment effect for the remedial students.

Admittedly, the example presented here is not without difficulties. Using this design would probably preclude the administration of any other remediation in order to assess the software's effects. However, it is doubtful (at least in the near future) that schools would provide no other remediation for poor readers besides the use of a computer program. It would also require a large number of subjects (Cook and Campbell, 1979). Nevertheless, this design may have some plausible applications to similar software evaluation problems.

Conclusion

This discussion of methods is not exhaustive. Other designs that may be suitable to the evaluation of various types of software have not been mentioned. Furthermore, other student outcomes mentioned by Coburn *et al.* (1982), such as ease of use, interest to students, and student enjoyment, have not been addressed. For these areas, rating scales, questionnaires, observation methods, and other formal evaluation techniques are in order. □

References

Borg, W.R., and Gall, M.D. *Educational Research—An Introduction* (2nd ed.). New York: Longman, 1971.

Campbell, D.T., and Stanley, J.C. Experimental and Quasi-experimental Designs for Research on Teaching. In N.L. Gage (Ed.), *Handbook for Research on Teaching.* Chicago: Rand McNally, 1963.

Coburn, P., Kelman, P., Roberts, N., Snyder, T., Watt, D., and Weiner, C. *Practical Guide to Computers in Education.* Reading, MA: Addison-Wesley Publishing Co., 1982.

Cook, T.D., and Campbell, D.T. *Quasi-Experimentation: Design and Analysis Issues for Field Settings.* Chicago: Rand McNally, 1979.

Daellenbach, L.A., Schoenberger, R.E., and Wehrs, W.E. *An Evaluation of the Cognitive and Affective Performance of an Integrated Set of CAI Materials in the Principles of Macroeconomics: Studies in Economic Education, No. 4.* La Crosse, WI: University of Wisconsin, 1977. (ERIC Document Reproduction Service No. ED 150 057.)

Evans, R.W. *Designing Computer-Based Education for Effective Information Retrieval: A Cognitive Science Approach,* 1982. (ERIC Document Reproduction Service No. ED 222 173.)

Fredriksen, J., Warren, B., Gillote, H., and Weaver, P. The Name of the Game Is Literacy. *Classroom Computer News,* 1982, *2*(5), 23-27.

Heck, W.P., Johnson, J., and Kansky, R.J. *Guidelines for Evaluating Computerized Instructional Materials.* Reston, VA: National Council of Teachers of Mathematics, 1981.

Kandaswamy, S. Evaluation of Instructional Materials: A Synthesis of Models and Methods. *Educational Technology,* 1980, *20*(6), 19-26.

Suppes, P., and Morningstar, M. *Evaluation of Three Computer-Instruction Programs.* Stanford, CA: Stanford University, 1969. (ERIC Document Reproduction Service No. ED 031 408.)

Wolf, R.M. *Evaluation in Education.* New York: Praeger, 1979.