Computer Center

Computer Software Evaluation: Who, When, Where, Why?

Theodore J. Crovello
Department Editor

Theodore J. Crovello is Professor and Chairman of the **Biology Department at The University of Notre Dame, Notre Dame, Indiana 46556.** After graduating from the State University College of Forestry at Syracuse, New York, he received his Ph.D. in Botany at The University of California, Berkeley. Dr. Crovello has long-term interests in the way plants satisfy people's physical and non-physical needs, in the geography of plants, and in the use of computers in biology.

For use in The Computer Center, Ted welcomes suggestions on what subjects should and should not be treated, summaries of educational computing centers, innovative uses of computers, and information about relevant books and events.

Along with people and computer hardware, software is an essential component of any computing system. Software refers primarily to the programs—the set of instructions that activate computer circuitry and hardware. In one sense, it is meaningless to claim that evaluation of software is more important than hardware, because both are essential components. Yet in another sense, software evaluation *is* more important because students usually are more affected by the style, etc. of the interactive session which a program creates, than by the hardware used. Furthermore, sitting before a microcomputer at home or at school is becoming commonplace for students, just like their turning on a television. But it is the program (on television or the computer) which will turn *them* on . . . or off!

Good educational software requires considerable knowledge and effort to develop. Unfortunately, many programs and program packages currently advertized for educational use are not very stimulating. This presents a double danger and a mandate. The first danger is that if an educator purchases a program and on a trial run does not like it, he may discard it and also discard his feelings that the wise use of computers can enhance education. The second danger can occur if the educator decides actually to assign students a poor program because he bought it and feels he has to make use of it. This will turn students against computers in education. The mandate to us as professional educators is to resolve to evaluate a program thoroughly *before* we decide to use it. The following paragraphs answer the who, when, where, and why of evaluation. I'll save the how to evaluate for a later time!

Who Evaluates Software?

Software evaluation is a multistage decision process. So, as with hardware evaluation, it is not a simple, one-step process that always can be done by one person, such as an educator. Yet the teacher or professor in the course is the major evaluator, not only because he is most familiar with the pedagogic goals the Computers In Biological Education (CIBE; pronounced "cyber") module is to serve, but also because he participates in most if not all of the evaluation stages and is best able to make sound recommendations about the value of a given program in his course.

Students are important software evaluators, and at several points. They rarely are included in the earliest stages, i.e., before a specific program is created or purchased. But they should be. They can make an important contribution, especially if this is to be their first use of computers in the course or program. All or some of the students should be asked what they think of computers and if they have actually used them. Do they see computers as their replacement and competitor or as an extension of their minds? A short session on this topic often alerts the educator to the degree of development in the affective domain that may be required before students can learn with computers. At this early stage of software evaluation, students also can indicate what parts of a course or segment could really use computer assistance. Remember, they are the people we are trying to educate. A CIBE module is not a success because we educators like it. It is successful only if our *students* benefit from it. And they can suggest needs, and even interactive strategies that may prove most valuable.

The more common role of students as software evaluators is by actual use of the program, followed either by their own assessment or by the assessment of the program by others in terms of changes in their grasp of the subject matter, etc. This will be discussed in the next section, but here we emphasize the possibility of a dual role of the student: as evaluator and as evaluatee! As such, the possibility of a confounding interaction effect must be considered. That is, students who have been asked to evaluate a CIBE program via their direct assessment may as a result develop a much deeper grasp of the program's subject matter than students who are simply using the program as a normal part of their course.

Students should be involved in program evaluation as much as possible to increase their feeling of accomplishment. They can feel they are contributing to the educational process, not just taking from it. Students take genuine, increased interest in education when they are asked for their valued opinions, and computers can provide the opportunity. The result is often a better grasp of the subject matter and better performance.

Other potential evaluators include other teachers in the department as well as colleagues from other institutions who teach the same subject. If the programs are in wide use (or are being offered by a publishing company or other vendor who hopes they will be!), then a published evaluation may have appeared in journals that regularly include CIBE. In the life sciences, COMPEACE's ™ register of computer programs may contain additional information on the program being evaluated as well as information on similar programs of which a particular instructor may not be aware. The COMPEACE ™ register will be described in a later contribution to this column.

Administrators, like department chairpersons and their superiors, usually have to approve program purchases. But such approval may come almost automatically since the cost of one CIBE program is usually less than $50. However, if every year an administrator were to receive requests for two programs from each of 10 or 20 faculty, he may ask, and should ask, that they present him with a well-thought-out and long-range plan for all software purchases. Naturally such plans will be more acceptable if they include suggestions as to how at least partial funding may be obtained from outside sources.

When Is Software Evaluated?

Software obviously is evaluated before purchase. But at least, informal evaluation should be an ongoing process after its purchase. The same is true for programs created by the educator, his organization's support staff, or by students themselves.

A particular CIBE program can be compared to a biological species (not just an individual organism). A program has a birth, and a period of time during which it is reproduced (we hope only legitimately!). It undergoes changes that sometimes result in the addition of new features or the deletion of old ones. Finally it dies when no one anywhere uses it. A given CIBE program may have begun its existence in a niche such as in a General Biology course. But as the program, the course, or perhaps both, changed, the program no longer may have seemed valuable enough and was extirpated from that course. Perhaps this first occurred at some locations, and then in all places where it had been used in such a course. Perhaps it was outcompeted by a similar but better program. Yet, perhaps the original program later seemed appropriate for a different course, either at a level below where it originally appeared, above it, or maybe both. In the last case, we can expect that any small changes needed to be useful at the lower level would be incompatible with those of the higher level. The result would likely be the rapid extinction of the original program and its replacement by two derivative programs, each more specialized than their common previous version. This process of modification of a program during its use within one type of course, its competition with other programs, of adaptiveness (with or without modification) to other courses, and its separation into two distinct programs, might be repeated several or many times. But eventually the CIBE program and its derivatives will probably no longer be selected for any course anywhere, and the program will die. Some copies of it may be preserved for posterity in dead program files.

My reason for presenting the above analogy is not primarily for entertainment. I believe that considering a CIBE program as possessing properties and environments similar to a biological species forces us to take both short- and long-range perspectives in our program evaluations. It also suggests at which stages deliberate evaluations of a CIBE program should be made. Regardless of the extent of the prepurchase evaluation, the program must be monitored closely during its first real use in a particular educator's course. This will assure immediate correction of any large-scale problems and incompatibilities, and also provide necessary specifics to carry out any "fine tuning" of the program to function effectively in the educational context of a given educator's course. Seeds of some plant species sometimes land on poor soil and germinate, but quickly die. So too a CIBE program may have seemed appropriate for a given course, but during the first semester of use, it became clear that it was hindering education rather than helping it. The professional educator must admit the fact and discontinue the program's use. This can be very difficult if the educator actually wrote the program! On the other hand, if some relatively slight modification of the program or of the way in which it integrates with lectures, etc. can improve the educational value of the CIBE module, it should be tried.

The end of the first semester is also an important evaluation point. A teacher can determine at that time if use of the program significantly increased achievement of the goals of the course by all or an identifiable segment of the class. If only a particular group of students showed an improvement, then a decision should be made about whether to provide the program in future semesters only to that group.

How does one determine if the CIBE use has significantly increased achievement? The seldom-met ideal would be to divide the class at random into two parts (possibly based on a pretest) and then to administer the "treatment" of the program to only one group. All other important factors would be invariant. Results of one or several tests given to both groups could then be analyzed by a t-test or a more involved analysis of variance design. If the computer-using group did significantly better, this might be reason enough to continue the program's use. But beware of the novelty effect on test performance! That is why significant positive differences during the first CIBE use must always be accepted cautiously, and await confirmation by several years of users. But each year's group of students again will be exposed to a novelty effect! Its magnitude will be reduced as more students bring more computer experience to the classroom, either from home or from previous courses.

Documentation of significant positive effects becomes more difficult if the class cannot be divided into a control and a treatment group, or if statistical expertise is unavailable. Verification is also more difficult if the characteristics used to measure improvement are less easy to quantify, or take too long. Examples include students' attitudes toward the subject and knowledge and concept retention one or two years after completion of the course. In such cases, teachers must rely on their professional experience with student reactions and test performance. For example, if the class cannot be divided into those with access to the CIBE program and those without, then let all students have access. Compare this year's "treatment" group with last year's "control" group that did not use the CIBE program. The resulting statistical analyses might be somewhat less reliable in our estimate of the effect of the CIBE program because it will confound any program effect with any effect due to differences between the two years of students, or between the way the teacher taught the course. This latter includes the teacher's change of enthusiasm (up or down) due to the presence of a computer in the classroom. While the possible confounding effect might be reduced somewhat by methods that consider average IQ, pretest scores, etc. of the students in the two years, such methods would mostly just increase the statistical power of the test. To put it another way, if these additional methods are *not* used *and* a significant difference *is* observed, you can be confident the significance is *not* due to the effects that were thought to be confounding the test.

After the first year of actual use, observations should be made every time the course is given, although perhaps not as formally. Teachers must be alert to a progressively decreasing interest in the CIBE module. This can be due to a combination of any of the following or other factors: decreased teacher enthusiasm about it; decreased emphasis on the topic to which the CIBE program is relevant; advances in the topic which make the program somewhat obsolete; advances in computers or CIBE which make the program less effective than it could be; and advances in students' computer awareness and computer literacy which make them dissatisfied with the program.

The decreased value of a CIBE program for most of the above reasons can be corrected. For example, the existing program can be modified, or a better one purchased. Whether either action is justified depends on consideration of the resulting cost/benefit ratio, which requires details of a particular situation for its estimation. Such judgments may have to rest on informal feelings rather than quantitative analysis. Regardless of the decision, the most important rule is: *never use a program that is past its useful time*!

Where Is Software Evaluated?

Educators can make preliminary evaluations of software anywhere, but major evaluations must take place in the regular classroom or computer laboratory. In other words, the most reliable evaluations are those performed in the educational context in which students will be using the computer. To do otherwise would confound any conclusions drawn from the evaulation data. So if higher space priorities in a department or school force the computers to be housed in an out-of-the-way building annex, that is where the evaluations should take place. Remember that no CIBE program can be effectively evaluated in a vacuum. As educators we can construct in our minds a scenario about how valuable a program may or may not be in our class, but ultimate evaluation of selected programs must be carried out in the normal settings of one's course.

Why Evaluate Software?

As mentioned at the outset, software is evaluated because it is an essential component of any computer system and in some ways it requires closer evaluation than computer hardware. Why? First, students (and sometimes teachers!) have little choice in the selection of hardware. But more importantly, most microcomputers usually have the same essential parts: a keyboard, a video display, etc. As a result, the hardware at this point is more or less a constant to students. After the first few minutes they will take it for granted, and possibly give it little conscious thought. In contrast, the CIBE program is the very reason they are using the computer. More importantly, the software sets the tone of the computer session. It conveys the personality of the computer for that assignment. This occurs through a combination of many factors, including the program's interactive strategy; the psychological tone of the program as evidenced by its "user friendliness"; the content and phrasing of its replies to a user's incorrect or correct choices; and the program's use of color and graphics.

The status of these and many other software traits determine whether it will have a positive educational effect, just as a teacher's personality and attitude toward students determines his effectiveness. Related to this is a totally unexplored question: Should the "personality" of a CIBE program be similar to that of the teacher, should it be the opposite, should it have no personality (an impossibility), or should it be neutral or low-key? A rigorous answer must await future research, but a useful guideline for the present would be to create or use CIBE programs that resemble an ideal teacher—one who is intelligent, friendly but firm, possesses a positive mental attitude, believes in positive reenforcement, and who knows the type of give-and-take banter appropriate for different grade levels. This may seem like a tall order for a computer program. But the field of artifical intelligence applied to computer dialogues promises just this!

Publications to Help You Evaluate Educomputing Hardware and Software

CROVELLO, T.J. 1982. *Educational computing systems: The COMPEACE ™ guide to evaluation of their hardware, software, and people.* Available from Computers For Education, Justice, and Peace, Box 554, Notre Dame, IN 46556. $10.00. 61 p. (Approaches evaluation of educational computing systems from several points of view, but especially from the perspective of the individual teacher or department head. Includes characteristics of hardware and software of special value to educational computing, suggests how teachers can develop valuable approaches to educomputing, etc. Written in the style of Crovello's *ABT* Computer Center department.)

HECK, W.P., JOHNSON, J., and KANSKY, R.J. 1981. *Guidelines for evaluating computerized instructional materials*. National Council of Teachers of Mathematics, 1906 Association Drive, Reston, VA 22091. $3.75. 30 p. (A very readable, informative introduction to evaluation of educational software.)

[Many Authors]. 1981. Special Issue—Microcomputers. *The Mathematics Teacher* 74(8). Available from National Council of Teachers of Mathe-

(Continued on p. 446)

enterprise conclusions neither based on scientific data nor verified by the scientific process. These conclusions, arising outside the field of science and resulting from ignoring or misinterpreting scientific data, have no place in the science classroom as a part of the body of scientific knowledge.

The NABT, through its obligation to biological education, will make every effort to educate the public as to the unscientific nature of efforts to equate non-science with the scientific enterprise. NABT will resist attempts to place non-scientific dogma into the classroom as science. Wherever such efforts are attempted, NABT should correct the record and provide adequate scientific evidence designed to allow decision-makers full access to the facts by means of which to judge the efforts to intercalate non-scientific material into science classrooms or to remove or change the data of science to accommodate a given set of conclusions derived from outside the scientific enterprise.

The credibility and usability of science depends upon maintenance of the integrity of science as a discipline. While no feature in this policy is to be construed as preventing the full range of applications of science and the elucidation of its social and humanistic implications, there is an obligation to insure that the scientific data thus used is both accurate and derived within the accepted procedures of the discipline. Without the maintenance of the integrity of the initial data with which one works, any subsequent applications or derivations may be ill-conceived and of little service to the human enterprise.

NABT has an obligation to maintain the integrity of biology as a scientific discipline. To this end it must act to resist efforts to include in science classrooms materials derived outside the scientific process. It must insist that the data and concepts of science as presented to students meet the accepted standards of the discipline, and data which can best be described as para-scientific (creationism, astrology, anti-germ theory, etc.) cannot be condoned as science within classrooms.

(Adopted by the Executive Committee Oct. 23, 1980)

Publications and Teaching Materials

Annual Science and Technology Report to the Congress, 1981, published by the Office of Science and Technology Policy in cooperation with the National Science Foundation, is now available from the Superintendent of Documents, Government Printing Office, Washington, DC 20402. Price is $6.50. Request stock number 038-000-0504-9.

A manual describing how University/Industry cooperative research centers work and how to get one started has been published by the National Science Foundation's Division of Industrial Science and Technological Innovation. The *Practice Manual* is available free from Louis G. Tournazky, Division of Industrial Science and Technological Innovation, National Science Foundation, 1800 G. Street, N.W., Washington, DC 20550.

Software Evaluation

. . . from p. 433

matics, 1906 Association Drive, Reston, VA 22091. $3.50. About 85p. (Contains 12-15 short articles on computers in education. Some are oriented toward precollege mathematics, but many of the points made are applicable to other disciplines.)

[Unknown Authors]. 1981. *Using microcomputers to teach science.* Cambridge Development Laboratory, 36 Pleasant Street, Watertown, MA 02172. $10.00 per booklet. 50 p. per booklet. (Crovello has not seen these, but the publisher provides this information: "One of the 50 page booklets gives a general overview of the topic, including hardware, software, and sources of further information. The other focuses on the various ways of using microcomputers to monitor and control laboratory experiments. These booklets will help both high school and college science instructors make more informed decisions on what hardware and software to buy.")

Human Sciences Program

. . . from p. 428

ignorance. Can we stem this tide? Will we make our voices heard in order that the fresh air of openness may again reach the minds of our offspring? We have no choice. We must.

References

SNOW, C.P. 1961. *The two cultures and the scientific revolution.* New York: Cambridge University Press.