

Integrating a computer component into the student psychology laboratory: Problems and prospects

DOUGLAS B. EAMON

University of Wisconsin—Whitewater, Whitewater, Wisconsin

This paper presents some commonly occurring problems associated with integrating a computer component into student psychology laboratories. Implementation of a computer component requires consideration of differences among students, objectives and teaching style of the instructor, course level and content, and courseware to be used. These factors are examined with respect to how they can affect the success of an attempt to use computers in the instructional laboratory.

The proliferation of software designed for student psychology laboratories and the wide exposure given the use of microcomputers in the teaching laboratory appear to have fostered several seldom stated, but widely believed, false impressions: that (1) nearly everyone is using Computer Aided Instruction (CAI) in student laboratories, and that (2) everyone who is using CAI is doing it successfully, because (3) programs exist to do nearly everything one might wish in a laboratory course, and (4) the programs can do whatever we expect them to do (only better, and with less work, than more traditional approaches). The purpose of this paper is to examine these and some related illusions by describing some commonly occurring problems which may arise in using computers in the student psychology laboratory.

STUDENT, INSTRUCTOR, COURSE, AND COURSEWARE

Successful use of computer-based materials involves integration of a variety of components. Included among these are the student and his/her level of knowledge, motivation, and needs; the instructor and his/her teaching style and ability, relationship with students, and computer sophistication; the course and its objectives and content, size, available facilities, and level and role in the curriculum; and the courseware and its availability, level of difficulty, and suitability for accomplishing the purposes as defined by the other three items. The first three of these factors interact in various ways. In this paper, I am concerned primarily with their interactions with courseware.

The Student

Students enrolled in methods laboratories are often fearful of the course from the outset, particularly if the course

is taught in conjunction with statistics. For many students, rumors of the use of computers compound this initial fear. The presumed greater computer sophistication of newly entering college students does not appear to be manifested in either expertise or even comfortableness with computer applications in the psychology laboratory. Although video games and personal microcomputers may be widely available to college and precollege students, laboratory applications require a sophistication not enhanced by these experiences. In particular, the computer in the laboratory is a research tool; its appropriate use thus requires not only the ability to turn on the machine and load and run programs, but knowledge of how to take advantage of the software in a particular context with a specific objective.

In this context, two problems related to student computer sophistication emerge: (1) students who have had little contact with computers may fear them; and (2) other students may wrongly feel that they already know a great deal about using them, and exhibit a false confidence.

"Computer phobia," as it is often called by students, may be easier to deal with if others in the class share the fear. In general, such fears are based on the expectation not that the computer will do something wrong, but that the student will, and that unknown consequences might follow. Introduction and demonstration of the programs by the instructor on a large-screen CRT or projector is very effective in relieving these fears, particularly if the instructor/demonstrator deliberately makes a variety of errors (inserting a disk upside down, inserting the wrong disk, selecting the wrong item from a menu, entering inappropriate responses to prompts, etc.). An ungraded assignment simply to sign onto and sign off the computer (if time-sharing on a main frame), or to run a program which provides instructions on using the computer, can also be helpful. The program *Apple Presents Apple* (Eisenberg & Tognazzini, 1982), although somewhat long, is well suited to this purpose.

Game programs may serve as icebreakers and provide an introduction to some important concepts about methodology. For example, hypothesis testing can be explored

Reprints of the paper may be obtained from Douglas B. Eamon, Department of Psychology, University of Wisconsin—Whitewater, Whitewater, WI 53190.

using the program *Lemonade* (Jamison, 1979). The *Lemonade* game is an engaging program in which the student plays the role of owner/operator of a lemonade stand. He/she manipulates variables including the cost per glass, number of glasses made, and the number of advertising signs made. These variables interact with a random weather variable to produce the dependent measures number of glasses sold and profit over a period of days. Spivey (1983) and others at the University of Kentucky have used this program as an introduction to scientific observation and experimental design; a short work sheet provides space for recording variables and asks the student to identify the independent variables and dependent variables, their levels of measurement, and so forth. More sophisticated game programs that may also serve this purpose are *Baffles* (Spain, 1983) and *Tribbles* (Von Blum & Hursh, 1976). In all of these cases, the objective is for the fearful student to run a nonthreatening program which will not crash unexpectedly.

Students familiar with computers from previous experience, student "sophisticates," may be overconfident. They are also likely to have seen and run far more exciting and interesting programs than those usually available for instructional purposes. Fast-paced action games or elaborate adventures have been a staple of their computer diet; these students often become bored with instructional material. In some cases, they may have had considerable programming experience and know several programming languages. I have tried to identify these students early and have asked them to serve as special assistants for the other members of the class. This approach has had mixed success: in some cases, the sophisticated students spent much of their time playing games on the computers, often with the students they were supposed to be assisting, and at the expense of doing the assignments related to the course. Other sophisticates spent their time copying (or attempting to copy) various programs available in the computer lab, again at the expense of helping their colleagues. And it is not unusual to discover that the sophisticates are not as sophisticated as they initially appear; one such individual had the unusual idea that all diskettes had to be formatted (initialized) before use—the result was that six or seven copies of programs assigned for class use were destroyed. The lesson here is that the instructor must know a good deal about the level of expertise of these students before allowing them to play a special role, and then make clear the nature of the role and what is expected.

A particularly difficult situation sometimes emerges when a student becomes extraordinarily interested in the computer and the programs, independent of their use in the course. The student "hacker" is distinguished from the sophisticate largely by his/her degree of enthusiasm, not by the amount of experience or expertise shown. These students often spend immense amounts of time at the computer, while failing to accomplish class and laboratory assignments.

As for the sophisticate, it is tempting to assign the hacker to some special projects or to allow some special

privileges; their enthusiasm is contagious and they are always available and willing to help. Unfortunately, such an approach is rarely successful: as Weizenbaum (1976) observed, the hacker rarely finishes any project, no matter how simple. This is especially true if the project is an alternative to or substitute for course work assigned to others in the class. The hacker, like the sophisticate, may often take advantage of special privileges and assignments as a way of avoiding class-related assignments; thus, any projects assigned should be clearly specified and considered only as extra work, rather than as substitutes for or alternatives to regular assignments.

The Instructor

Although many, if not most, teachers of laboratory courses may have had some experience in using computers, few can claim to be experts. Implementation of a computer-based laboratory does not require expertise in computer applications, but it does require familiarity with the hardware and software to be used.

An effective procedure for becoming familiar with both the hardware and software is to run the programs under two conditions: first, following instructions as closely as possible, and second, making every possible error (pressing the wrong key, selecting inappropriate continuations, failing to connect peripherals, etc.). This last approach is commonly used in debugging programs, and will give the instructor information on the sorts of error messages (if any) that the program generates, as well as likely sources of student misunderstandings.

Instructors vary in the degree of control they exercise over student projects. Some prefer to select the first one or two projects, and only later allow the students to independently design and run their own experiments. Even then, constraints are often placed on the types of experiments that are acceptable (e.g., it must be a factorial design).

Computer-based laboratories tend to encourage greater student independence in experiment design and selection. Often a data generator is used which itself contains several "modules" or experiment scenarios. Each module allows the student to select his/her own independent variable and dependent variable, and to control other variables which may (or may not) affect the outcome of the experiment. Artificial data are then generated which resemble what might happen if the experiment were actually run as described.

The flexibility of the student in making these selections has been widely regarded as a virtue (cf. Eamon, 1980; Main, 1978), based on the assumption that students able to select modules and manipulate variables of interest would become more involved in and motivated about the projects. This is not always the case. Students frequently select modules based not on personal interest, but on such factors as their estimation of the amount of effort needed to prepare the report for submission, the number of students in the class choosing the module, judgments of the difficulty of finding relevant literature, or, generally, their

estimations about how hard or easy it would be to complete the assignment.

In contrast, instructor-designed experiments, even if the same for each student enrolled, may be far more valuable in conveying the concepts and procedures intended by the exercises. Furthermore, discussion of problems involved in the design and outcomes of experiments can be greatly facilitated if all of the students run the same experiment. This is particularly true in introductory-level methodology classes, in which a clear-cut example can serve as a model for learning the concepts. In the computer-based laboratory, then, the instructor may prefer to assign the same experiment to all students, at least for a first attempt.

The Course

Most institutions provide psychology laboratory instruction in two formats: a methods laboratory and several content laboratories. Although they often share the same physical space, courses in these labs differ in both objectives and level.

The methods laboratory is frequently a freshman- or sophomore-level course, and has as its primary goal instruction in issues and techniques of scientific research. The philosophical foundations of science, goals and types of research, various designs and statistical analyses appropriate to them, interpretation of results, and reading and writing of research reports are commonly covered topics. Laboratory projects in this setting are frequently used to introduce the student to a particular issue or technique by way of example.

Content laboratories are often associated with advanced courses which require as a prerequisite the lower-level methods laboratory. In content laboratories, a primary aim is to teach substantive content about an area (e.g., cognitive psychology, physiological psychology, social psychology, etc.). Selection and use of laboratory projects thus tend to focus around the particular subject matter rather than the technique employed to investigate it.

Several types of applications or strategies (Castellan, 1983) for CAI may be implemented in either of these types of laboratories. These include demonstrations and tutorials, games, data generators, and experiment simulators. To these might be added a fifth type of application particularly suitable for the student instructional laboratory: experiment generators (Eamon & Butler, 1985).

One of the most pressing concerns in implementing a computer component is determining the extent to which the course can (or should) be computerized. That is, what is an appropriate balance or mix between the computer component and a traditional component?

Many instructors who make extensive use of computers in laboratory courses require students to run at least one project in which real, as opposed to simulated, subjects are used. The process of designing and planning the experiment, obtaining subjects and interacting with them, developing suitable instructions and stimuli, locating and

maintaining traditional laboratory equipment, and so forth all provide important lessons about the research process and the frustrations which often accompany it. Preservation of these aspects of a course, as well as the likelihood of unexpected complications surrounding the implementation of computer-based materials need to be considered. For example, there may not be enough computer time or facilities available for students to run the programs within the expected time: Some programs may take unexpectedly long times or widely varying times to run, and data can be lost through disk, printer, or program failures. Computer-based modules, then, should be introduced slowly and cautiously, with careful classroom testing at each step, and with generally greater flexibility in due dates for computer-related assignments and projects.

Nonetheless, implementation of a fully computer-based course provides some unexpected opportunities not otherwise available. One such opportunity involves production of student "journals." In one course, students ran computer simulations and wrote empirical or theoretical articles regarding their findings. A component of the course grade depended upon the number of publications in these journals and the number of other students who cited these publications in their own papers. It is unlikely that such a system could be implemented in other than a fully computer-based laboratory.

Implementation of a computer-based component will necessitate several changes in a laboratory course. One of the first changes many instructors make is to increase the number of laboratory projects required. Particularly if simulations of experiments using data generators are used, students find it relatively easy to run experiments and obtain data for analysis and interpretation. The pedagogical value of increasing the number of reports is questionable, however: more does not necessarily make better. An alternative, which also takes advantage of the ease of generating data using the computer, is to alter the nature of the projects by specifying that they be conducted as programmatic research. Virtually all data generators provide the opportunity to manipulate a variety of potential independent variables and control other variables at various levels. Rather than doing a series of single experiments, students might conduct and report a number of related experiments to be reported in a single paper. This approach more closely resembles published research in major journals, and provides the opportunity for the student to understand the relationships among a set of results.

A second addition might be the inclusion of a section on statistical analyses using the computer. Programs for statistical analysis and considerations involved in integrating them into the student laboratory are discussed elsewhere (Butler, 1986; Butler & Eamon, 1985) and will not be examined here.

Self-designed projects using computer programs represent another modification some instructors choose. Many students who have used computer programs to simulate well-known experiments recognize their potential in presenting stimuli and collecting and analyzing data. If

the option (or requirement) of running an experiment of their own design is available, these students often desire to implement their experiment on the computer. Unfortunately, students rarely have the expertise to write or modify existing programs to suit their needs, and instructors almost never have the time. "Experiment generator" software exists for such purposes (e.g., Digitry Company, Inc., 1981; Dlhopsky, 1984; Eamon, 1982; see Eamon & Butler, 1985 for a list of these programs) and can be used to advantage in these cases. Unless an experiment generator is available, however, such projects should be discouraged. Even simple modifications of existing programs can turn out to be very complex and time-consuming; students are likely to spend much of their time in this process rather than running subjects and collecting data.

Computer-based bibliographic searches are also good to include in these courses, if the library can provide these services.

The Courseware

Three types of courseware are of particular interest in the student laboratory: data generators, experiment simulators, and experiment generators. Each presents special problems.

Data generators. Programs which produce simulated data based on student-entered experiment descriptions are in widespread use in instructional laboratories. A number of characteristics of these programs make them desirable and usable from the instructor's standpoint: ease of use, the ability to quickly generate a number of experiments under varying experimental conditions, the ability to rerun the same experiment several times, and so forth.

In addition to the problem of the students' reasons for selecting experiments to be simulated (mentioned above), these programs may create other difficulties which should be carefully examined by the instructor contemplating their use.

If the American Psychological Association (APA) style of written experimental report is required, students nearly always have to fabricate portions of the report to meet acceptable completeness requirements. I know of no program in which the written descriptions of the experimental modules available from the program distributors provide sufficient information to allow the student writer to describe the subject selection procedure, assignment to conditions, subject ages, sexes, information regarding manufacturer and model number of the (simulated) equipment used, names (and sources) of tests administered, and, in many cases, a clear description of the experimental procedures used. The effect of this need to fabricate information in such cases is uncertain, but one disturbing possibility is that students using these programs might develop an unhealthy attitude toward research: "Make up what you need; it's just a game, anyway," or "Who cares what you say? Nobody will know the difference."

This problem is even further exacerbated by the nature of the data produced, which are, of course, simulated (or, in the views of some students, "faked"). Limiting the

use of data generators only to collecting numbers for statistical analysis (with no written reports) undermines their strengths and defeats their purpose. A primary reason for using these programs is to provide an opportunity for students to test hypotheses without the necessity of acquiring elaborate equipment or unusual subject samples (schizophrenics, etc.).

A third problem appears when students have successfully used the programs once or twice. Discovering how easy it is to perform simulated experiments, students often run many versions of the same experiment, or minor modifications of it, hoping to find a combination of independent and dependent variables which produce a statistically significant result. Having experienced success in this "effort," little attention is later paid to the crucial planning stages of experimentation, but considerable time is spent at the computer running many experiments hoping to find one that "works."

Fourth, even (or perhaps only) the most dedicated students discover in a literature search that their carefully planned "original" experiment has been performed and reported many times. This can be discouraging, to say the least, particularly given the usual emphasis in these courses on the need to generate and test original hypotheses and to add new information to the existing body of scientific knowledge.

This problem is compounded by the limitations built into the programs themselves. The programs are usually reasonably faithful representations of what is already known about a subject area. If a student proposes a truly original hypothesis involving manipulation of some independent variable not provided on the list of available IVs, or some procedural variation designed to examine an original hypothesis, the programs can be more frustrating than instructive.

Experiment simulators. Experiment simulators are programs in which the computer acts as a data acquisition device to allow students to run actual subjects (often themselves) in classic or well-known studies. The focus of these projects is on the data collected and its theoretical interpretations.

Traditional laboratory courses often require students to record data on data sheets. These are then brought to class, collected, rerecorded onto a single master form, and distributed to students. Some programs for student laboratory courses provide automatic storage of data on disk. Ideally, these data can be combined into a single data file and made available to each class member for analysis and interpretation. Unfortunately, this ideal remains largely an illusion for current software. For example, the widely used *Laboratory in Cognition and Perception* (Levy, Fischler, & Griggs, 1979) does not store data on disk at all, and statistical routines are not provided; all data collected must be copied from the screen or printed and analyzed in the traditional manner. Many of the *Computer Programs for Experimental Psychology* (Perera, 1979) provided excellent and appropriate statistical analysis routines along with the programs themselves, but no provi-

sions are made for collecting data which have been stored on several diskettes. Programs in the *Computer Lab in Memory and Cognition* (Keenan & Keller, 1982) allow data to be collected into a large summary data file (even if data were initially stored on several diskettes), but the files cannot be read by general statistical packages because they include nonnumeric information and lack of consistency in data file formatting. Most other programs have similar problems. Thus, one of the most touted virtues of the computerized lab, ease of data collection and analyses, is generally not available, especially if a mix of programs is used.

Students in content labs are often required to replicate classical research reported in the literature. Interestingly, many of the computer implementations are modifications of original research conducted without the aid of computers (for example, using tachistoscopes). Even experiments done originally with computers are frequently modified for a variety of hardware and programming reasons. These differences raise questions about the degree of fidelity to be expected for computer-based experiment simulations as compared to the original research to be replicated. If the intention of the instructor is to provide students with a nearly exact replication of the original classic experiments, computer-based materials may be inappropriate. In addition, computer replications of experiments may yield results noticeably different from those obtained in the original research. These differences are likely to be detected by students, who, as a result, may become skeptical of the entire research process.

Computers are often said to allow control of many experimenter and extraneous variables which might otherwise affect the results of an experiment. But situations are likely to arise in which an experiment simulation using a computer consistently fails to replicate results obtained using noncomputer procedures. Similarly, there may arise cases in which experiments originally done using computers may fail to replicate if done using traditional lab equipment.

Experiment generators. Experiment generators are typically used when the requirements of the course include an individual project. They are designed to allow the student to set up and run experiments of his/her own design on the computer. Experiment generators generally require the user to enter instructions to be given to subjects, stimuli, information specifying the condition for stimuli presentation, and acceptable responses. Data are then collected from actual subjects and stored to disk for later recovery and analysis.

More than any other type of application, experiment generators require some degree of computer sophistication on the part of the user. Frequently, special hardware (e.g., clocks, special input/output devices for collecting responses or controlling other peripherals) is required. Compared to other types of applications, therefore, these programs can be expected to take longer for students to understand and use effectively. For the same reason, instructors should expect to spend considerably more time

assisting students in their efforts. Also, these programs are accompanied by long and often complicated user's guides; it is not uncommon that the student (and instructor) will take longer to read and understand these guides than would be needed to set up the same experiment using standard or traditional laboratory equipment.

In spite of the apparent generality of many of these programs, most have limitations which make them inappropriate for many applications. For example, virtually all are entirely text oriented; creation and display of graphics stimuli is difficult or impossible. None provides complete statistical analysis routines for data collected, and in most cases the data files created cannot be read by commonly available statistical packages. Most assume that the user will want to create within-subjects experimental designs (between-subjects designs often require special techniques or are simply impossible to set up). Most have severe limitations on the size of the stimuli set which can be used (8000-12000 bytes is typical). Few have flexible or effective randomization procedures for stimuli presentation. Although some claim timing accuracy to 1 msec, this may vary considerably from trial to trial, especially if BASIC "PRINT" statements are used to present stimuli on screen (on the Apple II, for example, the time needed to write a stimulus to the screen varies with the number of characters in the display as well as with the memory location of the item to be displayed). Slow decay CRTs compound this problem.

Thus, while experiment generators can be used effectively in the student laboratory, the instructor should not expect that every student will be able to use them easily or appropriately.

SUMMARY

In this paper, I have attempted to describe a number of problems which might be encountered in integrating a computer component into a student laboratory course. Although the tone of these comments has been necessarily negative, it is clear that computer applications have the potential to enhance many such courses, and they will surely increase in the next years. Integration of student, instructor, course, and courseware in an effective computer based laboratory can be facilitated by recognition of and attention to these problems.

REFERENCES

- BUTLER, D. L. (1986). Integrating statistical software into laboratories and laboratory courses. *Behavior Research Methods, Instruments, & Computers*, *18*, 241-244.
- BUTLER, D. L., & EAMON, D. B. (1985). An evaluation of statistical software for research and instruction. *Behavior Research Methods, Instruments, & Computers*, *17*, 352-358.
- CASTELLAN, N. J. (1983). Strategies for instructional computing. *Behavior Research Methods & Instrumentation*, *15*, 270-279.
- DIGITRY COMPANY, INC. (1981). *Cognitive testing station* [Computer program]. Edgecomb, ME: Author.
- DLHOPOLSKY, J. G. (1984). *Software methods for experimental psychology* [Computer program]. Bayport, NY: Life Science Associates.

- EAMON, D. B. (1980). LABSIM: A data-driven program for instruction in research design and statistics. *Behavior Research Methods & Instrumentation*, **12**, 160-164.
- EAMON, D. B. (1982). CEDATS: A cognitive experimental design and testing system. *Behavior Research Methods & Instrumentation*, **14**, 142-145.
- EAMON, D. B., & BUTLER, D. L. (1985). Instructional programs for psychology: A review and analysis. *Behavior Research Methods, Instruments, & Computers*, **17**, 345-351.
- EISENBERG, J. D., & TOGNAZZINI, B. (1982). *Apple presents Apple* [Computer program]. Cupertino, CA: Apple Computer Co.
- JAMISON, B. (1979). *Lemonade* [Computer program]. Minneapolis, MN: Minnesota Educational Computing Consortium.
- KEENAN, J. M., & KELLER, R. A. (1982). *Computer lab in memory and cognition* [Computer program]. Iowa City, IA: CONDUIT.
- LEVY, C. M., FISCHLER, I. S., & GRIGGS, R. A. (1979). *Laboratory in cognition and perception* [Computer program]. Iowa City, IA: CONDUIT.
- MAIN, D. B. (1978). The implementation of EXPER SIM. *EXPER SIM readings*. Iowa City, IA: CONDUIT.
- PERERA, T. B. (1979). *Computer programs for experimental psychology* [Computer programs]. Bayport, NY: Life Science Associates.
- SPAIN, J. S. (1983). *Baffles* [Computer program]. Iowa City, IA: CONDUIT.
- SPIVEY, J. E. (1983). Software development for computer-assisted instruction in experimental psychology. *Behavior Research Methods & Instrumentation*, **15**, 183-186.
- VON BLUM, R., & HURSH, T. M. (1976). *Tribbles: An introduction to the scientific method* [Computer program]. Iowa City, IA: CONDUIT.
- WEIZENBAUM, J. (1976). *Computer power and human reason*. San Francisco, CA: Freeman.