SESSION VI GETTING STARTED WITH MICROCOMPUTERS IN UNDERGRADUATE EDUCATION

Walter vom Saal, *Presider*

Getting started with microcomputers in undergraduate education: Hints and guidelines

WALTER VOM SAAL

Millersville University, Millersville, Pennsylvania

DAVID A. ECKERMAN

University of North Carolina, Chapel Hill, North Carolina

PETER BALSAM

Barnard College, Columbia University, New York, New York

and

CYNTHIA McDANIEL

Northern Kentucky University, Highland Heights, Kentucky

Microcomputers can serve many functions in undergraduate education, including control of laboratory experiments, presentation of classroom demonstrations, generation of handouts, and monitoring student performance. Panel members of a symposium presented several general guidelines for the use of microcomputers, and specific guidelines for selecting and purchasing equipment and for daily operation of a microcomputer facility.

At the 13th annual meeting of the Society for Computers in Psychology, November 16, 1983, the authors participated in a symposium on Getting Started With Microcomputers in Undergraduate Education. A variety of uses of microcomputers in undergraduate education were described, including control of undergraduate laboratory experiments, presentation of classroom demonstrations, generation of handouts, and monitoring student performance. Walter vom Saal described the use of Atari microcomputers for student designed research in a research methods laboratory course; David Eckerman described laboratory control with Apple IIs connected to a hard disk; Peter Balsam described instrumenting TRS-80s for laboratories in animal learning, perception, human learning, and statistics; Cynthia McDaniel described a number of applications of microcomputers in an introductory psychology class.

The authors may be contacted at the Departments of Psychology of their respective institutions; zip codes are, respectively, 17551, 27514, 10027, and 41076.

We then suggested the following guidelines based on our experience using microcomputers in undergraduate education. It should be emphasized that this is not an exhaustive list, and the items are not listed in priority order. However, we did feel that these guidelines might be useful to others establishing or planning to establish a microcomputer facility.

GENERAL GUIDELINES

- (1) "If you want something done right, do it yourself." BUT:
- (2) Don't get hooked on computers.
- (3) Don't get hooked on programming.
- (4) Don't get hooked on the glamor of technology.
- (5) Don't assume your colleagues will be as excited about computers as you may be. Some will be enthusiastic, some neutral, some negative. Many will express positive views but do little because of lack of time, other commitments, or a general discomfort with computers.
- (6) If additional faculty involvement is desirable or

necessary, plan for ways to encourage it. Consider work sessions, seminars, minicourses given by you or others, shared projects, release time from courses or committee work, regular meetings with you, and funding for off-campus workshops. Try to plan to have one or more "loaner" units for interested faculty to take home.

(7) Seek help from others on campus. Computer science (CS) profs may help you select hardware and software; CS students may write programs for you as student projects; electronics majors may construct interfaces.

(8) Seek help outside campus. Read magazines and join users' groups. (Write Cyndi McDaniel at the address above for *Micropsych* newsletter.)

SELECTING AND PURCHASING EQUIPMENT

(9) Carefully define your exact needs before you buy anything. Specify details: speed, timing, input and output requirements, sample screen displays, graphics requirements, etc.

(10) Be sure to see (or speak with someone who has seen) both hardware and software before you buy.

(11) Try to locate a resource center where you can try out available software.

(12) Insist on seeing the documentation of programs you might purchase, as well as the hardware.

(13) Standardization within the department is desirable.

(14) Compatibility with other equipment across campus is desirable.

(15) Be aware that almost every program, commercial or home grown, has bugs in it. This includes built-in operating systems that are part of computer ROM. These bugs are often unadvertised! To find out about these, talk to other users and read computer magazines and newsletters.

(16) You may choose to write your own programs. However, don't expect computerization to save you time unless (a) software exists and (b) you've seen the software working.

(17) Be prepared to see better, less expensive systems appear just after you've ordered your system but before it has arrived.

(18) On the other hand, a system is not obsolete if it serves your purposes.

(19) Peripherals may cost far more than the microcomputers themselves.

(20) You may not be able to do everything with one system.

(21) Recognize that you may soon want to expand, and plan for that.

(22) For word processing, a disk drive, 80-column screen, decent printer, and reasonably sophisticated program are essential for anything but the simplest letters and memos.

(23) Consider the pros and cons of networked versus independent systems.

(24) Pros of networking include:

(a) Lower cost if you need a fairly large number of work stations that can share expensive peripherals such as high-quality printers, plotters, biofeedback input units, etc.

(b) Communication between stations is facilitated.

(c) All units will share only the most recent version of each program, instead of having assorted versions with minor variations.

(25) Cons of networking include:

(a) Possibly higher cost depending on the number of stations required.

(b) Sophisticated special purpose software is required to allow independent access to peripherals and the central unit.

(c) Possible response time limitations (not necessarily, if each station works as an independent unit after downloading a program).

(d) Sharing peripherals may impose limitations.

(e) Physical limitations on where units may be located.

(f) The entire lab depends on one device.

(26) In considering networking, an important distinction is whether stations can also be used independently ("smart terminals" or stand-alone units, vs. "dumb terminals").

DAILY OPERATION

(27) Think about the physical layout: places for cables, papers, both experimenter and subject to sit, etc. A small work station for each computer is a very worthwhile investment.

(28) If you have programming done for you, establish criteria for your programs.

(a) Insist on thorough documentation. DON'T let your programmer keep saying, "I'm going to do that next." Rule 1 in this area is that all programmers would rather spend time putting fancy additions on their program than writing documentation. Rule 2 is that all users would rather have good documentation than fancy additions. One way to combine these conflicting "rules": Ask your programmer to make the program largely self-documenting by putting into the program extensive user instructions that appear on the screen.

(b) Insist on lots of "COMMENT" or "REMARK" lines within the program.

(c) Encourage programmers to use a modular programming format, with relatively independent subsections (each defined and documented!)

(d) Insist on the simplest possible program that will do the job.

(e) Use menus that allow the user to select options.

(f) You may wish to establish a standardized line-numbering system. For example, use odd-numbered lines for comments in BASIC programs. This allows ex-

tensive comments, but an abbreviated version of the program can be produced later by deleting all odd-numbered lines.

(g) Insist that the name, identifying number, brief description, and date of most recent revision appear at the beginning of each program.

(29) Consider the effects of power surges, power failures, lightning strikes. A power outlet strip around the room with separate master on/off switch, always turned off by the last user, may prevent one lightning strike from destroying everything you have.

(30) Establish a system to keep track of your hardware. You may wish to keep a sheet of paper on each item with spaces for information on ordering (price, date, source, purchase order number), use, and notes on maintenance and repair.

(31) Establish a system to keep track of your software. Recognize that any good program will soon have sex and reproduce itself in multiple versions with minor differences. Develop some naming/numbering system that can handle the multiple versions of each program that will soon be floating around.

(32) Develop some system to assure regular maintenance (cleaning disk drives, etc.)

(33) Consider the consequences of inevitable breakdowns and repairs: budget for repairs; have a source for repair work; have backup systems.

(34) Don't allow yourself to become the only available expert. As soon as new hardware or software arrives, immediately teach two other people how to use it so others can ask them instead of you. Another alternative is simply to refuse to learn how to use it yourself.