SESSION VI MISTAKES WITH MICROCOMPUTERS: A SYMPOSIUM

Richard E. Mayer, *President*

My many mistakes with microcomputers (or Four years of fun trying to get my computer to run)

RICHARD E. MAYER

University of California, Santa Barbara, California 93106

This paper presents examples of mistakes committed in the process of developing an inexpensive computer laboratory for problem solving research and a larger multuser cognitive psychology laboratory at the University of California at Santa Barbara. Major mistakes include failure to adequately plan whether the system will work, failure to check for compatibility among components, failure to develop adequate software, and failure to anticipate maintenance problems.

The increasing capabilities and decreasing costs of computer equipment have had an impact on psychological research. Since the mid-1970s, computer laboratories for research and instruction have been springing up in psychology departments around the country. Many are being developed under the supervision of psychologists who are new to the computer revolution. This is to be expected, since computers that are easy to use and low in cost have been on the scene for only a few years.

This paper summarizes my attempts to build a computer laboratory for psychological research at the University of California at Santa Barbara (UCSB). In particular, my goal is to share some of my favorite mistakes with you. My hope is that my story will help you to avoid, or at least anticipate, some wrong turns on your own road to computer self-sufficiency.

I have classified my errors into four major types: (1) start-up mistakes, (2) hardware mistakes, (3) software mistakes, and (4) maintenance mistakes. In the following sections, I will describe and exemplify each type of mistake.

I will draw my examples from experiences with two recent development efforts. The first experience, which I call the "small lab," dates back to 1975, when I first came to UCSB. I was delighted to learn that I would be able to put together a modest single-subject computer

laboratory. The laboratory cost less than \$5,000 and consists of a "personal" computer, floppy disk, printer, subject terminal, and control terminal. It is designed to run single subjects in reaction time experiments. The history of my 4-year struggle to get this system to run is presented elsewhere (Mayer, 1980), so I will simply present highlights in this paper. The second experience at laboratory building, which I call the "large lab," involves a more recent 2-year effort to help establish a multiuser cognitive psychology computer laboratory in our department. This laboratory cost closer to \$50,000 and consists of a computer, a hard disk, a floppy disk, six subject terminals, two slide projectors, a printer, and a control terminal. It is designed to run six subjects independently and simultaneously in reaction time studies, using both alphanumeric and graphic presentation of stimuli.

START-UP MISTAKES (or IS THIS TRIP NECESSARY?)

Start-up mistakes can be made when you fail to adequately determine what your needs are and whether they will be best served by the proposed equipment. The central core of all start-up mistakes is ignorance—ignorance of what you really want, ignorance of the capabilities of what you are buying, ignorance of whether the laboratory will ever be able to run your experiments. There are so many interesting and creative ways to commit this mistake that I cannot list them all here;

Reprints may be obtained from Richard E. Mayer, Department of Psychology, University of California, Santa Barbara, California 93106.

however, I will call on my own vast storehouse of mistakes to give some examples.

Small Lab

When, in 1975, I was given money to buy a modest computer system, I was delighted. In my ecstasy, I quickly looked through the appropriate catalogs with our shop technician and selected a fine package of equipment. Here, I committed the classic start-up mistake, for I had not really investigated whether this equipment was the best (or even adequate) for the research I had in mind. In fact, it was not adequate; there was not enough memory to both load BASIC and use it, the paper tape device I bought was too slow and prone to error, and the terminals would not display sentences as I thought they would. In addition, I did not check to see if there was a laboratory like mine actually running anywhere in the country. The answer, I later learned, was a resounding no. Finally, I had not ever clearly defined what I wanted the computer equipment to do. When it all arrived, I was at a loss to figure out how to make it "work."

Large Lab

My second attempt at computer laboratory building came 4 years later and benefited somewhat from the earlier fiasco. In planning the multiuser laboratory, we spelled out what we wanted the laboratory to be able to do in some detail. We solicited input from five or six different experts on campus and in town concerning the kind of equipment we would need. (Interestingly, two "experts" emphatically claimed that what we wanted was impossible.) We also solicited input from colleagues around the country who were running computer laboratories in similar research domains, in hopes of copying what they had. What we wound up with was not an exact copy of existing laboratories, because new technological advances had already made much of their equipment "out of date." However, our laboratory was close enough to some existing laboratories that we hoped to be able to share software. Also, we did not go ahead with the purchase of equipment until we could line up an electronics expert who would build needed interfacing and a programmer who would develop needed software. In spite of our efforts, we still encountered many start-up problems. These included long shipping delays, need for cabling, clean wiring, and temperature control in the laboratory rooms, and lack of documentation for the slide projectors.

Recommendations

In retrospect, it seems that many of the most troublesome problems could be avoided with a small dose of planning. You should carefully define what you want the system to do and then go out and get all the information you can about equipment and software that can accomplish your goal. Producing a copy of an existing (well-running) laboratory, complete with borrowed software, is a fine idea. In fact, with the increasing number

of psychology laboratories, this is now a worthwhile recommendation for many users.

HARDWARE MISTAKES (or IT DOESN'T FIT)

The second major class of mistakes involves hardware. Hardware mistakes occur when you buy faulty or incompatible equipment. The central core of the hardware mistake involves lack of interfacing—failure to make sure that each piece of equipment is really compatible with the others. It is easiest to make hardware mistakes if you buy many different components from many different fly-by-night companies.

Small Lab

My first run-in with the hardware mistake came when I plugged in all the equipment I bought for my small lab. I soon realized that it did not fit together. Interfaces had to be built for the terminals (so they would "blank" and not "scroll") and for the paper tape reader (so that it could communicate with the computer at a less error-prone speed). In addition, we bought the computer in kit form. While this saved an initial \$150 to \$200, it took months of time and aggravation to finally get it built. In spite of this, the computer was subject to "blow-ups," usually occurring at a crucial phase in research. When we bought additional memory boards to boost the computer's capacity, we soon found that the power supply was not adequate to keep them working.

Large Lab

My second experience with hardware mistakes occurred in conjunction with the large lab. Almost all of the equipment for the large lab was purchased from one major computer company. However, interfacing was still needed for terminals and slide projectors, with the latter job requiring several painful months. In addition, the terminals had to be modified in order to increase their graphics capabilities, and the slide projectors did not convert into a tachistoscope as easily as we had planned. In addition, our friendly "major computer company" tended to send the wrong premade interfaces and then take precious months to replace them with what we really needed.

Recommendations

These experiences tend to encourage thinking about interfacing before all the components arrive. The ideal would be to buy a fully integrated system in which all parts are made by the same company and which will work as soon as you plug everything in. However, if a plug-in system cannot be found to suit your needs, make sure there is going to be adequate documentation and personnel to provide the needed interfacing. If you buy from fly-by-night companies, be prepared for "compatible" components that may be slightly less than reliable.

SOFTWARE MISTAKES (or WHY DOESN'T IT TALK TO ME?)

The third major kind of mistake involves software. Software mistakes occur when you fail to make provisions for developing programs to run your experiments. For example, if each experiment you run requires that you write a lengthy program, perhaps a program in machine language, you are likely to avoid use of the laboratory or to spend more time programming than doing anything else. A failure to make provision for a software operating system is at the heart of many sad computer laboratory tales.

Small Lab

As an example, consider my astonishment when I put together all the equipment for my small lab. It was finally all compatible and it was all in good repair. There I was with a ready-to-work system that had one problem: I couldn't talk to it. It "knew" BASIC, but to write each experiment as a BASIC program would have exhausted the next few years of my research life, and I had serious doubts as to whether it would ever be possible to get my experiments to run effectively in BASIC. Luckily, we found some money for a programmer who wrote an operating system for us. His job was not easy, because of the many components we had added onto the computer: Each required its own special routines. However, the programmer's operating system allowed me to describe the stimuli and the design of experiments without having to do any actual programming. It took months to develop and debug and wound up costing as much as the computer. But when the programming was finished, the small lab was ready to run subjects.

Large Lab

Somewhat wiser from this experience, we approached the multiuser laboratory with more caution. It seemed clear that a major task in the development of the laboratory would be the creation of a set of basic subroutines, or even better, an interactive program for generating experiments. We made arrangements with a software expert before we bought the equipment, with hopes that the equipment and his programs would arrive at about the same time. We built software costs into the overall budget. Still, we were in for some surprises. Due to lack of good documentation and peculiarities of the interfaces we used, modifications had to be made to the programs before they would run on our equipment. Although the package is wonderfully easy to use, there is considerable work involved if a researcher wants to do a study that does not fit the requirements of the program. In addition, we have found that a multiuser laboratory benefits greatly by having an expert in charge: There are almost continual needs for new specialty programs, revisions of existing programs, help in diagnosing program bugs, and assistance to new users.

Recommendations

It is easy to ignore the problem of software development until after the equipment arrives. Many problems can be avoided, however, if you make explicit arrangements for software development early in the project. This includes the honest assessment of cost and inclusion of software costs in your budget. Your goal should be a general-purpose operating system, an interactive program that allows you to generate a wide variety of experiments with little or no actual programming effort. Also, do not forget programs for dealing with data once they are collected.

MAINTENANCE MISTAKES (or WHEN WILL IT BE READY?)

The final kind of mistake I will discuss is the maintenance mistake. In committing the maintenance mistake, you simply fail to face the fact that your system will need continual care. A good way to insure that you will make this mistake is to view your computer system as an ordinary appliance. For example, your trusty refrigerator or TV or toaster may never need repair; you just plug it in and it works for years. Even if there is a problem, a repair person can fix it quickly and locally. If you have this same kind of view of your computer system, you are ripe for many maintenance mistakes.

Small Lab

My small lab provides more examples of maintenance mistakes than I would like to recite. For example, each time a memory board went "bad," I had to determine which one was the culprit. (I eventually got very good at this, taking it as a sort of problem solving task.) Once the bad board was located, I had to ship it off to the repair factory in another state. After several weeks or months, the repaired board would appear in the mail and I could resume my research. After a year or so of this method of repairing parts, we hit upon two great solutions: We found a "computer store" in a nearby town, and we began to keep a supply of spare parts in the department. However, even these improvements were less than ideal; for example, each time the floppy disk needs adjustments (i.e., about every 6 months), it is a 2-h trip to our friendly computer store.

Large Lab

We have been able to avoid many of these problems with the large lab. For example, the laboratory is supported by maintenance contracts. Thus, when there is a breakdown, it is usually repaired within 1 or 2 days. In addition, our department has hired people who have expertise in computer hardware and software. They have been instrumental in keeping all systems running. Yet, laboratory maintenance is not without problems; indeed, one of the most obvious problems is the cost of maintenance.

Recommendations

Before you buy, you must be sure that you will be able to take care of the equipment. Large laboratories probably require maintenance contracts, so figure this cost into your budget. Small laboratories need to be supported by local repair facilities, either in the department or on campus or at a nearby repair shop. If you must mail your components to post office boxes in distant states, you will not have a smoothly running laboratory.

GENERAL CONCLUSIONS

Developing a computer laboratory can be time-consuming, expensive, and frustrating. Yet more and more of us are finding ourselves attracted to the virtues of computerized research. I firmly believe that the biggest mistake of all would be to give up. I am happy to report that both laboratories that I have been involved

with at UCSB are now running effectively. They have already contributed greatly to our research capabilities, as they allow us to conduct studies that were not previously possible. In fact, each of the presenters in this conference has been successful—ultimately. Each of us has helped to put together a laboratory, using a minimum of funds and enabling new research.

In closing, I would like to point out that the job of putting together a laboratory, either a modest one or a multiuser laboratory, is getting easier all the time. There are more choices, more reliable hardware, more ready-to-use software, and more communication among users than there were 5 years ago.

REFERENCE

- MAYER, R. E. My many mistakes with microcomputers. *American Psychologist*, 1980, 35, 117-119.