

# Effective ways to dispose of unwanted time and money with a laboratory computer

DAVID E. KIERAS

*University of Arizona, Tucson, Arizona 85721*

Four general types of mistakes are presented that are commonly made in choosing, purchasing, and setting up a laboratory computer. These are: avoiding advice, building your own hardware, buying from a fly-by-night manufacturer, and being ignorant of your machine.

I've always had the notion that a computer is actually an error amplifier. Thanks to modern computer technology, we can make mistakes on a larger and more impressive scale than ever before. Of course, the really impressive mistakes with computers are made by administrators who have the resources and impact to make errors in computer use on a grand scale. We experimental psychologists are limited by the fact that when we manage to waste time and money with a computer, it usually affects only a small number of people—usually no more than a department, often just an individual researcher. However, we make up for this lack of scale in our errors by the small amount of computer time required to make them. That is, the key feature of our mistakes with computers is that we make most of our best ones before we even start using the computer. We do this by focusing our creative effort on the process of choosing, purchasing, and setting up our laboratory computers, and by exploiting this fertile ground for as many mistakes as our ingenuity allows. Once the computer has lapsed into a state of humdrum routine use, it turns out to be very difficult to make interesting errors.

This ingenious style of wasting resources is so important because now that computer hardware is getting so inexpensive, almost everybody is in the position to exercise bad judgment in the process of setting up a laboratory computer. Years ago, laboratory computers were so few and far between that our opportunities were rather limited, and usually reserved only for senior faculty. Now, even lowly assistant professors such as myself have the privilege of making interesting errors, using the most modern of computer technology.

What I will describe in this paper is four important errors associated with choosing, purchasing, and setting

up a laboratory computer. If one is very careful, one can dispose of impressive amounts of time and money with just these four errors; of course, there are many other approaches, just as in any artistic endeavor. Now I realize that there are undoubtedly a few of you who for some reason do not think it appropriate to indulge in the sport of computer mistakes at public expense. Thus, for you, I will point out how you can avoid the appearance of impropriety by suppressing the natural desire to make mistakes. My comments and examples are drawn largely from my own accomplishments as a waster of time and money, with some examples based on hearsay about masterpieces produced by other experts in the art of computer error.

## THE ADVICE AVOIDANCE MISTAKE

Although very general, the advice avoidance mistake is the most important mistake of all. If you have mastered the techniques of advice avoidance, it greatly facilitates making the other, more specific mistakes. Generally, the technique is to simply rely on your own experiences and background as much as possible in making decisions about what computer to buy, or how to set it up. Of course, the less you know, the better, because the mistakes you can make will be more impressive. If one is so rash as to find out what mistakes others have made, one's artistic opportunities are considerably curtailed; there is a natural tendency to avoid mistakes that you know others have already produced.

One might think that advice avoidance would not be practiced very much; after all, we are trained to collect data, and we try to arrive at rational decisions based on data. However, we are also encouraged to do our own thinking, and also to function independently when it comes to methodology and techniques. It is a sign of inadequate training if we must ask a colleague how to do an analysis of variance; rather, we will struggle with the problem ourselves. Correspondingly, we are supposedly able to equip our own laboratories—Why lose face by asking for advice?

If you do not like to waste time and money, then any computer purchase decision should wait until you have collected as much specific advice as possible. If you know what you want your laboratory computer to do,

Apologies are due for violating the normal serious and scholarly tone of this journal. While this paper is deliberately written in a humorous style, the waste of research resources resulting from the described mistakes is no joke. It is hoped that readers will enjoy this paper, but will take these matters seriously. Work on this paper was supported in part by the Personnel and Training Research Programs, Office of Naval Research, under Contract N00014-78-C-0509, Contract Authority Identification Number NR 157-423. Requests for reprints should be sent to David Kieras, Department of Psychology, University of Arizona, Tucson, Arizona 85721.

contact others who do similar work and find out specifically what their experiences have been. One feature of computer nuts is that they love to talk about their computers and their experiences. If you ask enough people, you will have no trouble getting lots of advice. Some of it may be contradictory, but at least you will have some ideas about what you are getting into. Clearly, though, you must get a large enough sample. I have made a serious error in advice seeking by relying on local experts, who, rather than helping me avoid errors, were actually extremely adept at making them themselves. A larger sample of advice givers might have prevented me from making a serious blunder.

A possible by-product of a thorough search for advice is that you may find someone who is doing almost exactly the same kind of experiments and has already solved all the problems in getting a particular machine to perform these experiments. If so, you can eliminate most of the difficulties of setting up a computer laboratory by exactly duplicating this already working system. Notice that the duplication needs to be exact, however; changing anything at all can easily put you back to Square 1.

### THE DO-IT-YOURSELF MISTAKE

This mistake is simple to state, but somewhat complicated to execute. It is to attempt to build your own hardware. Naturally, you cannot entertain making this mistake unless you have some hardware expertise available. The trick is to not have too much expertise available. This is not as difficult as it sounds, since high-speed computer circuitry is actually very easy to get to work unreliably. Only those technicians and designers who have been brainwashed by extensive conditioning can design and build hardware that actually works reliably, and take unacceptably short amounts of time to do so. Hence, this mistake is available to a great many prospective laboratory computer planners.

A good example of this mistake appears in the history of the IBM 1800 laboratory at the University of Michigan some years ago (see Kieras, 1973). The computer was originally equipped with IBM's video terminals. IBM raised the rent from a ridiculous level to a preposterous one, and the director of the laboratory decided to replace the terminals with some of the new (at that time) stand-alone RS-232 video terminals. Since IBM never heard of RS-232 until recently, it was necessary to build our own interface. The terminal manufacturer helped us along by supplying a design for the interface.

Actually, this was a rare case of multiuniversity computer mistake. Three departments cooperated on the choice of terminals and exchanged ideas on the interface design. This made it possible to arrange the effort so that all three universities involved all made the same type of mistake, but in individually unique ways. Our mistake was interesting in that we almost had too much expertise on hand. But thanks to missing and inaccurate informa-

tion from IBM, we were able to build an interface with several one-shot multivibrators in it, all adjustable, to handle the tricky timing problems. The resulting interface was a very satisfactory result, in that it worked most of the time, so that we went ahead and tried to collect data. But every time the temperature changed substantially, the interface would fail, and we could then shut down until the adjustments were tweaked with the aid of an oscilloscope.

While this was a very fine mistake, your appreciation of it will be heightened when I tell you that the whole process of equipping the computer with new terminals occupied about a year, and since the original terminals had been removed, a very nice full year of idle time on the computer was obtained. It is a sign of the decline of academic standards that a few years later, this enjoyable and interesting interface was completely rebuilt to a design that hardly ever fails.

A variation on this mistake is to have the hardware built by someone with an unusual and original form of incompetence. A case I was told about was one in which the complex interface had been built using the new method of wire-wrap connections, but using stranded wire. So the interface worked very well, except that periodically a strand of wire would stray from its proper place and disable the interface.

Now, if you are not interested in wasting time and money with computers, it follows that you should buy all of your computer hardware from a single manufacturer who has the responsibility for delivering a complete, working system and repairing it under either warranty or a maintenance contract. The amount of money you may save, but probably won't, by building your own hardware will be insignificant compared with the almost certain loss of time, energy, and data.

### THE FLY-BY-NIGHT MANUFACTURER MISTAKE

This mistake is a very easy one to execute and can produce spectacular results, but it requires a certain amount of persistence and fortitude, since you may have to insist on your artistic judgment without the encouragement of your colleagues. Above all, this mistake requires patience, since its real essence is to delay as long as possible the date at which the machine is delivered or becomes usable. Consequently, this mistake achieves its effect by the absence of the machine, rather than an active involvement with it.

I consider myself something of an expert in this mistake, having accomplished it recently in a most spectacular fashion. I had no idea that I could do so well at computer mistakes. This should be encouraging: Never underestimate your talent at computer error.

Due to the artistic excellence of this mistake, it takes some time to describe. We had to collect bids for our laboratory computer purchase. We were expecting a bid from Data General, but the salesman failed to file it. DEC's bid was over our available budget. That left us

with an attractively low bid by a small local firm for a system consisting of a Data General computer, equipped with laboratory I/O interfaces designed and built by this firm. When I timidly inquired of the university administrators whether the bidding process could be reopened, I was rebuffed. This made the low bid look even more attractive. After some token hand-wringing, we accepted it.

The delivery date arrived, and the firm did not deliver the computer, explaining that it would be another week. After another week had gone by, the firm again asked for another week. They explained that they were just about to get the circuit boards back from the etching shop. After more delay, they agreed to deliver the Data General computer so that we could at least learn how to program it. After some months, most of the I/O interface system was delivered. Not only was it incredibly inefficient and difficult to program, but it was rather unreliable as well. More months went by while the firm made modifications in their hardware, which we then tested with real experiments and subjects. It was an admirable example of cooperation between industry and academics—They built the equipment, and we tested it. After about a year after the original delivery date, the firm decided that they couldn't get the video terminal interface to work and agreed to supply some Data General interfaces as a temporary measure. It took me about a week to write and test the software for these interfaces, and they have worked completely reliably ever since. We then asked the firm to replace the rest of the interfaces with Data General products, and we were rebuffed. We thereupon junked their interface hardware, came up with another few thousand dollars, and ordered some Data General interfaces. We demanded some of our money back from the local firm. Thanks to a clerical error, we had not paid them for some of the hardware we wanted to keep, so our loss was only a few thousand dollars. The local firm shortly thereafter went bankrupt, insisting all the while that their hardware worked. Our present, all Data General, computer is quite reliable and easy to program.

Thus, with this one mistake, I managed to waste a considerable amount of time—not only my own, but our engineer's as well—and several thousand dollars, and to delay the start of the laboratory's useful life by a full year.

Note that there were three steps in making this mistake: first, the careful selection of the fly-by-night firm; second, the cunning repeated extensions of the delivery date; third, the dramatic junking of the unsatisfactory equipment and replacing it with new equipment. You might ask how I had the patience to wait a whole year. The answer is that I had part of the machine to work with, and I had the distraction of trying to write grant proposals at the same time. When the money arrived to do computerized experiments, there was no working computer to do them on: a well-executed mistake.

For those of you who do not enjoy wasting time and money, here's what I would suggest: Never, never, *never* buy a computer from a firm that might be fly-by-night. Even if their price is seductively low—Keep your money and wait until you can buy something good. The system we ended up with is very satisfactory (Kieras, 1979), but we could have had it running at least a year earlier by ordering everything from Data General in the beginning, instead of trying to save money. So, buy only from one of the major manufacturers of computer equipment that have established reputations and will want to protect them.

If for financial or administrative reasons you must deal with a possibly fly-by-night manufacturer, prepare as detailed a set of specifications as possible. The key part of these specifications is an acceptance test that specifies what the machine must actually be able to do when delivered. This should be something very similar to your actual application. Since it may take you some time to get the programming done for your actual experiments, you may not find some of the flaws in the machine for some time. If these are design flaws, rather than simple malfunctions, you will be in serious trouble. One way to protect yourself would be to include in the specifications a description of what functions the machine is supposed to provide in your laboratory and a stipulation that the manufacturer guarantees that the machine will perform these functions in a reasonably efficient and easily programmable fashion. Finally, absolutely do not hesitate to cancel the contract immediately if the firm fails to deliver a working computer at the contracted time, and do not hesitate to take administrative or legal action if the manufacturer does not satisfy the contract.

### THE IGNORANCE-IS-BLISS MISTAKE

This mistake is open to most psychologists because our training does not normally include formal training in computing. What this mistake involves is remaining as ignorant as possible of the computer and its software. This mistake mostly affects the initial programming of the computer, since usually some system-level software has to be written before the computer can be used, and the first experiment programming sets the pace and style of those that follow. Maintaining ignorance makes two specific forms of mistake easy to make (see Kieras, 1976):

The first specific form of the ignorance-is-bliss mistake is that the wheel can be reinvented frequently if you are adequately ignorant. Modern computer science has figured out efficient and simple ways to do a lot of things; by remaining as ignorant as possible of these, you can have lots of fun reinventing them, and thus waste lots of time. I have had a lot of fun this way myself; during graduate school, I reinvented multitasking, queues, and several other techniques. But I thought that

parsing FORTRAN-style arithmetic statements was too complex. However, later we had a programmer who was getting a bachelor's degree in computer science. He solved this problem by simply copying a flowchart from his class notes. In my present position, unfortunately, we have a highly skilled programmer who is a computer science doctoral student, and who knows state-of-the-art programming techniques (e.g., Ganoë & Kieras, 1980). Consequently, I've had to find other wheels to reinvent.

The second specific way to capitalize on ignorance is that if you choose to hire a programmer, your chances of getting an incompetent programmer are considerably improved if your ignorance is maintained. If you don't know much about your computer, you can't tell whether anybody else does either. Consequently, your chances are very good that you will hire someone who will help dispose of your time and money. This mistake is probably very popular and successful among experimental psychologists (Kieras, 1976).

Again, for those of you who like to avoid mistakes, the only thing I can recommend is that you plan on learning a lot about computers. A complete background would include not just expertise in a high-level language, but also assembler language experience—preferably on more than one machine—and an understanding of how operating systems work. You should especially know how the interrupt servicing software in your own system works. This kind of background will enable you to distinguish good from bad programming and know what the limitations and possibilities of your machine's software actually are. An example of the important practical advantages that can result from such understanding is

that sometimes there are simple "kludges" that greatly extend the performance of the system software. But to find them and use them requires knowledge at this advanced level.

## CONCLUSION

What I've described, in a flippant fashion, is some of the general problems that experimental psychologists have in trying to set up a computerized laboratory. Many more specific examples of these mistakes could be supplied, since a successful, error-free laboratory seems to be very rare. This means that the waste of effort and money involved in laboratory computerization must be astounding. The greater availability of computers, especially cheap ones with rather crude software capabilities, could just mean that in the future even more psychologists will be wasting even more resources with computers.

## REFERENCES

- GANOË, W., & KIERAS, D. A virtual terminal program for data transfer between a Data General MicroNOVA laboratory computer and a DECsystem-10 timesharing system. *Behavior Research Methods & Instrumentation*, 1980, 12, 251-255.
- KIERAS, D. E. A general experiment programming system for the IBM 1800. *Behavior Research Methods & Instrumentation*, 1973, 5, 235-239.
- KIERAS, D. E. Staff programmer vs. system: The issues. *Behavior Research Methods & Instrumentation*, 1976, 8, 174-177.
- KIERAS, D. E. Doing it the vendor's way: Running multiple subjects in reading experiments using Data General's Diskette Operating System. *Behavior Research Methods & Instrumentation*, 1979, 11, 221-224.