Using MEL in a networked classroom

ALFRED A. VIEIRA
University of California, Berkeley, California

With Micro Experimental Laboratory (MEL) and network software in a computer-assisted class-room, an instructor can run class demonstrations and projects with great control and flexibility. The network allows quick file transfer between the instructor's and the students' machines. MEL allows easy reprogramming of experiments for more interactive demonstrations, and data merging for comparison of individual and group data. In this article, the pros and cons of running MEL on a network are discussed, and its use at the University of California at Berkeley is outlined.

Why use a network? When one gives a class demonstration, one would like all the students to be as involved as possible. At the same time, one wants the flexibility to change the demonstration as one sees fit. Running a demonstration in a networked classroom satisfies all these concerns. Students can have their own terminals and make individual responses; they get a good sense of what it is like to be subjects in psychology experiments. At the same time, the instructor can easily send new programs and data files to the students' machines, moving easily from experiment to analysis and back again.

A network can be set up so that all file-moving tasks are done from a single, central computer. There are several advantages to such a network design. Files can be distributed quickly and easily, with no need to copy software onto each machine by floppy disk. The use of batch files can automate file-maintenance tasks to free up time for the instructor or teaching assistant. Most importantly, the students can concentrate on the material presented, instead of being distracted by the everyday mechanics of using the computers themselves.

Problems in Using a Network with MEL

Both MEL and the PC-Network program that we have been using are large software packages with sophisticated requirements (see Schneider, 1988, and *IBM PC-Network Program*, 1985, for information on the requirements and capabilities of these programs). This means that there are a few areas where their requirements conflict, although the impact is minimal for classroom use.

The PC-Network program that we have been using must remain memory-resident, and in some configurations it can take up more than 100K of the PC's 640K system memory. This means that very large MEL programs and some of the MEL tutorials cannot be run with the network in place. These are not serious problems, however, for two reasons: (1) programs to be used for class demonstrations are generally scaled-down versions of experimental programs, with smaller memory requirements, and

(2) the MEL tutorials are a self-paced learning system, and there is no need to have the network operative while using them.

Depending on the network configuration one chooses, the network software may have to perform various polling or message-passing tasks at each of the machines connected to the network. This means that the network software may have to steal processing cycles from a running MEL program, which could make timing less precise. Again, this is not of much concern when MEL is used in the classroom, because the data collected are for demonstration purposes and not for publication. The strategy I have used to avoid this problem is to run the experiment without the network, and then install the network later, to make the data analysis more convenient.

Specific Benefits

Earlier I mentioned some general reasons why one might want to use a network in a classroom setting. While setting up a network to work with MEL, I discovered some additional ways in which the two systems work especially well together.

The FORM program used in MEL to generate experiments allows one to make quick modifications of programs. In conjunction with the network, this gives a lot of flexibility for classroom use. One can run an experiment in one way, reprogram it, distribute the new program, and be running the second version in just a few minutes. This is ideal if one is working with teaching assistants; they can reprogram while the last demonstration is being explained. Perhaps most interesting from a teaching standpoint is that students can ask "what if" questions about the material and get their answers immediately in class.

Because MEL is easier to learn and use than traditional programming systems, it is possible for students to perform original experiments, or modifications of class demonstrations, as projects for an experimental design course. This was one of the first applications of our instructional lab at Berkeley. The network allows the instructor to easily distribute "shell" programs (programs that have certain forms already filled in for common functions, such as collecting responses, block design, etc.) or

other example programs to the students' machines. These can be modified for projects. It is also simple to maintain a library of example programs in a shared directory on the central machine, so that students can have access to an "idea bank" for experimental projects.

One of the most useful tools for our statistics and experimental design course was MEL's ability to merge data files. Coupled with the network, the MERGE program allowed us to fetch data files generated on the students' machines and combine them into a single class data file. While the students analyzed their individual files, we were able to fetch their data and perform the same analysis on the group data. Seeing both the group data displayed on the blackboard and their own data on their local machines, the students gained a true appreciation for individual differences and the greater reliability available from group data.

How Is Our System Organized?

Some of the network benefits that I have described were made possible by the particular configuration of the PC-Network program used, so I will describe briefly the network design in our lab.

In order to give the instructor the most control of file movement, the hard disks of all student machines are shared on the network at the level of the root directory. To the instructor, the hard disk of any of the remote machines can be treated as if it were physically located in the instructor's local machine. In the PC-Network program, this means that every machine is designated as a file server. Although this has its cost in memory, it allows instructors to keep the lab's filing system neat; they can make and delete directories and files on all of the machines in the lab.

Because all of the machines have their hard disks shared on the network, there is a possibility that a student experienced with computers could access other machines and generally wreak havoc on the system. To protect against this admittedly unlikely possibility, a password has been assigned that limits access to the remote machines. This password is hidden even from the instructor, because network file operations are carried out through the use of batch files (see Norton, 1987, for information on advanced DOS batch files).

These batch files are located only on the instructor's machine, and they cannot be accessed by any other machine on the network. They treat the disks of the remote machines as if they were the DOS drive D: of the instructor's machine, and provide powerful methods for creating and removing directories and files, fetching files from remote machines, and distributing files to remote machines. Many of them allow more than one machine to be accessed. For example, one command allows a whole diskful of files to be sent to all machines on the network; one can start it going, go out to lunch, and pick it up when one gets back.

In our system, the instructor's machine has a RAM disk that is shared on the network and that can be accessed by all the machines on the network. It is like a small, fast disk treated as DOS disk D: of the students' machines. The instructor can place files here that the students should have access to. For example, we have an editor loaded into this disk for students to use on the remote machines. We also have a printer attached to the instructor's machine that is shared on the network, and all machines can print using another batch file. Each printing is identified by the machine number that requested the output, so it is easy to determine which printout belongs to each student.

In short, using MEL and a network together can open up new ways to use computers in the classroom; it can ease many of the mundane chores of using a computer laboratory for instruction.

REFERENCES

IBM PC-NETWORK PROGRAM: USER'S GUIDE. (1985). Boca Raton, FL: International Business Machines Corporation.

NORTON, P. (1987). *Peter Norton's DOS guide*. Englewood Cliffs, NJ: Prentice-Hall.

SCHNEIDER, W. (1988). Micro Experimental Laboratory: An integrated system for IBM PC compatibles. *Behavior Research Methods, Instruments, & Computers*, **20**, 206-217.