SESSION X

SYMPOSIUM

MICROEXPERIMENTAL LABORATORY (MEL): AN INTEGRATED SOFTWARE SYSTEM FOR COMPUTERIZED EXPERIMENTATION FOR RESEARCH AND INSTRUCTION ON IBM PC-COMPATIBLE COMPUTERS

Walter Schneider, *President*
University of Pittsburgh

Enhancing a standard experimental delivery system (MEL) for advanced psychological experimentation

WALTER SCHNEIDER
University of Pittsburgh, Pittsburgh, Pennsylvania

Enhancements to the Micro Experimental Laboratory (MEL) and the ways in which they facilitate development of psychological experiments are discussed in this paper. New procedures are described for (1) presenting and manipulating pictorial graphics, (2) inputting response box, voice key, joystick, digitized speech, and mouse responses, (3) outputting lights, speech, and tone stimuli, (4) editing and analyzing digitized speech, and (5) implementing multitask processing. Methods for extending the language, developing test batteries, sequencing subjects, and managing and analyzing data are described.

The Micro Experimental Laboratory (MEL) is an integrated software system for experimental research that is used in over 250 laboratories in 13 countries. MEL allows the user to author experiments by filling in blanks on forms. It can run most discrete-time experimental procedures, such as reaction time, reading comprehension, and questionnaire experiments. MEL currently runs on IBM PC-compatible computers and is expected to run on Apple Macintosh computers before the end of 1989. MEL is an extensive system (e.g., over 100 files, 10 floppies, 500 pages of documentation). The system has been described in detail by Schneider (1988). MEL won the 1988 EDUCOM/NCRIPTAL award as the best software product for behavioral sciences. An earlier version of the

system has been reviewed by Butler (1988). In this issue, there are descriptions of the use of MEL in undergraduate laboratories (St. James, 1989), on networks (Vieira, 1989), and for multiuniversity research projects (Swanson et al., 1989). This paper describes recent features that have been added to the system to facilitate a wider range of experiments, to speed the development of test batteries with MEL, and to simplify analysis and reporting of results.

MEL is a program with continued support and enhancements. The extensive user base provides suggestions for enhancements that are implemented in new releases. The vast majority of user requests have been implemented. In the past year, the new enhancements include support of advanced pictorial graphics, response box and voice-key input, mouse input, tracking tasks, speech input/output, language extensions, test battery development tools, subject sequencing and management, and new data analysis and display formats.

This research was supported in part by Office of Naval Research Contracts N00014-87-K-0397 and N00014-86-0678 and Army Research Institute Contract MDA903-86-C-0149. Reprint requests can be sent to Walter Schneider, Learning Research and Development Center, 3939 O'Hara St., University of Pittsburgh, Pittsburgh, PA 15260.

Advanced Pictorial Graphics Support

MEL allows easy importation, display, and modification of pictorial images. Images are stored in computers as *bit maps*—a map of all the dots, or pixels, that define the image. For example, on the IBM VGA graphics screen, the image is defined as a set of pixels, 640 wide by 480 high, each of which can be one of what is typically 16 colors selected from a color palette of 264,144 colors. MEL can display bit maps that are stored in PC-Paintbrush¹ format. Pictures initially can be created with PC paint type programs allowing creation of colored images. Existing images can be scanned with either low-cost hand scanners² or high-resolution scanners. In addition, pictures in a variety of formats (e.g., .PIC files or files created on a Mac computer) can be converted to PC-Paintbrush format with utility programs.³ Pictures can be presented with millisecond timing of the stimulus presentation and the subject's responses.

There are extensive routines to allow manipulation of bit-mapped screens. Bit-mapped images can be loaded from disk, copied to the screen or to internal virtual images, and written to disk. Either full or partial images can be copied to different locations on an image. For example, in a recognition experiment, subjects might see a police lineup of eight pictures. Thereafter, the screen might go blank, and a section of one of the eight images (e.g., the eyes and nose sections of a picture) might be presented in the middle of the screen. The subject's task might involve indicating the position of the picture. Virtual images allow storing of internal pictures (e.g., 40 pictures of objects represented on four virtual images stored in memory but not visible to the subject). Then, under program control, a segment of one of the pictures for a specific image can be copied from the virtual image and displayed in any location of the subject's display. In this way, one can treat a sequence of images like sequences of letters, creating many arrangements of the pictures.

Bit-mapped images can be modified with normal image drawing routines in MEL. MEL supports a variety of line-drawing graphics. For example, a picture of the brain can be displayed with lines drawn on the picture, and the subject can be asked to label (i.e., specify the names of) the parts being pointed to.

Pictures also can be displayed in conjunction with questionnaires to allow free mixing of graphics and text. For example, a physics problem could be presented with a line drawing of the pulleys and weights with a multiple choice question below it.

Response Box, Voice Key, Lights, and Sound Generation

Experimental research requires flexibility in the type of input and output devices needed to interact with subjects. For example, neurological patients or children may not be able to successfully operate a computer keyboard. MEL supports a special-purpose response box interface⁴ for subject input and output. The response box interface plugs into the standard printer port on a computer. The response box provides four push button keys and a voice-

key input. The voice key produces a simulated response when the speaking volume of a subject exceeds a given threshold. Both the voice-key input and the response input can occur simultaneously without loss or delay of the signals. The response box provides subjects with a simpler way to respond than is provided by the keyboard (the standard MEL input device). The response box can be rewired to allow different arrangements and numbers of keys for alternative tasks, can be reconfigured to support up to 16 response keys, and will accept any type of input through a closure switch.

The response box outputs support light emitting diode (LED) displays (two red, one yellow, one green) and sound output on a headset (tones and white noise). The LED outputs can be rewired to control external equipment (e.g., by using an opto-isolator or reed relay to activate another piece of equipment). For example, one could use the response box to operate the controls of a video tape recorder.

The response box provides headset output to enable individualized presentation of tones and white noise to subjects. Many subject-testing laboratories have multiple workstations in the same room. This makes it difficult to provide auditory feedback to individual subjects. Allowing one subject to hear another subject's feedback tones can confound a study and should be avoided. For older PC computers, one can rewire the speaker output to a headset to provide individual feedback. However, this rewiring may void a manufacturer warranty. Unfortunately, newer computers (e.g., IBM PS/2 line) tend to use piezo-electric wafers for auditory output that cannot be rewired to output through a headset. The response box eliminates the need to rewire the computer and provides the experimenter with adjustable volume control. The box includes a white-noise option that masks the sounds of other subjects (e.g., keyboard noises). I have found that allowing subjects to wear headsets with white noise provides a unique context that facilitates cooperation of subjects.

Speech and Sound Input and Output

Some experiments require auditory input and output. For example, when doing a Stroop (1935) experiment, subjects are required to name the color of a word while ignoring the text (e.g., say "red" to the word "green" printed in red ink). Speech output is useful when the experimenter wants to divide the subject's attention between visual and auditory stimuli. The advantage of speech stimuli is that they can be listened to without regard to the position of the subject's eyes. Adding speech input/output allows a greater range of experimental procedures. Recently, the advent of low-cost speech/input output devices has opened up a range of novel uses of computers.⁵ The low-cost boards generally are designed for telephone-answering systems and produce telephone-quality speech, which is acceptable for many applications. Some of these boards use signal compression procedures, allowing storage of extensive speech inexpensively (e.g., 10 min of continuous speech on a 1.2-Mb floppy). In a typical experiment, a subject speaks only a short period

of time during a trial (e.g., .5 sec out of a 5-sec trial). Thus, a full session's worth of speech can be recorded and stored easily and cheaply. Storing the data in digital form allows much more efficient postprocessing (see below) than does analog auditory tape.

Speech input and output can be controlled by MEL. There are special commands to play or record a fixed duration of speech. The sound is input via a direct memory access (DMA) board that stores the data without interrupting the ongoing program. This allows speech input/output to proceed concurrently with the critical display and response timing. For example, a list of words could be spoken by the computer, while the displays are changed and responses are being made. The experimenter records simply by using a record command. The recorded sounds and experimenter-specified information can be stored on disk for later analysis.

After speech data is collected, they can be analyzed with both human and machine analysis. A sorting program allows rapid postexperimental scoring of the speech input. The speech can be replayed by trial number, subject response, or the correct response. For example, in a Stroop experiment, the data might contain 1,000 utterances of color names. With analog tape methods, an experimenter might listen to the original hour of utterances, while checking off the errors using an answer sheet. The data then would be entered into the computer. With MEL, the words can be replayed trial by trial. The experimenter can listen to the utterances, with the correct answers displayed on the screen, and can press a response for each error to score the spoken input. Since the intertrial intervals are skipped, a 50-min session could be coded in 10 min. At any point, the sound can be backed up to replay sections of input. A more efficient way of scoring data is to play the data grouped by the correct answer. For example, in the Stroop experiment, all of the responses with a specific correct answer would be replayed, then the next answer, and so forth. Thus, the experimenter would hear the same input for all the corrects, and the errors would stand out (e.g., hear "red, red, . . . red, green, red . . ."). This allows for accurate coding of responses even when the experimenter is somewhat bored with the coding task (a common problem). Also, the data can be output with intentional errors (play a sound with a different correct response) to allow assessment of a coder's accuracy. As an added benefit, the system can also display the amplitude of the speech waveform and automatically record voice onset times.

Digitized auditory protocol analysis provides important advantages over scoring protocols from analog tapes. Recording the data in digital form speeds transcription and facilitates timing and verification procedures. Transcribing an analogue tape is slow, and accurate timing data is difficult to maintain. When a coder reverses the tape, the detailed timing data is usually lost. Also, the forward and backward movement of the tape is inexact. The digital protocol system can examine the voice amplitude envelope of the speech to ease transcription. The tape can be backed up a fixed period of time or to the previous si-

lence gap. A fixed segment of sound (between silence periods) can be replayed until the operator presses the enter key. The sound amplitude envelope can be displayed so that the recorded speech can be matched closely to the transcribed text. A segment of sound can be marked and replayed multiple times until the user presses the enter key. Sound segments can be randomized to keep context effects from influencing the person doing the transcription (see Ericsson & Simon, 1984, chap. 6). Since digitized speech can be replayed at any time, it is often unnecessary to actually transcribe the text. For example, the coder might simply press a key, coding whether an aggressive or conciliatory statement is made by the speaker at each time segment. Transcriptions can be verified at random intervals by a supervisor who listens to random segments of the auditory input and compares it to the typed text. The amplitude envelope of the data can be analyzed (e.g., how much of the time was a subject silent during various conditions) by the computer without human input.

MEL has a speech editor that allows for the recording and manipulation of speech sounds. The editor displays the speech amplitude waveform, and sound segments can be cut out and/or moved. The experimenter can produce a series of sounds (e.g., the spoken digits), which can be selected in the experimental program in random orders, as needed in an experiment.

Mouse Input/Display and Tracking

MEL allows real-time input and display of the mouse input. A mouse is a computer input device for pointing. The subject moves the mouse on a desktop and pushes the button indicating an input. The mouse provides a graphic pointing device that can move in any direction on the screen. This allows diagonal movement between random points on the screen. A moving cursor is displayed on the screen, reflecting the mouse movements. The joystick (a movable lever) and the track ball (an inverted mouse where one moves the ball rather than the mouse) also served as input devices.

In contrast, use of cursor keys that require a subject to make multiple vertical and horizontal keypresses to get to a new location can be a serious problem, because it forces the subject to plan the movement pattern and attracts his/her attention to portions of the display that may be irrelevant (e.g., to the corner of a display when trying to move across a diagonal). Mouse input is particularly useful when asking subjects to identify an object on a complex display.

MEL can be programmed to read the position of the mouse and to record a mouse buttonpress as a response. In addition, portions of the display can be treated as response areas, or hot spots (e.g., a box around an icon). Any mouse input in one of these areas is treated as a keypress for later scoring.

MEL supports a concurrent tracking task to provide a continuous workload task. The task is a compensatory tracking task (see Kantowitz & Buck, 1983) that can be controlled either with a control stick connected to an analog port or with a mouse input. The stick can control the

position, velocity, or acceleration of a moving cursor. The subject tries to keep the cursor on a cross hair in the middle of the screen. The accuracy of the stick position is reported.

Multitask Processing

MEL allows two separate experimental procedures to be programmed independently and then run concurrently. For example, in a dual-task experiment, one might want the subject to categorize words as animals while doing mental addition. Programming the combined task can be complex (e.g., what happens if the person makes two responses for one task before the other?). Also, many contingencies need to be programmed (e.g., when is a response meant for task A or B?). MEL greatly simplifies such programming by allowing the experimenter to specify two different event tables, with separate sets of keys for each task. The *event tables* are lists of real-time display and response events. The two event tables are run concurrently, each controlling the screen, collecting responses, and providing feedback independently. They can be locked so that an event on one event table can trigger or terminate the other table. This allows simple programming of stimulus onset asynchrony (SOA) experiments or dual-task experiments.

Extending the MEL Language

MEL is an open architecture with facilities for extending the language to support special functions and for interfacing with new hardware. This architecture allows a few sophisticated users to develop enhancements for a whole community of researchers. For example, a department might hire a full-time or student programmer to add enhancements to MEL. There are three methods of adding enhancements to MEL. First, one can write subroutines in the MEL language that can be included in other people's programs. For example, in my laboratory, my programmers have written subroutines for generating and displaying performance graphs to subjects (i.e., accuracy as a function of blocks in the experiment). These subroutines are now in a library that other experimenters can use. By passing the routine an array of numbers, subjects can have pleasing graphs displayed or printed. As more users develop MEL applications, departments will develop libraries of such subroutines. Psychology Software Tools, Inc., maintains a public domain library to facilitate the transfer of such subroutines.

The second method involves linking programs to the object code version of the MEL RUN program. This requires that the laboratory own a copy of the Microsoft Pascal compiler (version 4.0) and any other compilers that will generate code that is to be linked with the compiler (e.g., Microsoft C, Assembly, or FORTRAN). The user writes code in any of these languages and links it to the MEL program. The user then writes MEL code that calls the user-written subroutines during experiment initiation, trial generation, real-time event execution, or experiment termination. In my laboratory, we used this method to program a social interaction laboratory. We wrote Pas-

cal subroutines to pass messages between MEL programs on a local area network. The experimenter called a special function to pass messages to other machines.

The third method for enhancing MEL is to request that Psychology Software Tools make the changes. If it appears that a large number of users would benefit from the change, it is likely to appear in the next release of the system. If not, Psychology Software Tools will add special functions to the language for a fee. For example, an Air Force laboratory wanted a special tracking stick interfaced. Special routines were written for the Air Force by the programmers that had developed MEL.

These three methods provide a wide range of procedures to extend the MEL language. Because of this extensibility, virtually any experimental procedure that can run on an IBM PC-compatible computer can be implemented with MEL.

Test Battery Development Tools

Simplifying the distribution of experimental procedures between laboratories greatly speeds the dissemination of research techniques (see Swanson et al., 1989). MEL provides a standard environment to speed development and distribution of test batteries. First, experiments written in MEL can be run on almost all IBM-compatible computers and, in the future, on Apple Macintosh computers.⁶ As long as certain programming conventions are met (e.g., displaying text in a limited subset of colors), developers can write a program that will run any of millions of computers that exist in university and research laboratories. Second, the expanding base of MEL users (e.g., projected 500 universities and 100 hospitals by December, 1989) provides a large number of users that already have the equipment, software, and knowledge to deal with software written in this language. Third, MEL allows developers to distribute their own experiment files and demonstration copies of the MEL RUN system without charge. Fourth, MEL includes a menu front end that can be configured to allow unsophisticated users to select and execute experiments from a test battery. Fifth, routines are available to simplify individual patient reports that can be used in clinical settings. Sixth, extensive sample programs are provided to generate test batteries.

Subject Sequencing and Scheduling

MEL provides an extensive subject management system. The subject program provides a method of counterbalancing subjects across sessions, days, and/or computers, including running different subjects on different experimental tasks. Subject floppies can be designed so that subjects need only put the floppies in machines and follow directions written on the computer screen. Experiments can be run over a period of weeks with little external experimenter input. The sequence (SEQ) program sequences the subject between experiments, enters specific conditions, and can even initiate automatic data analyses at the end of each session.

A subject pool management program is being developed to manage large subject pools. Some psychology depart-

ments have to coordinate the running of thousands of subjects in hundreds of experiments. The program (1) keeps track of student participation in experiments, (2) selects students randomly based on their schedules, characteristics, and previous experimental history, (3) produces notices indicating when subjects are to appear, (4) creates weekly rosters to post when subjects are to appear (categorized by class), (5) records participation credit, and (6) reports the number of subject hours used by each experimenter.

Data Management Tools

MEL provides tools to manage large volumes of data. A MERGE program allows data to be combined across multiple floppies that provide automatic backup facilities. If an experiment is run on many computers, the data can be collected on multiple floppies or via a network (see Vieira, 1989). The MERGE program will combine the data onto one master file. The MERGE program maintains pointers on the multiple original data collection and master data analysis file to prevent an experimenter from accidentally deleting data or remerging the data repeatedly (causing the same observations to be counted twice). The system also maintains the data on the original file to allow recovery of the data if the master file is lost. A utility backup program (called BK) copies only new data back to the backup device. This prevents the common operator error of accidentally copying the backup over the new data.

Data Analysis and Graphics Options

The graphics and statistical support are being enhanced in MEL to simplify use and offer a wider range of options. MEL now supports line, scatter, and bar graphs. The user can rescale the graphs, print graphs, file graphs for later examination or batch printing, or incorporate graphs into documents that are produced by popular word processors. The output graphs can be printed on dot matrix, ink jet, or laser printers.

New statistical routines for group tests, analysis of variance (ANOVA), and simple regressions have been added. A special effort is being made to make the statistics usable by individuals with limited statistical backgrounds. For example, one can perform an ANOVA by simply specifying what variables are fixed or random effects, and within or between subjects. The ANALYZE program then generates the general linear model for the design and specifies the terms for the F tests or quasi- F tests.

Conclusion

MEL provides a comprehensive system for psychological experimentation. Researchers can now use standard tools (e.g., concurrent speech recording with presentation of pictures) that were unavailable (due to cost and lack of software support) only a year ago. MEL is being continuously enhanced to provide extensive tools for experimentation that are easy to use. The way in which psy-

chologists do research is changing as standard support tools become available. With a standard interface, researchers at hundreds of universities can now exchange runnable experimental procedures. Individual laboratories previously developed software systems from scratch. These systems were generally very limited tools, used in a single laboratory, and not well documented. No single university psychology department could afford the immense expense required to develop, document, and maintain a comprehensive software solution. MEL is providing a standard that is supported by the combined purchases from hundreds of departments. This allows a concentration of resources that enables the development of a comprehensive system. The tools provided in MEL allow researchers a standard platform for development and distribution of experimental procedures. The system is extensible enough to allow researchers to enhance the system for specialized needs. My goal in developing MEL has been to allow psychologists to focus on research issues rather than on implementation issues.

REFERENCES

- BUTLER, D. L. (1988). A critical evaluation of software for experiment development in research and teaching. *Behavior Research Methods, Instruments, & Computers*, 20, 218-220.
- ERICSSON, K. A., & SIMON, H. A. (Eds.). (1984). *Protocol analysis: Verbal reports as data*. Cambridge, MA: MIT Press.
- KANTOWITZ, B. H., & BUCK, J. R. (1983). Feedback and control. In B. H. Kantowitz & R. D. Sorkin (Eds.), *Human factors: Understanding people-system relationships*. (pp. 369-395). New York: Wiley.
- ST. JAMES, J. (1989). The MEL Library in the undergraduate research methods course. *Behavior Research Methods, Instruments, & Computers*, 21, 245-247.
- SCHNEIDER, W. (1988). Micro Experimental Laboratory: An integrated system for IBM PC compatibles. *Behavior Research Methods, Instruments, & Computers*, 20, 206-217.
- STROOP, J. R. (1935). Studies of interference in serial verbal reactions. *Journal of Experimental Psychology*, 18, 643-662.
- SWANSON, J. M., NUECHTERLEIN, K., ASARNOW, R., TAYLOR, E., HUNT, R., SHAYWITZ, B., POTKIN, S., EARLY, T., CLONINGER, R., & POSNER, M. (1989). Use of a common software system for collaborative clinical research on attentional processes in schizophrenia and ADHD. *Behavior Research Methods, Instruments, & Computers*, 21, 250-254.
- VIEIRA, A. A. (1989). Using MEL in a networked classroom. *Behavior Research Methods, Instruments, & Computers*, 21, 248-249.

NOTES

1. Paintbrush is a trademark of the ZSoft Corporation, Marietta, GA.
2. For example, Logitech sells a hand scanner for \$164. It can scan 4-in.-wide photographs from textbooks into bit-mapped images.
3. For example, the program HIJAAK by Inset Systems, Danbury, CT.
4. The response box interface is made by Random Access and is distributed by Psychology Software Tools, Inc., Pittsburgh, PA.
5. For example, the Digital Card by Computer Age, Nobleton, Ontario is under \$100.
6. Psychology Software Tools, Inc., has projected the release of a RUN program in September, 1989. This release will allow programs written on the IBM to run on IBM or Apple Macintosh computers. A full Macintosh version of the MEL system is projected for release in January, 1990.