SESSION IX RESEARCH AND INSTRUCTIONAL APPLICATIONS

S. E. EDGELL, *University of Louisville, President*

An on-line psychology laboratory for teaching and research

ALAN M. LESGOLD and ROBERT J. FITZHUGH

Learning Research and Development Center, University of Pittsburgh, Pittsburgh, Pennsylvania 15260

This paper describes a PDP-15 timesharing system in which on-line experimentation and an undergraduate laboratory course co-exist with many other users. Design features include precise stimulus and response timing, easy experiment programming, easy system access, and 15 or more simultaneous users. Students can be both experimenters and subjects in realistic, variable course modules.

Current psychological experimentation frequently requires accurate response latency measurement, close control of stimulus presentation, and stimulus generation contingent on prior responses. As the need for precision has increased and computer costs have declined, an increasing number of psychologists are using the computer to control experiments. A variety of experiment control systems have been developed at universities and research laboratories (see the March, 1975, issue of *American Psychologist* for an overview), and a number are commercially available from computer manufacturers. A significant number of experimental psychologists have become computer proficient, and a computer users subculture has developed in which bytes and analog-to-digital converters are discussed much as were relay racks a decade ago.

In spite of these dramatic changes, the undergraduate's laboratory experience often consists of the psychology of flash cards and memory drums, even at universities where computer-controlled research on cognition is most advanced. There are many reasons for this. Some on-line systems are designed for the experienced user and are complex or specialized for particular experiments. Others lack higher level software features and are

simply awkward or difficult to use, particularly by freshmen students with no math or computer background. In addition, most existing systems would be unable to accommodate more than a few students without disrupting ongoing experimentation.

At the Learning Research and Development Center (LRDC) at the University of Pittsburgh, we have developed a system which simultaneously supports both teaching and on-line cognitive research. We have enlarged the scope of an operating system designed originally for computer-assisted instruction so that it can support multiple on-line experiments, a computer-based cognitive psychology lab course, and a variety of other applications, satisfying the essential timing demands of each. These operating system modifications combined with a number of programming techniques allow on-line experimentation to be mixed freely with other uses on a single general-purpose shared system.

THE SYSTEM

The timesharing system which provides these services is called the Experimental Time-Sharing System (ETSS). The system has been described elsewhere in detail (Fitzhugh, 1973; Fitzhugh & Glaser, 1975). It is based on a DEC PDP-15 computer with 48K words of memory, a one-million-word drum memory system, two DECTAPE drives, three IBM 2314-equivalent disk drives providing 75 million characters of storage, several internal clocks, and a terminal controller capable of controlling 64 terminals of varying types and speeds, of which 32 are installed currently. Although some of the hardware, particularly device controllers, was specially

Portions of the research reported herein were supported by grants from the National Science Foundation and the National Institute of Education, United States Department of Health, Education, and Welfare. The opinions expressed do not necessarily reflect the position or policy of the sponsoring agencies and no official endorsement should be inferred. Jim Pellegrino played a major role in the development of the undergraduate laboratory system described in this paper.

constructed by LRDC, the hardware configuration is "standard" to the extent that an equivalent system could be built today using off-the-shelf, commercially available hardware.

The ETSS operating system was developed in-house and is a general-purpose timesharing system in which user programs are swapped, or transferred, to and from drum memory as required. The current limit on user program size is 16K (18-bit words), and the average user program is in the 12K to 14K range. Scheduling is driven by user input/output requests and by an internal clock, or "watchdog timer," which reschedules any program which attempts to execute continuously for more than 100 msec. The scheduler queue is a FIFO queue (First-In, First-Out) ordered by user priority. Users running experiments are customarily assigned a high priority to provide the responsiveness required in on-line experimentation. This responsiveness is dependent in large part on the fast swapping drum which enables ETSS to swap and execute an average of 7 to 10 users per second.

The system supports multiple languages including FORTRAN, FOCAL, BASIC, ALGOL, and SCAMP (a string-manipulation language). There is a text editor, a debugging system, a linking loader, a statistical package, and a variety of utility systems. Experiment control programs are typically written in FORTRAN which has been augmented in several areas. For example, video-display cursor control commands have been added to the FORMAT statement. A variety of "canned" subroutines are available to perform functions such as response latency calculation and the reading of responses. These serve to further simplify and speed programming, and most programs are developed and debugged within days or weeks. None requires the months of development all too frequently required in an earlier era of assembly language programming and rudimentary, unreliable operating systems.

LABORATORY EXPERIMENTATION WITH A TIMESHARED COMPUTER

Although most laboratory research at LRDC can be classified as dealing with discourse comprehension or other higher level cognitive processes, the experiments that are conducted span most of the methodologies of cognitive psychology. Current experiments include studies of tachistoscopic letter-string perception, prose comprehension speed, running memory for wording during prose comprehension, learning of verbal lists, gathering of protocols for geometry theorem proving, and word categorization speed. Though the experiments differ, they share similar requirements for stimulus generation and control, response timing, and data collection and reduction.

Stimulus Generation and Control

There is general need in a cognitive psychology laboratory for tight control over the timing of stimuli.

This is particularly true for tachistoscopic work but is generally true for experiments in which either memory or response time is measured. We have developed several solutions to the stimulus timing problem to meet different timing needs.

For tachistoscopic work, the use of a video terminal with a mask over part of the screen is an ideal solution. A sequence of displays is composed on a masked portion of the screen and is then scrolled (using line feeds) past a window in the mask (see Fischler, 1976). If the display has several components, they initially are written in a vertical column beneath the window and are raised in turn by sequences of line feeds. On a system such as ETSS, in which long records are transmitted in a continuous burst, nonprinting characters can be placed between line feed characters to produce measured intervals between display segments. This process is illustrated in Figure 1 by a program which will (assuming a 1,200-baud line to the terminal) display A for 1/60 sec, a blank field for 1/60 sec, and a masking stimulus (\$) thereafter, all in the upper left of the screen.

To produce these tachistoscopic effects, it is important to understand the relationship between three asynchronous time lines. These are the timing clock for character transmission to the terminal, the refresh rate of the terminal, and the time required to execute certain terminal functions. In our system, records are transmitted continuously. A 120-character record will take 1 sec, with aperiodicities of less than 100 microsec. As all of our video terminals have a refresh rate of 60 Hz, at a 120-Hz character transmission rate, the screen is refreshed once for every two characters received. Each character is executed by the terminal prior to the arrival of the next character. Therefore, the fundamental time unit for displays is 1/60 sec. In each time unit

```

C      ASSUME LF TO CONTAIN A LINE FEED CHARACTER
C      IN A1 FORMAT; DUMMY TO CONTAIN 000(ASCII)
C      IN A1 FORMAT
C
C      ERASE SCREEN AND HOME UP (LOGICAL UNIT 1
C      IS THE TERMINAL).
C
1000  WRITE (1,1000)
      FORMAT (1X,2CHF)
C
C      WRITE DISPLAYS ON COVERED PORTION OF SCREEN
C
      WRITE (1,1001)
1001  FORMAT (1X,'A'//1X,'$')
C
C      SEND CURSOR TO BOTTOM OF SCREEN
C
      CALL CURTO (24,1)
C
C      DISPLAY STIMULUS
C
      WRITE (1,1002)LF,DUMMY,LF,DUMMY,LF
1002  FORMAT (5A1)
C....

```

Figure 1. Tachistoscopic display routine (FORTRAN) using a 120-character/sec transmission rate.

```

C      DOUBLE INTEGER ARRAY (4), LATEN
C      ACTIVATE RESPONSE TIMING FEATURE
C      CALL PDTIME(ARRAY,, TRUE.)
C      WRITE DISPLAY
C      WRITE (1,1000)
1000  FORMAT (1X,'NOW')
C      READ 1-CHARACTER RESPONSE: IRESP WILL CONTAIN
C      THE 1-CHARACTER RESPONSE
C      CALL CHAR (IRESP)
C      GET TIMES: ARRAY (1) IS START TIME OF WRITE
C      TO TERMINAL, ARRAY (2) IS STOP TIME. ARRAY (3)
C      IS TIME OF FIRST CHARACTER FROM TERMINAL,
C      ARRAY (4) IS TIME OF LAST CARRIAGE RETURN
C      CHARACTER. ALL TIMES ARE IN MILLISECONDS
C      CALL PDTIME (ARRAY,,FALSE.)
C      COMPUTE LATENCY AND STORE IN LATEN
C      LATEN = ARRAY (3) - ARRAY (2)

```

Figure 2. Program to present a display, accept a response, and calculate the response latency.

(1/60 sec), two characters can be received from the computer and executed by the terminal. In Figure 1, one dummy character is needed after the first line feed to assure that the contents of the terminal's memory stay constant for exactly one screen refresh, permitting A to be displayed.

A problem that arises from the restriction of presentation times to integral numbers of refresh cycles is that extremely short displays become impossible. For example, a 10-msec display is not unusual in letter recognition studies; however, the shortest display interval possible with the terminal is 16.7 msec. We solve this problem by controlling screen brightness with a pair of polarizing filters over the window. For many purposes, the information available from a brief display is a function of the product of intensity times duration.

For longer displays, different procedures must be used. For example, ETSS currently transmits records of up to 310 characters so that the maximum display cycle we can control with dummy character timing at 120 characters/sec is $310/120$, or 2.58 sec. For longer intervals, the program writes the display on the screen, asks the operating system to suspend program execution for the desired time interval, and then clears the terminal screen. Since experiment control programs are given priority by ETSS, they resume execution very soon after the suspension interval is complete, giving display precision of ± 5 sec. This is generally adequate for durations that are too long for the dummy character procedure outlined above.

Response Timing

Response timing is a more serious problem in a timesharing system since central processing is partially

asynchronous with terminal activity. As a solution to this problem, the ETSS system records important I/O times in milliseconds whether or not they relate to the program that is currently executing. These times can be recovered later by the experiment control program when it regains access to the CPU. The times available to a program are the start and completion times of both the last record written to the terminal and the last complete or partial record received from it.

As an example, Figure 2 shows a FORTRAN routine that writes the word NOW on the terminal screen and computes the latency of a one-character response.

Using these and related techniques, the ETSS system can handle a broad range of visual presentations of verbal information with keypress responses. Timing of stimulus duration and response latency can be controlled and computed through a combination of several procedures. The procedures we have described depend upon the ability of programs to request that the operating system log the times of inputs and outputs for terminals they control and, on occasion, deny the program access to the CPU for precise time intervals.

Data Collection and Reduction

The ETSS approach to data analysis is to write un-analyzed data to the disk during on-line experiments and perform analysis and reduction tasks independent of data gathering. Because of the priority scheduling scheme in ETSS, data analysis need not compete with on-line experimentation even if both are run at the same time. We can get analyses of a subject's data very soon after he finishes, even though other on-line experiments are in progress. The analysis program running at lower priority can only gain access to the CPU while on-line programs are waiting for I/O to be completed.

The same priority arrangements also permit new programs and stimulus material to be entered into the system without interfering with on-line experiments. They are entered under a low-priority account number and then transferred to the directory of the appropriate on-line number after testing.

CONTROLLING EXPERIMENTAL APPARATUS

To increase the range of stimuli and responses beyond those possible with standard video terminals, we have constructed general-purpose apparatus controllers called "intervening black boxes," or IBBs (Fitzhugh, 1974). The important characteristic of the IBB is that it communicates with the computer using standard ASCII characters. The IBB is connected to any terminal line on the computer and can be controlled with a FORTRAN program using conventional READ and WRITE statements. The IBB will open and close relays, start a millisecond clock when a specified contact is grounded, stop the clock when one or more different contacts are grounded or when a specified time has elapsed, and report back the elapsed time and the specified response that stopped the clock.

One example of use is to present pictorial stimuli with a slide projector and record times for either a vocal or keypress response. In such an application, the IBB is instructed to close a relay to advance the projector, start the clock when a photocell detects a display on the screen, stop the clock when a voice key is activated, and report back to the computer the response latency. Other IBBs now being constructed are more specialized for particular laboratory needs. Future IBBs will be based on inexpensive microprocessors which will provide additional flexibility and which will reduce the complexity of the unit.

THE UNDERGRADUATE LABORATORY

In the past 2 years, we have developed a computer-based laboratory course in cognitive psychology that also operates on the ETSS system. Students participate both as subjects and experimenters in a series of classic paradigms such as Sperling's (1960) partial report procedure, the Clark and Chase (1972) sentence verification task, and the Craik and Tulving (1975) levels of processing work. The students run on video-display terminals (DEC VT-52). After participating in an experiment and, in some cases, watching a computer simulation of their performance, students receive complete, annotated line-printer output that includes all of their trial-by-trial data plus statistics that can serve as the basis of a lab report.

For example, a student using the Clark and Chase module logs on and requests the Clark-Chase program. The program asks for the student's name and then requests a time limit for the run. This feature accommodates late students and other scheduling problems. After displaying a reminder message about response procedures, the program presents as many trials as possible in the requested time period. Each trial uses a randomly selected treatment combination, one of 16 in the experiment. The student presses the return key to see a stimulus and responds by pressing 1 for true or 2 for false. If a mistake is made or if the response takes more than 4 sec, a buzzer sounds and an error message appears. At the conclusion of the experiment, the student receives a line-printer listing showing the response time for each trial sorted by conditions, the mean for each treatment condition, the estimated parameters in the Clark and Chase model, and the predicted means.

As can be seen in this example, the capabilities developed for our on-line research lab are also available to the undergraduate laboratory; students can participate not only in the traditional verbal memory exercises but also in the chronometric and sensory memory procedures that have been of great importance in the growth of cognitive psychology. By writing general programs that have many options, the experimental paradigms can be used for the basic lab course

and also for occasional exercises in advanced courses. For example, our lower level lab students use the Neisser scanning paradigm to study the basics of letter recognition while advanced students can explore whether scan processing is better described as serial exhaustive or serial self-terminating (this paradigm and one other were adapted from the Dartmouth Project COMPuTe, Bewley, 1974). Also, advanced students can compare the several Clark and Chase models rather than studying only the simplest.

The system is sufficiently easy to use so that a colleague teaching a psycholinguistics class was able to send his computer-naïve students to try the Clark and Chase sentence verification task without supervision. After a 15-min introduction to the computer system, all students were able to run themselves as subjects. The sole problem was that one student logged in on the wrong account number resulting in the temporary loss of the line-printer output for his run.

The undergraduate lab uses stimulus control and response measurement capabilities that would not be available otherwise for undergraduate courses, providing a more realistic introduction to experimental psychology. As the ETSS system was already available, the only cost was for additional video terminals (ours were provided through an NSF Instructional Scientific Equipment grant). The undergraduate lab programs also are available to graduate students who find them a useful introduction to on-line experimentation.

CONCLUSION

LRDC has operated an on-line behavioral research laboratory since the mid-1960s. Initially, a DEC PDP-7 was used with a special laboratory control operating system that was developed in-house. After a period of unreliable service, the system was gradually refined and provided adequate service for a number of years. However, programming was restricted to assembly language, and program development and debugging could not take place while experiments were running. This made preparation of an experiment time consuming and discouraged researchers interested in using the computer for short-term studies.

The ETSS system was developed with these shortcomings in mind, and particular attention was paid to program development facilities. The availability of large system features, higher level languages, and the ability to program and debug while experiments are running has altered the way in which the computer is used and has greatly reduced the time required for program preparation. ETSS is used in a routine and casual way, and a computer-based experiment is no longer the major undertaking of 5 or 10 years ago.

At the current time, for example, five new experiments are being programmed, modifications are being made to two others, and four are running. During the

month of October, 1976, ETSS was used for a total of 2,947 terminal h; of this total, 189 h were devoted to on-line laboratory experimentation. This number is expected to double or triple as the academic year proceeds. The system is available 22.5 h/day, and experiments are run whenever subjects are available and without regard for other users who may happen to be using the system at the time. Terminals are located throughout the building for program development and check-out, and six video terminals are reserved specifically for lab experiments.

The undergraduate laboratory course is offered each term by the psychology department. Thirty-two students are enrolled this term, 50 are expected in January, and 100 per term are expected next year. During the month of October, students in the course used the system for 61 terminal h, or about 2 h/student. Four video terminals are used for the computer exercises.

During the hours of 9:00 a.m. to 3:00 p.m. on a normal working day, the system typically supports from 15 to 20 simultaneous users. Two or three may be running laboratory experiments, five or six may be using an experimental information retrieval system, up to four undergraduate laboratory students may be running, and the remainder may be developing new programs, analyzing data, or running the computer-assisted instruction programs that are available. The system is able to meet the timing and control requirements of the laboratory

user while supporting this diverse work load because of the priority scheduling and timing features of the operating system. Our overall experience with ETSS during the past 2 years has demonstrated that on-line experimentation can be conducted successfully on a timeshared computer.

REFERENCES

- BEWLEY, W. L. *Cognitive psychology: A computer-oriented laboratory manual*. Hanover, N.H.: Dartmouth College Bookstore, 1974.
- CLARK, H. H., & CHASE, W. G. On the process of comparing sentences against pictures. *Cognitive Psychology*, 1972, 3, 472-517.
- CRAIK, F. I. M., & TULVING, E. Depth of processing and the retention of words in episodic memory. *Journal of Experimental Psychology: General*, 1975, 104, 268-295.
- FISCHLER, I. Beating the memory drum: Rapid sequential visual presentation (RSVP) on a 4K PDP-8. *Behavior Research Methods & Instrumentation*, 1976, 8, 233-234.
- FITZHUGH, R. J. *A general-purpose time-sharing system for a small- or medium-scale computer*. Pittsburgh, Pa: University of Pittsburgh, Learning Research and Development Center, 1973.
- FITZHUGH, R. J. Laboratory control with a medium-scale time-sharing system. *Behavior Research Methods & Instrumentation*, 1974, 6, 131-137.
- FITZHUGH, R. J., & GLASER, R. A general-purpose computer system for a research and development center. *American Psychologist*, 1975, 30, 219-225.
- SPEERLING, G. The information available in brief visual presentations. *Psychological Monographs*, 1960, 74 (Whole No. 11).