Behavior Research Methods & Instrumentation
1975, Vol. 7 (2), 233-238

SESSION XII SYMPOSIUM: SKED: NEW SYSTEMS, DATA ANALYSIS, APPLICATIONS IN ELECTROPHYSIOLOGY AND HUMAN EXPERIMENTATION, AND DISCUSSION OF FUTURE DEVELOPMENTS

ARTHUR G. SNAPPER, *Western Michigan University, Presider*

A new OS/8 SKED

ARTHUR G. SNAPPER
Western Michigan University, Kalamazoo, Michigan 49001

A new SKED run-time system and compiler have been designed for use under the OS8 operating system. OS8 is a set of programs designed by DEC for the PDP8 computer with 8K or more core memory locations and a mass-storage device (disk, or DEC-tape). The advantages of OS8 include operator convenience, device independent input-output, standard file formats, and convenient program chaining as well as a set of standard data analysis programs. The new compiler, OSCOMP, differs from the previous version in two ways. The first new feature is the ability to process named input and output files on any OS8 compatible peripheral. The second feature is the utilization of 8K of core, permitting compilation of longer state tables than could be processed with the earlier version. Furthermore, with a disk as the OS8 peripheral, the compilation process is essentially instantaneous, for state tables previously requiring from 3-30 min with paper tape devices. The new run-time system, OSRTS8, contains a variety of new features. The most important improvements are the abilities to record data on the OS8 peripheral as well as to read state tables stored as files on the mass-storage device. Other new features include chaining of state tables, automatic start, automatic output file specification, and capability for as many as 12 simultaneous stations.

Consider the following hypothetical situation. Behavioral pharmacologist Dr. Shute Emup, of High Laboratories and Pharmaceuticals, has completed a series of experiments. He has been studying the effects of a number of behaviorally active compounds on discrimination learning in the albino rat. His laboratory consists of 96 rat boxes, each of which is programmed by traditional relay circuitry. It took him only 6 months, several years earlier, to set up the discrimination procedure in which he was interested, and he has continued to study it under a variety of drugs administered to the rats. He recently has discovered that one compound improves the acquisition of the discrimination and that the improvement can be replicated repeatedly. Dr. Emup now realizes that he may have an important drug for which behavioral alchemists have been searching for many years, a learning facilitator.

This work was supported in part by Research Scientist Development Award K2-MH-70483 from the National Institute of Mental Health.

However, Dr. Emup is a careful scientist. He knows that the improvement in performance on the discrimination procedure may have been caused in many ways. The rats really may be super intellects or maybe they are just more hungry than usual (the world does not need a drug that increases appetite). Other possibilities occur to Dr. Emup. Possibly the drug increases the attentive capacities of the rat. Possibly the drug reduces the effects of distracting stimuli. Perhaps the drug increases the ability of the rats to hear the clicks of the relay programming equipment as the reinforcement condition is set up.

Dr. Emup immediately designs a research program to test all of the different possible mechanisms of action of his new drug. He finally arrives at six different experiments, each designed to test for one of the

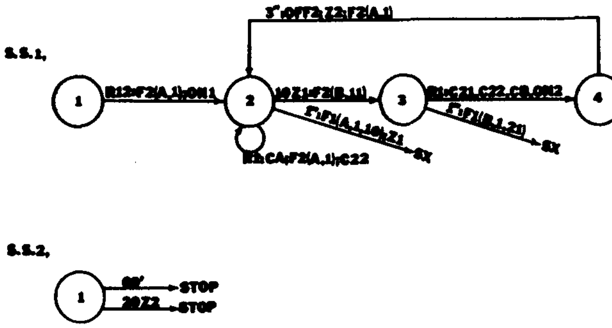

Figure 1. State diagram of DRL schedule.

different possibilities by a careful analysis of the behaviors involved in his original experiment. Fortunately, Dr. Emup is able to obtain $20,000 to purchase a computer system for conducting the different studies. He buys a 12K PDP8e with a disk and a Teletype, as well as an interface to connect the computer to his rat boxes. He also discovers SKED, a system to simplify the programming of the experimental control and data acquisition capacities of the computer.

While waiting for the delivery of his computer (90 days), he completes reading the programmed text describing the state notation language used to program the computer, and he reads the manuals describing the operation of the system. He is able to design completely the program for the six experiments, and when the equipment arrives he immediately connects the computer to his experimental stations. Each rat box holds its occupant, and serves both as a living cage and as a test chamber. Each experiment is designed to run for 3 h per day, and it is possible to conduct each experiment in 12 stations simultaneously. The data produced by each experiment is to be analyzed and plotted on a graph on the Teletype.

The daily routine of conducting the six experiments now begins. Each morning Dr. Emup comes to his lab and looks over graphs produced from the day before. He sits at the computer and reanalyzes some of the data according to the current status of the experiment. For 6 h every day, the equipment is free so each rat can be weighed, fed, and otherwise attended to, and the chamber can be tested for correct operation of the manipulanda and stimulus generators.

Dr. Emup also has time at the computer to develop programs for future experiments. As he learns the SKED system, he finds that it takes only a few hours to design a complex experiment involving concurrent chains and sequential analysis of the data produced by these procedures. Before he leaves his laboratory every night, Dr. Emup types a few simple statements into the computer terminal, and the series of six different experiments on 96 subjects begins. The next morning when he comes into the laboratory, yesterday's data once again are waiting for him in graphic form on the Teletype.

Although this story may seem like science fiction, the development of an OS8 SKED makes it possible to conduct research in exactly this fashion as the normal operation of a large laboratory.

SKED is a system of software and hardware designed to permit rapid and convenient programming of the PDP8 computer family for the purpose of acquiring data from and controlling behavioral experiments. The system is based on the state notation language that provides a method for easily describing the sequential features of reinforcement schedules and data acquisition functions (Snapper, Knapp, & Kushner, 1970).

DRL PROGRAM

An example of a state diagram is shown in Figure 1. The schedule of this state graph is DRL (differential reinforcement of low rate, Ferster & Skinner, 1957) in which the subject must wait for 10 sec after a reinforcement for the opportunity to earn a second reinforcement. Early responses reset the critical delay and initiate a new 10-sec interval. The state graph also includes notation of a distribution of unreinforced IRTs (interresponse times) and a distribution of reinforced IRTs, each requiring 10 recording counters.

The state graph may be read in this way: The experiment starts with an R12 response (pushbutton closure by the experimenter) at which time a houselight is turned on, as indicated by ON1, and State 2 is entered. Every 1 sec in State 2 a timer advances a variable recording counter and generates a Z1 pulse. If 10 Z1 pulses occur in State 2 without being interrupted by a response of the subject (R1), State 3 is entered. If, on the other hand, an R1 occurs in State 2 the timer is reset and a recording of the R1 in the IRT distribution of unreinforced responses is made. In State 3 the variable recording counter number is incremented every 1 sec. When an R1 occurs, it is recorded in the distribution of reinforced IRTs and a 3-sec reinforcement is presented as is shown in State 4. State Set 2 terminates the session after 60 min or 20 reinforcements, whichever comes first.

PROGRAMS TO DEVELOP THE EXPERIMENT, CONDUCT THE EXPERIMENT, AND ANALYZE THE DATA

Once the state diagram of the desired experiment has been developed, a number of steps are necessary to actually conduct the study and acquire the data specified in the diagram. The first step is to type the diagram in a form shown in Table 1. This state table was produced with the help of a computer program known as the Editor, which permits corrections of typographical and other errors through the use of some simple commands, which are typed on the same terminal on which the text is entered into the computer. The

Table 1
State Table of DRL Schedule

```
/THIS IS BENCH1.PA
/BENCHMARK STATE TABLE
/DRL WITH IRT DISTRIBUTION
/ON1=HOUSELIGHT
/ON2=SR
/R12=SESSION START
/R1=LEVER PRESS
/SESSION LENGTH 60' OR 20 SR'S
/SETUP FOR DRL 10"
/IRT BINS 1"
/COUNTERS 1-10 EQUAL IRT'S LESS THAN CRITERION
/C11-21 ARE REINFORCED IRT'S
/C22=TOTAL R'S
/C23=TOTAL SR'S
/REQUEST 23 COUNTERS

S.S.1,
S1,
        R12:F2(A,1);ON1--->S2/START SESSION FROM TTY
S2,
        1":F1(A,1,10);Z1--->SX/INCREMENT IRT DIST.
        10Z1:F2(B,11)----->S3/SET UP SR AFTER 10"
        R1:CA;F2(A,1);C22---->S2/RECORD IRT, RESET TIMER
S3,
        1":F1(B,1,21)---->SX/INCREMENT B TO 21"
        R1:C21;C22;CB;ON2---->S4
S4,
        3":OFF2;Z2;F2(A,1)--->S2
S.S.2,
S1,
        60'--->STOP
        20Z2---->STOP
$
```

corrected version may then be stored on one of a variety of media that can be accessed by the computer, e.g., paper tape, magnetic tape, or magnetic disk. The output file of the Editor then serves as the input file to a program called the SKED Compiler that translates the state table into a format that is usable by a program called the Run-Time System (RTS).

The RTS program is an executive routine that enables as many as 12 compiled, independent state tables to control different experiments at the same time. The RTS has a series of commands for loading the state tables into different stations, starting the experiments, monitoring the results, modifying parameters and producing an output file containing the data acquired by the experiment. In most cases, after the data has been collected, the experimenter will also desire to use the calculating power of the computer to analyze the data and to produce tabulated or graphical summaries of the raw data. The data analysis is usually conducted by means of a mathematical programming language such as FORTRAN, BASIC, or FOCAL (Snapper, Lee, Burczyk, & C. Simoes-Fontes, 1974).

ORIGINAL SKED SYSTEMS

The SKED system was originally developed in 1966 when the cost of the minicomputer and the associated peripheral devices was much greater than today's cost. The original version of the system was designed for paper tape (high or low speed) as the media on which programs, state tables, and data were stored. Later versions of the system were designed to be used with a small inexpensive magnetic cartridge system, the Tennecomp 1375. This peripheral, which permitted loading and recording programs as fast as high-speed paper tape devices, increased the convenience and decreased the time necessary for program development and for the operations necessary to conduct the experiment and analyze the data.

In the sequence of operations that produce the state table, conduct the experiment, and analyze the data, it is necessary to read four separate programs into the computer: (1) the Editor, (2) the Compiler, (3) the RTS, and (4) the data analysis program (i.e., FOCAL). Furthermore, each of these programs requires one or more input files and each produces one or more output files for use by other programs. Each time one of the programs is entered into the computer, or when it needs an input file or produces an output file, the speed of the peripheral used for these purposes becomes the major factor in the productivity and convenience of the entire system. Using low-speed paper tape, the time needed to read the files can be as long as 30 min.

The limitation of paper tape peripherals is not unique to the SKED system but extends to all types of computer usage. For this reason, computer suppliers have developed a variety of different media for storing and loading program and data files. These media include magnetic tape, magnetic disk, magnetic disk cartridges, cassette systems, and the newer floppy-disk systems. Each of these media has different advantages and disadvantages in terms of convenience, cost, capacity, and speed.

DEC has produced an operating system, OS8, to permit the use of a variety of these media with the PDP8 in an organized fashion. The OS8 operating system is used in the same way for all of the different devices that can support it. That is, DEC-tape, disk, or magnetic tape can be used to run OS8 programs without the necessity for developing new programs for each different device. In the past, the most efficient way to use a new mass-storage peripheral (for example, the Tennecomp system described earlier) was to revise the Editor, Compiler, RTS, and data analysis packages for use with the particular device. The OS8 system, on the other hand, treats all devices in a similar fashion, so that any program that can use one of the devices, can use them all with only the name of the device changed in the specific program.

THE OS8 SYSTEM

OS8 organizes the files contained on the mass-storage peripheral and provides a variety of programs that permit the files to be manipulated, listed, and stored in a directory under a specific name. For example, to use the Editor under OS8, the operator only needs to type RUN EDITOR on the terminal. Input and output files also can be specified with a simple name. To produce a new state

Table 2
Comparison of Program and Data Transfer Operations of the SKED System

Operation	Slow Tape (Sec)	Fast Tape (Sec)	T* (Sec)	OS8 Disk (Sec)	OS8 DEC-Tape** (Sec)
Load Editor	220	7.3	26	1	13
Load Table	280	9.3	26	2	19
Output Table	280	46.6	26	2	19
Load Compiler	500	16.6	26	2	27
Pass 1	280	9.3	26	2	20
Pass 2	390	13.0	19	2	25
Load RTS	500	16.6	26	2	39
Load Bin Table	200	6.6	200†	2	120
Output Table	30	5.0	1	1	54
Load FOCAL	820	27.3	52	1	15
Load Analysis Program	250	8.3	250†	1	27
Load Data	30	1.0	26	1	32

**Tennecomp*
***The speed of DEC-tape operations depends on the locations of files on the tape.*
†The slow-speed paper tape reader is assumed for these operations, but the Tennecomp can be made faster under some circumstances. For example, the state table must be loaded with paper tape reader only once by the RTS. Then the RTS and tables may be recorded together on Tennecomp so that the daily loading of the RTS and binary programs requires only 52 sec. Also, the data analysis program can be recorded with FOCAL, reducing the time to load both to 52 sec.

table with the Editor with the name TABLE.PA which will then be stored on the mass-storage device, requires only the command CREATE TABLE.PA. This command will automatically call the Editor and enter a new file name TABLE.PA in the system directory while deleting any earlier file with the same name.

After the program has been typed and saved on the system device, it can be called for use by the Compiler by name, and the Compiler output file can also be stored by name. The RTS then can load the named state table into the computer for the stations by which it is to be used, with another simple set of commands. Thus, the OS8 system increases the convenience of the computer for preparing programs, conducting experiments, and recording the data of behavioral studies.

VIRTUAL MEMORY WITH OS8

OS8 and the associated mass-storage device also provide a number of new options that increase the power of the computer. For example, one of the most frustrating limitations of data analysis programs on the PDP8 is the limited size of the memory. Under FOCAL, only a few program statements and variables produced from the data can be contained in the computer at one time. Other data analysis languages have even more restrictions on the use of space. This limitation means that it may be impossible to program a comparison of the IRT distributions produced so easily in the state table of the sample experiment. The lack of core space will make it impossible to hold data from more than one session at a time and still have room for a meaningful program. The only strategy for overcoming this limitation is to break up the data analysis into small steps. The first step might collect the data from one distribution and then calculate the mean and variance of the distribution, and finally punch these numbers on paper tape to permit room for the data from another session or subject. After all the data from all the sessions has been analyzed, then the paper tape containing the intermediate results might be read by the computer under the control of another FOCAL program to conduct a statistical test or plot a graph.

OS8 uses a similar strategy to conduct long and complex calculations on data produced by SKED. The operations of the program might also be broken down into small steps. After one stage of the analysis was complete, the program would automatically load the next stage of analysis and also store the output of the preceding stage as a named file. Programs that use the intermediate results produced by earlier stages of the analysis routine then can call the output file of the previous programs as an input file for the current program.

The critical feature of this procedure to the user of the system is that it proceeds automatically. With OS8, it is no longer necessary for the operator of the computer to load the successive data analysis programs or to handle the intermediate files. The computer program takes care of these cumbersome and time-consuming loading and filing operations.

FURTHER ADVANTAGES OF OS8 SKED

SKED has been rewritten to use the OS8 system for rapid and convenient manipulation of input and output files. The first advantage that accrues to the user of OS8 SKED is the ability to produce named files from the Editor and to process these files using the Compiler. The speed of this and other operations of the SKED system is contrasted in Table 2 with the time required to complete the same operations with the older paper tape and Tennecomp systems.

Although the high-speed paper tape system and the Tennecomp system are approximately the same speed as the OS8 DEC-tape system, the convenience and other features of OS8 make the latter much preferable to the former. The Disk OS8 system is much faster than all the others, making this option the most desirable for the user.

The Disk and the TC08 DEC-tape systems share the ability to exchange data with the memory of the computer through data break. This feature means that data can be transferred into and out of the computer without slowing the response of the system to the 110-msec clock associated with the RTS. The TD8E DEC-tape, on the other hand, requires total attention of the computer program for data transfer. The result of this is that the TD8E may not be used to load state tables or to record data from the RTS without disturbing the basic timing of the computer, while the Disk and TCO8 permit these operations to occur while experiments are being conducted.

For the two data-break options under the new OS8 SKED, it is possible to create an output file at any time or under any condition from a simple state table command. This feature then allows the experimenter to collect a large quantity of data from an experiment without restrictions due to core storage. For example, we are currently collecting every sequential interresponse time and duration of leverpress from each of 12 subjects running concurrently in the same session. Each subject may produce as many as 3000 responses during the session. If the data were to be retained in core, 6000 core locations would be required to store the data from a single session and only one station could be used at a time with a 12K computer. The solution to this problem is to record 100 sequential IRTs and durations in core and then to request that the data be transferred to the disk as part of an output file. Since the program was designed to permit new data to be acquired while the output file is being produced, this scheme then permits the operation of 12 sections at the same time, each of which can transmit data to the disk for later data analysis.

The option of recording every event is very powerful for answering some experimental questions concerning the nature of sequential dependencies in reinforcement schedules. However, this operation is not the standard method of recording data. The state table can be made to sort IRTs into distributions as in the example of Figure 1. In this case, since the quantity of data for each station is relatively low, it is possible to store the data for the entire session in core memory and to create an output file at the end of the session.

If all 12 stations are being used to conduct experiments that start and stop at the same time, then the TD8D DEC-tape option is as useful as the data-break devices for loading state tables and for recording the data files. On the other hand, if some experimental stations start and stop at different times and other programs are to be loaded into these stations to utilize the full capacity of the laboratory, then the TD8E cannot be used to load and store data from experiments that finish when other stations are still in progress. Once again, this would disrupt the timing of the stations in operation while the data transfer takes place. However, the data-break OS8 peripherals will permit stations to be started and stopped without interference with other stations.

CHAINING STATE TABLES

Another feature of the data-break options of OS8 has been incorporated in the OS8 SKED to increase the automation of the behavioral laboratory. Consider the case in which the laboratory has more than one set of 12 experimental stations. In this case it might be reasonable to use each station as both a living and a testing cage. For example, six sets of 12 stations could be used in this way. Each set of stations are to be used for daily 3-h sessions, and the remaining time is to be used for program development and data analysis of the results produced in the preceding 18 h. To permit the experiments to be specified only once each day, a chaining F3 (Snapper & Hamilton, 1974) has been incorporated as a new function of the system. The experimenter needs only to specify the list of six separate binary state tables once each day of the system.

The first experiment will then be started. When it is complete, it will automatically produce the output file containing the data from the current session. In the state table of the experiment is an F3 function that then initiates the necessary steps required to load and start the next experiment for the other stations. Each experiment, until the last, then calls in the next experiment and records the data. The final experiment then closes the output file, returns to the OS8 batch monitor and then proceeds to operate the data analysis routines which produce summary statistics and graphs of the statistics for the preceding sessions.

Although this type of automation may seem overly elaborate and expensive, since the same result could be obtained by having the researcher use the same 12 stations for the sequential experiments, manually removing the subjects for each session, it could be cost justified. For example, if each of the experimental stations contained a Rhesus monkey who lived in and was tested in a restraint chamber, the procedure described here would not be unreasonable. In other laboratories the stations used in different experiments might contain different species of subjects or different manipulanda, stimulus generators or reinforcement devices. The major advantage of this strategy would be that the experimenter would have access to the computer during his normal office hours for producing new programs for reanalyzing data. The actual experiments would be conducted automatically outside of office hours.

DISADVANTAGES OF THE OS8 SKED

There are some problems in the OS8 SKED, some of which will be corrected by further programming. The first difficulty is the amount of core space required for calling the OS8 subroutines. In the most flexible configuration of SKED, the RTS loses nine pages of core (1152 decimal locations) to these OS8 programs. Under some circumstances as many as three of these pages can be recovered for state table storage by relinquishing some of the features of SKED. Of course, if it is necessary, the older paper tape version of SKED can be used to run longer state tables, but much of the convenience and speed of the system will then be lost.

For the case in which 12 or less stations are to be run concurrently and the output files for the data are to be created at the end of the experiment, it would be possible to develop a new SKED loading and recording program that would then be replaced by the actual RTS during the session. First, the state tables would be passed through the loader to determine their core locations. Next, OS8 programs would be used to merge the state tables and an RTS without a loading program in it. After the session, the RTS would be replaced by a separate program that would create the output file. In this way, the subroutines within the RTS for loading and dumping could be replaced by state tables, saving all nine pages of memory.

The second problem of the system arises from the file structure of OS8. The only convenient file types are ASCII and binary formats. The RTS produces ASCII output files of the recording counters thus requiring six characters per counter. The current data analysis packages can read these ASCII files, but a new file type that permitted integer files with one location per recording counter would shorten the time for creating the output files and shorten their length by a factor of four. Furthermore, data analysis would proceed more rapidly on this type of file. Plans have been made to develop files having this option for the OS8 system.

The final problem with the current version of OS8 is that it is impossible to run OS8 programs as background programs with the RTS responding to the 10-msec clock in the foreground. If one had a computer with 20K or more of memory, it should be possible to conduct data analysis and to program new experiments while the RTS was not busy. It may be possible to develop this option through the use of a new software system developed by DEC called RTS8.

Detailed description of the operation of the OS8 SKED routines is available from the SKED USERS GROUP, Psychology Department, Western Michigan University, Kalamazoo, Michigan 49001. Membership in this group provides a newsletter, manuals, and the programs necessary for running the SKED system.

REFERENCES

Ferster, C. B., & Skinner, B. F. *Schedules of reinforcement.* New York: Appleton-Century-Crofts, 1957.

Snapper, A. G., Knapp, J., & Kushner, H. Mathematical description of schedules of reinforcement. In W. N. Schoenfeld (Ed.), *The theory of reinforcement schedules.* New York: Appleton-Century-Crofts, 1970. Pp. 245-247.

Snapper, A., & Hamilton, B. Programming special functions in the SKED system. Behavior Research Methods & Instrumentation, 1974, 6, 173-176.

Snapper, A., Lee, D., Burczyk, L., & C. Simoes-Fontes, J. FOCAL, FORTRAN, and BASIC programs for reformatting and analyzing data collected by the SKED program. Behavior Research Methods & Instrumentation, 1974, 6, 176-180.