

SKED systems: Paper tape, Tennecomp, and OS/8

KEN STEPHENS and AMY BARMEIER
Western Michigan University, Kalamazoo, Michigan 49008

The SKED system has been developed at different levels of complexity and power for differing hardware configurations. The simplest of these, and the least expensive, is the paper-tape SKED system. Some of the drawbacks associated with the use of paper tape are absent in the Tennecomp magnetic tape cartridge system. Users with mass storage capabilities can make use of the OS/8 operating system (copyright DEC) to speed up all aspects of the SKED system (especially editing, compilation, and loading of programs), and take advantage of a file-oriented system structure for data storage and analysis.

SKED is a process control language for precise control of psychological experimentation. It is specifically written for any model of computer from the PDP-8 family,¹ and exists in several forms for differing configurations of hardware options and software support systems. Hardware differences can be categorized according to processors, peripheral devices, and interfaces. Versions of SKED have been written for the PDP-8 and the PDP-12 to accommodate as little as 4K of core memory or as much as 32K.

Peripheral options include input/output (I/O) devices and mass storage devices. The minimum configuration must have some method of entering programs to core, such as low-speed Teletype paper-tape reader, some method of interacting with the programs, such as the Teletype, and some method of producing output, such as low-speed tape punch. However, other peripheral devices can greatly facilitate each of these three I/O functions. For example, a high-speed paper-tape reader and high-speed tape punch greatly speed up the input and output, and other terminals such as the DECwriter or CRTs improve the interactive I/O functions. The use of a Tennecomp cartridge system² as a peripheral is one step above a paper-tape configuration in terms of speed of interactions. The Tennecomp may be used as an input/output device in a manner directly analogous to paper-tape peripherals. Input programs may be read from the Tennecomp, and output (data, state tables) may be written directly onto the cartridges. A variety of mass storage devices have been adapted to the PDP-8 family, and all of these may be used with SKED to improve the program storage and loading processes. These storage devices include disk systems (RK8 and RK8E from DEC and the Diablo disk have all been used), floppy disks (from DEC, Sykes, or Xebec), and magnetic tape (the DEC TD8E and TC08 DEctape systems).

A critical feature of any system is the interface connecting the processor with various possible experimental

stations. State Systems, Inc.³ markets a low-price interface specifically designed for use in conjunction with SKED. The DEC UDC and DK8E interfaces are also good general purpose interfaces which can be adapted for use with a SKED system. It is also possible to construct an interface out of standard digital logic equipment.

SKED software varies accordingly to accommodate each of the possible hardware configurations, but always consists of a SKED compiler and a SKED run-time system (RTS). Other support software, such as an editor program, is necessary for all systems. However, for sophisticated hardware configurations such as those which include a mass storage device, operating systems and time-sharing systems (e.g., OS/8 and RTS-8¹) add power and convenience for the SKED user.

A BEHAVIORAL ANALYSIS OF PAPER-TAPE, TENNECOMP, AND OS/8 SYSTEMS

Dingler, Kadden, and Snapper (1975) briefly described the steps necessary for operation of the SKED system, which are: state diagramming the experimental procedure, translating the state diagram into a state table, creating an ASCII version of the state table using an editor program, compiling the ASCII state table and production of a binary tape, debugging the state table, and loading the binary into, and operation of, the run-time system.

Although these six steps are necessary for all hardware configurations, the operations are different in terms of response effort for the user and immediacy of reinforcement, or time taken by the computer system to complete each operation. In this paper we shall compare paper-tape, Tennecomp, and OS/8 systems in terms of these critical user variables.

Generation of State Graph

The first step toward ultimately running on the SKED systems is to translate a verbal description of an experimental procedure into a state graph. The nota-

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

tional system used for this is identical for paper-tape, Tennecomp, and OS/8 systems (Lee, Stephens, & Duncan, 1974). In all cases, the final product is an unambiguous diagram descriptive of the procedure to be implemented.

Translation of State Graph into Linear Computer Format

Once the experimental procedure has been correctly diagrammed, the state graph must be translated into a linear format understandable by the computer. This is accomplished by using an editor program, and the resulting product is referred to as a state table. The behavior of typing in the state table will be equivalent for all three systems: however, they differ in the speed required to input the editor program into core and output the ASCII product of this typing. The paper-tape system requires that a paper tape of the SYMBOLIC EDITOR be read in, either with the low-speed Teletype reader or a high-speed reader, and then started at 0200, which requires some interaction with the switch register of the PDP-8. The output must then be punched, either on the low-speed Teletype punch or a faster tape punch. With Tennecomp, the SYMBOLIC EDITOR is read in from a Tennecomp tape cartridge, which requires about the same amount of time as the high-speed tape reader. Then the output is recorded onto one track of an empty tape cartridge. Both of these operations with Tennecomp require less than a minute. With the OS/8 system, the user has a choice of editor programs: a file-oriented version of SYMBOLIC EDITOR, called EDIT, and the programmable editor language TECO, which offers greater power and speed to the accomplished programmer. The editor of choice is simply called into core from the mass storage peripheral device by a command to the OS/8 keyboard monitor (e.g., R EDIT), and a command to the OS/8 command decoder then specifies the name of the input file (if there is one) and the name of the output file (i.e., *TABLE1.PA < TABLE.PA). With the concise command language (CCL) option, both of the preceding commands can be given at once (e.g., EDIT TABLE1.PA < TABLE.PA). Depending on which peripheral devices are used, this step can take from 2 to 20 sec.

Compiling

Once an ASCII version of the state table has been produced, this state table must be compiled into a binary program. Although there is a version of the SKED compiler program available for each of the three systems, the function of these compilers is comparable and, in all cases, a binary program is the desired output. With a paper-tape system, a tape of the compiler must first be read into core using a paper-tape reader. A high-speed reader will greatly speed up this process. The compilation consists of a two-stage procedure, thus the ASCII state table from the editor program must then be read into core twice. After the first pass, the user enters the number of counters required for his experiment. After the ASCII tape is read in for the second pass, a

machine language object tape (binary) is punched. This entire process can take up to 10 min, depending on the length of the state table.

The compiler may also be stored directly on a Tennecomp cartridge and read into core from the cartridge. Once the compiler has been read in, the cartridge containing the state table is read into the computer. The user must then enter the number of counters he requires and depress the "CONTINUE" switch to initiate the second reading of the state table. The output from the compiler, if one is using a Tennecomp system, is a binary paper tape which will later be loaded into the RTS and the two saved together on a tape cartridge. Thus, the major advantage of this system over a paper-tape system concerns the time saved when reading the compiler and state table into core. With the OS/8 system, the user calls in the compiler (e.g., R OSCOMP) and specifies the input file (the ASCII state table) or the device on which input is to be found, and the output file (the binary object file) or the device to which the binary will be output (i.e., FILE.BN < FILE.PA, FILE.BN < PTR:, or PTP: < FILE.PA). The number of counters does not have to be specified with OSCOMP if the correct number follows the first "=" in the source file. Compilation with OS/8 SKED, then, requires a minimum of user interaction and takes only a few seconds.

Debugging

From experience, we have determined that it is the exception, rather than the rule, for an ASCII state table to be error free. Each of the SKED compilers, upon encountering an error in syntax or format, informs one of this error by typing on the Teletype both the line of the ASCII state table in which the error was found and a four-digit error code. There are error-code diagnostic lists for determining the exact nature of the unacceptable ASCII input. This process is identical for the three SKED systems, but the procedures involved in correcting the error(s) are markedly different.

The differences between a high- or low-speed paper-tape reader becomes critical at this stage if one is using a paper-tape system, since debugging the state table is largely a matter of input/output time. If an error was found by the compiler, one must read in the editor program, read in the ASCII state table, make the necessary corrections on the state table, punch a new source tape, read in the compiler, run the two passes, and punch a new binary object tape. This whole process can be extremely time consuming because of the time necessary to read in each of the programs and can take up to 30 min.

The sequence of behaviors required to correct state table errors are identical for a Tennecomp system. However, with this system, the input/output time for the editor and compiler programs is considerably reduced because of the increased speed of Tennecomp cartridges over paper-tape readers, thus somewhat reducing the response cost involved.

When the OS/8 SKED compiler finds an error, it

prints the same type of error diagnostic as the other systems, and then exits to control by the keyboard monitor for a new EDIT or TECO command, with specification of input and output files. However, if either editor program was used immediately before compilation, the file specification step will automatically be set up, as CCL will recall the latest used file arguments, just by typing "EDIT" or "TE." Then the correction can be made quickly and control returned to keyboard monitor for the R OSCOMP step. In most cases, this will occur in less than a minute after the first error was found.

Loading and Other Functions of the Run-Time System

Once an error-free binary version of the desired state table has been produced, one must first load the appropriate SKED run-time system into core and then load the binary program for the desired experimental chambers. This is an extremely tedious process on a paper-tape system, since the run-time system may take 30 min to read in at low speed. There always exists the possibility of a checksum error and, if this occurs, the entire program must then be read in again. The process is somewhat facilitated if the input device is a Tennecomp cartridge, due to the increased speed of this device over a low-speed reader. With OS/8, the run-time system and state table binaries exist as files on the systems device and can be quickly and easily called in. In addition, for the user who regularly uses one set of state tables, these can be loaded together with the RTS and saved, and later called into use as one program.

DATA OPERATION

Data Collection

When the state tables are compiled, memory locations are set aside as counters. The strategies for data collection with the SKED system have been described extensively elsewhere (Hamilton, 1975), but these will not vary between the various hardware configurations. The method of dumping data from core does differ, depending on the transfer medium available. With the simplest systems, data must be punched on a low- or high-speed tape punch, which is time consuming. This dumping can be done by manually typing P and turning on the punch, or by leaving the punch on and having DODUM statements in the state table determine when the dump takes place. With Tennecomp, the process is similar, except that the statement in the state table which can automatically cause the dump is called TODUM, and it can only dump four stations at a time. The OS/8 run-time system dumps data into an output file which is stored on a peripheral storage device. The dump can be initiated manually, by a DODUM statement, or (at the user's option) when the last active station finishes, it may dump automatically.

Data Storage

Once data have been conveniently collected by the SKED run-time system, provisions must be made for

their failsafe storage. Paper tape is an extremely poor method for long-term storage, as it is subject to the ravages of coffee spills, poor folding, inaccurate labeling, or disastrous rips. Additionally, if an experiment has been in process for many months and each day's data meticulously saved, the experimenter may find his laboratory piled high with reams of paper tape.

The use of a Tennecomp cartridge for a long-term data storage also is far from optimum practicality. Although Tennecomp Systems makes cartridges ranging from 4K to 64K, these cartridges generally contain only four tracks. Most experimenters find it convenient to store 1 day's data in each of the four tracks, and thus an experiment lasting for 1 year would involve the use of nearly 100 Tennecomps for data storage alone! These tapes are also vulnerable to the same sorts of unplanned physical accidents as paper tapes. In the OS/8 system, a great deal of data can reside on the mass storage medium. However, even with these capabilities, disks and DECtapes fill up with ASCII data. A more convenient data format has been developed which takes only one-fourth as much space as ASCII format. The use of this stripped-down ASCII, or integer format, requires special programs for transfer, and translation back to ASCII for data analysis purposes. In the long run, it is best to transfer data to a permanent storage facility on a larger computer system, since all magnetic storage media are subject to their own flaws, such as loss of the format, physical damage, etc., which can cause loss of all files on the structure.

Data Analysis

The differences in ease of data analysis routines for paper-tape, Tennecomp, and OS/8 SKED systems are due largely to the discrepancies of input/output times for the data and the analysis programs associated with each system. For example, a paper-tape system requires that the FOCAL (1969) language program be read into core to begin. The experimenter must then read in paper tapes of his specific analysis program and data. All results of the analysis are subsequently punched or sent out via the Teletype. In any event, response cost involved in the entire procedure is high, laborious, and time consuming. Availability of either a high-speed reader/punch or a Tennecomp cartridge markedly accelerates the entire procedure by reducing the I/O times. Although this excess time is somewhat eliminated, the experimenter is still forced to engage in the chain of behaviors. There are several data analysis options open to the OS/8 user: a special version of FOCAL, BASIC, FORTRAN II, and even FORTRAN IV. These languages can all be stored on the systems device and called into core by keyboard commands. Then the specific analysis programs written in these languages can be called up and run with all the core-saving options of internal chaining. Of course, without the proper time-sharing software, this analysis must be done while the computer is not controlling experimentation. Some users prefer to deposit the data on some portable storage

medium, such as DECTape or floppy disk, and then transfer for analysis on a PDP-11 or larger computer. However, the new time-sharing SKED system (Snapper, 1975) permits analysis of data via BASIC while experimentation is ongoing.

CONCLUSION

The preceding discussion has presented a behavioral comparison of paper-tape, Tennecomp, and OS/8 SKED systems regarding the responses necessary to design an experimental procedure, implement the procedure on a SKED system, and collect and analyze the actual data produced by any experimental subject. It is obvious that the highest response cost as reflected by actual time spent is associated with a low-speed paper-tape system. The availability of a Tennecomp device reduces a great deal of the input/output times for various operations, but does not actually alter the number of significant experimenter behaviors required. There is a significant response cost involved in learning the OS/8 system, but the advantages of having a file-oriented system seem to outweigh this factor. Because the OS/8 system requires

at least one peripheral storage device and the support software from DEC, it is more expensive, but the added power and flexibility make it a much more attractive system than paper-tape or Tennecomp systems.

REFERENCES

- DINGLER, R., KADDEN, R. M., & SNAPPER, A. G. An introduction to state notation and SKED. *Behavior Research Methods & Instrumentation*. 1976, 8, 69-72.
- LEE, D. M., STEPHENS, K. R., & DUNCAN, M. D. *The notational system*. Kalamazoo, Mich: SKED Users Group, 1974.
- HAMILTON, B. E. Data-collection strategies in SKED. *Behavior Research Methods & Instrumentation*, 1975, 7, 243-247.
- SNAPPER, A. G. Time-sharing SKED and OS/8: A new version. *Behavior Research Methods & Instrumentation*. 1976, 8, 80-82.

NOTES

1. Copyright Digital Equipment Corporation, Maynard, Massachusetts.
2. Copyright Tennecomp Systems, Oak Ridge, Tennessee.
3. State Systems, Inc., P. O. Box 2215, Kalamazoo, Michigan.