SESSION VIII CONTRIBUTED PAPERS: SYSTEMS AND TECHNIQUES

N. JOHN CASTELLAN, *Indiana University, President*

A Simple Information Management system for use in on-line experimentation

H. J. DURRETT, JR.

*Computer Laboratory for Instruction in Psychological Research
University of Colorado, Boulder, Colorado 80302*

The hardware and software capabilities of the modern on-line laboratory permit the development of increasingly complex experimental procedures. These procedures often involve the collection of data from many participants over extended periods of time. The procedures may require the conditional presentation of experimental tasks which are defined by numerous parameters or determined in part by previous participant behavior. The generation, organization, retrieval, and analysis as well as the documentation and monitoring of these experimental procedures often require a source of on-line information. A Simple Information Management (SIM) system for use in on-line environments is one means of providing this information. The SIM system has been used to implement experimental procedures for the study of human problem solving and has general properties which make it of interest.

INFORMATION REQUIREMENTS IN THE ON-LINE LABORATORY

The Computer Laboratory for Instruction in Psychological Research (CLIPR) has a configuration which includes several direct access devices that make possible the rapid on-line storage and retrieval of information (see Bailey & Polson, 1975, for a complete description). This capability has been productively used to the advantage of the overall resources of the laboratory in several systems. A system employing a batch generation program in conjunction with a real-time interpreter program has been previously discussed (Durrett, 1974). A system which exploits the direct access device capabilities to support a Simple Information Management (SIM) system is described in this paper. The system was developed to implement complex experimental procedures designed to study human problem solving behavior. Information management systems are widely

used but only recently have been developed for the on-line laboratory.

Researchers have designed specialized languages (e.g., SCAT and SKED; Polson, 1973; Kadden, 1974) and systems for the analysis of data (Restle & Shaffer, 1974) which do minimize some of the information problems of on-line experimentation. However, many of the information requirements of the on-line laboratory begin before programs are written or before data have been collected. Thus, programming languages and data analysis systems are only a part of the overall requirements for experimentation. The information needs begin with the specification of the experimental procedure and its control parameters. Frequently, large amounts of information are required during execution (e.g., multiday experiments and computer-assisted instruction). After experimental results are obtained, other information documenting or aiding in the analysis of the data may be required. Clearly, information requirements are not limited to any single portion of an experiment.

An on-line information management system can be designed to meet the information requirements of experimentation. Such systems can provide a standard method of organizing, documenting, and implementing experimental procedures. These systems aid in the implementation of procedures that use historical events

This research was undertaken within the Institute for the Study of Intellectual Behavior and the Computer Laboratory for Instruction in Psychological Research. This work was supported in part by National Science Foundation Grants GB-34077X and GJ453. I thank Peter G. Polson for his encouragement and comments on an earlier draft of this paper. Requests for reprints should be sent to H. J. Durrett, Jr., Department of Psychology, University of Colorado, Boulder, Colorado 80302.

to determine future activities or that require a "closed-loop" experimental design. Further, such systems permit on-line modification and monitoring of parameters. In contrast to complex commercial information management systems that require substantial hardware and software support, it is possible to implement simple systems having similar characteristics. An efficient system for the on-line laboratory can be implemented using a direct access device.

INFORMATION MANAGEMENT SYSTEMS: TYPES AND TERMINOLOGY

Information management systems can be classified into two broad categories. One system is characterized by a data structure that must be known to the user. This system is capable of minimizing the software overhead and maximizing the flexibility of the system as it adapts to user's needs. However, software development is often complex and tedious. The other system is characterized by a data structure that is *not* known to the user. While the user's knowledge of the system is minimal or easily learned, such systems often have large software overhead and are not easily adapted to special tasks of the user. A successful system for the on-line laboratory must be a compromise of these two approaches making minimum demands on hardware and software resources and having an easily learned structure which allows varied uses. The SIM system is designed to meet these specifications.

Information management systems can be described using standardized terminology (see CODASYL, Data Description Language, 1973). The central component is the data base, which is a standardized collection of information and its relationships. A data base can serve as a repository of information collected from different sources. This information can then be accessed by various unrelated procedures. The SIM system data base is contained in a single direct access file and is accessed by a number of procedures.

The smallest unit of named data in a data base is a data item. In the SIM system, data items consist of either integer or alphanumeric values. A named collection of data items is termed a data aggregate. There are two kinds of data aggregates: vectors and repeating groups. A vector is a one-dimensional sequence of data items, all having identical characteristics, while a repeating group is a collection of data items that occur a number of times within a record. The data aggregate consists of data items, vectors, and repeating groups.

A named collection of zero, one, or more data items or data aggregates is called a record. There may be an arbitrary number of records in the data base. In the SIM system, records are referenced by integer numbers in the range one to the maximum number of physical records in the direct access file. A named collection of records having a similar function is called an element of the data

base. (The term "area" is specified by CODASYL but will not be used in this context.)

The complete description of the structure of a data base is termed a schema. It includes the names, descriptions, and specifications of all of the elements, records, and the associated data items for use by the information management system. A data base can be precisely defined as the collection of all records which are controlled by a specific schema.

AN OVERVIEW OF THE SIM SYSTEM

The SIM system consists of a data base, a set of utility programs, and a data base access routine that performs operations on the data base. All routines are written in FORTRAN IV, but any language that permits the use of direct access devices can be used. The data base is a single direct access file. The utility programs perform operations on the data base such as initializing, updating, inputting, and outputting the contents of the records. The utility programs are executed in a batch processing mode, but updating is done on-line. A set of data recovery and statistical analysis programs also operate on the data base. (One program generates a list of research participants for input to the participant accounting system used by the Department of Psychology.) The most important programs interfacing to the data base are the real-time or on-line experimental programs. These programs are independently written by the user of the SIM system and are designed to implement a particular experimental paradigm. They are not formally a part of the SIM system. It is possible for the user to monitor experiments using the system's on-line update program that observes information which is modified in the data base. Figure 1 presents a diagram of the interrelationship of the various programs and the data base. A single interface routine is used by the utility programs and the ex-

Figure 1. Organization of the Simple Information Management system.

perimental program to access the data base. This central interface routine is called ACCESS.

ACCESS: A ROUTINE FOR INFORMATION STORAGE AND RETRIEVAL

The ACCESS routine is used by all utility programs, as well as the on-line experimental program, for the storage and retrieval of information from the data base. This FORTRAN IV routine performs the following operations: (1) Assigns a particular direct access file for use as the data base. (2) Constructs the data base according to a specified data base schema. (3) Handles input or output specifying the alphanumeric name of any element or data item defined by the schema. (4) Stores information in a specified data item. (5) Retrieves information from a specified data item.

ACCESS has the following sextuple argument sequence: operation flag, element, record, position, length, and value vector. The *operation flag* is an indicator of the operation to be performed, either to (0) retrieve information from the specified data item and place it in the value vector, or (1) store the information contained in the value vector in the specified data item. *Element* is the number of an element; *record* is the number of a record within an element; *position* is the starting position of the data item or data aggregate within a record; *length* is the total number of data items that are to be operated on; and the *value vector* is the scalar or vector containing the data items which are to be operated on.

The data base contains four types of information controlled by ACCESS: (a) integer data items or aggregates, (b) alphanumeric data items or aggregates, (c) integer data items specifying data base parameters, and (d) alphanumeric data items specifying the name of the entities in the data base, such as file, element, or data item names. The internal housekeeping functions of the ACCESS routine involve checking the ranges of calling arguments and other similar tasks. These types of operations can be designed to the specifications of a particular implementation and are familiar to most programmers. The routine ACCESS uses several conventions for storing or retrieving information and for constructing the data base schema. Appendix A provides detailed information on the conventions used to construct the data base schema.

To simplify the input or output of information from external sources such as terminals, cards, or disks, the SIM system uses utility programs. These programs are not used by the on-line experiment program that must call ACCESS directly with internal argument sequences. These programs input information in free form and output it in a stylized format. The representation of all input or output to or from the utility programs is identical to the calling sequence of ACCESS. For free-form input, the argument fields are separated by the delimiter slash ("/") and values within a field are separated by commas (","),. Thus, for a simple case of a single value

input to the data base, the following sequence would be used:

1/1/2/3/1/27/

Here, Element 1, Record 2, Position 3 would receive a data item of Length 1 whose value is 27. A request for output uses a similar format but the final argument is omitted. Thus, the output request

0/1/2/3/1//

would produce the value input above, or 27.

DATA BASE FILE STRUCTURE

The direct access file structure used by the ACCESS routine is a function of the characteristics of the computer system on which the SIM system is implemented. The more closely an implementation conforms to the existing operating system software, the simpler the implementation. There are a number of sophisticated file structures, such as linked lists, that could be used. Usually, the more complicated the file structure, the more complicated the software. Since one of the major design criteria of the system was to minimize the use of real-time resources, little could be gained by using complex structures aimed at conserving direct access storage. Therefore, a simple file structure was adopted for the data base.

The SIM system data base is composed of physical records that are 512, 16-bit words long. Each physical record is partitioned into some number of positions or data items, as specified by the data base schema. Each partition contains a single logical record of a single element. Thus, the physical record "I" contains the corresponding logical records "I" of each element. When a reference is made to the Ith record of an element, the entire physical record of the direct access file is read into memory. The logical record requested from an element and the data item from that element is then obtained from this physical record.

The first M records of the direct access file are available for the user's information and are structured according to the data base schema. The remaining records in the file (M + 1 to N) are used by the SIM system to maintain the information specifying the schema, such as file, element, and data item names and parameters. This portion of the file has an arrangement that is a function of the particular system implementation specifications. The final record of the file contains parameters maintained by the SIM system that summarize the characteristics of the data base and allow for error checking and protection. A schematic diagram of the SIM system file structure as implemented on the XEROX Sigma 3 at CLIPR is shown in Figure 2. As can

NOTE: Each physical record is 512 16 bit words. Logical records are formed by partitioning this physical record.

Figure 2. Organization of direct access file used by the SIM system data base.

be seen in Figure 2, the file must have at least as many records as are required for the data base schema and the longest element. If elements have vastly different record lengths, large portions of the direct access file will not be utilized. This problem can be overcome by combining very small elements and by using a standard number of data items in each element.

A DATA BASE SCHEMA FOR ON-LINE EXPERIMENTATION

Because any integer value or alphanumeric string can be stored or retrieved, the user has flexibility in constructing a schema. An example drawn from a procedure used in the study of human problem solving will illustrate how a data base schema is constructed. The experimental procedure has been used by a number of experimenters (Millward & Spoehr, 1973; Polson & Durrett, Note 1). Similar schema are applicable to other types of experiments as well as computer-assisted instruction.

The human problem solving data base schema is extracted from the structure of experimental design. Such experiments involve some number of participants assigned to a condition which consists of several separate sessions. At each session, the participant solves several problems constructed from a set of stimulus materials. The problems must be solved within a specified period of time using a previously learned rule. The participant is given instructions throughout this entire process. After some specified number of sessions, the experimental condition ends and the participant is awarded credit.

This brief summary of the typical human problem solving experiment can be represented by the hierarchical structure shown in Figure 3. To translate this structure into a data base schema, the specifications for each element and its data items and aggregates are entered using ACCESS. This particular hierarchy has nine different types of information that can be represented as elements of the data base.

Shown in Figure 4 is the complete list of all elements

and data items to be contained in the data base. Some data items are integer values, while others are pointers referencing records contained in other elements of the data base [starred ("★") data items have record number values]. No distinction between these two types of information is made, since any integer value can be stored in a data item. Obviously, the real-time experimental program must access the information stored in the data base as appropriate to the intended application. In this way, the SIM system and data base remain independent of the experimental procedure. Clearly, the experimental program must exercise some care when retrieving data items. The ACCESS routine monitors all requests to prohibit any illegal operations but cannot control the use of values.

Figure 3. Hierarchy of information required for human problem solving experiments.

Element 1: Participant Information

1. Participant identification number
2. Date of entry into system
3. Total number of times participant in experiment
4. Total participation time
5. Sequence number of last data record generated
6. Session sequence indicator
7. Problem sequence indicator
8. Participation status flag
9. Terminal-hardware status flag
- *10. Experiment number 1 record number
11. Condition number
12. Date of participation
13. Total hours participation
14. Beginning sequence number of data
15. Ending sequence number of data

Element 2: Experiment Information

1. Experiment identification number
- *2. Starting instruction number
- *3. Ending instruction number
4. Number of conditions in experiment (C_n)
- *5. Condition number 1 record number
- *4+ C_n Condition C_n record number

Element 3: Condition Information

1. Condition identification number
- *2. Starting instruction record number
- *3. Ending instruction record number
4. Number of sessions in condition (S_n)
- *5. Session number 1 record number
- *4+ S_n Session number S_n record number

Element 4: Session Information

1. Session identification number
- *2. Starting instruction record number
- *3. Ending instruction record number
4. Number of problems in this session (P_n)
- *5. Problem number 1 record number
- *4+ P_n Problem number P_n record number

Element 5: Problem Information

1. Problem identification number
- *2. Starting instruction record number
- *3. Ending instruction record number
- *4. Stimulus material record number
- *5. Timing parameter record number
- *6. Solution parameter record number

Element 6: Timing Parameter Information

1. Timing parameter identification number
2. Inter-problem interval in seconds
3. Inter-trial interval in seconds
4. Delay before issuing prompt message
5. Prefeedback delay period in seconds
6. Feedback delay period in seconds
7. Maximum problem solving time in seconds

Element 7: Solution Parameter Information

1. Solution parameter identification number
2. Maximum number of trials allowed
3. Number of trials required to reach criterion
4. Categories required at criterion
5. Number of categories in problem
6. Number of examples in a category
7. Relevant category number 1
8. Relevant category number 2
9. Response assignment mode (either fixed or random)
10. Number of possible responses (R_n)
11. Value of response number 1
- 10+ R_n Value of response number R_n

Element 8: Instruction Parameter Information

1. Instruction identification number

2. Instruction timing mode (Indefinite, Fixed, or Variable)
3. Instruction timing parameters (Static or Dynamic)
4. Instruction timing factor
5. Number of records of instructions (I_n)
- *6. Instruction record number 1
- *5+ I_n Instruction record number I_n

Element 9: Stimulus Material Information

1. Stimulus identification number
2. Number of examples in category (V_n)
3. Alphanumeric name of category
4. Alphanumeric name of example 1
- 3+ V_n Alphanumeric name of example V_n

Note: * data items have record number values

Figure 4. Elements and data items for human problem solving experiments as specified in the data base schema.

An example of the input necessary to specify the timing parameter element of the data base is shown in Figure 5. These input records are presented in the format accepted by the utility programs. The data items are stored as indicated by the schema. The SIM system utility output procedure provides a stylized listing of the information. Figure 6 presents an example of what is produced from the information input in Figure 5. All representations of information conform to the ACCESS routine calling sequence.

REAL-TIME EXPERIMENTAL PROGRAMS

The intent of the SIM system is to allow experimenters to design, implement, and execute complex psychological experiments that would otherwise be difficult because of their information requirements. The critical component for the experimenter is the real-time program that conducts an experiment and uses the information contained in the SIM system data base. The experimental program selectively stores and retrieves information as necessary by using the ACCESS routine.

The particular schema presented in Figure 4 implements a linked record structure to be used by the experimental program. Although the details of the program are beyond the scope of this paper, it is useful to see how the program would use the information specified by this schema. The program would initially input a particular number from a terminal and locate the corresponding record in Participant Element 1. After retrieving the contents of the selected participant record, the program would use the experiment record number (Element 1, Data Item 10) to retrieve a record from Experiment Element 2. The condition number (Element 1, Data Item 11) would be used to select the condition record number (Element 2, Data Items 5-4+ C_n) from Experiment Element 2. The appropriate record of Condition Element 3 would then be retrieved using the obtained number. Once the condition record had been retrieved,

Input required to define data base, element, and positions for all records in Element 6.

```
1/0/0/0//PROBLEM DATA BASE/
1/6/0/0/0//TIMING PARAMETERS/
1/6/0/1/1//TIMING PARAMETER IDENTIFICATION NUMBER/
1/6/0/2/1//INTER-PROBLEM INTERVAL IN SECONDS/
1/6/0/3/1//INTER-TRIAL INTERVAL IN SECONDS/
1/6/0/4/1//DECAY BEFORE ISSUING PROMPT IN SECONDS/
1/6/0/5/1//PRE-FEEDBACK DELAY IN SECONDS/
1/6/0/6/1//FEEDBACK DELAY PERIOD/
1/9/0/7/1//MAXIMUM PROBLEM SOLVING TIME IN SECONDS/
```

Input required to define the values of data items in a single record of Element 6.

```
1/6/1/1/7/ 1,60,2,10,5,2,3600/
```

Figure 5. Sample input specifying the timing parameter element of the data base and the content of a single record.

the session sequence indicator (Element 1, Data Item 6) would be used to select the next session record number (Element 3, Data Items $5.4+S_n$). Retrieving this record from Session Element 4, the program would use the problem sequence indicator (Element 1, Data Item 7) to select the next problem record number (Element 4, Data Items $4.5+P_n$). When the problem record is retrieved from Problem Element 5, the contents of the specified records of Timing Element 6, Solution Element 7, Instruction Element 8, and Stimulus Material Element 9 would be retrieved. With all of this information, the experimental procedure begins execution. With execution underway, the program updates data items in Participant Element 1 for future operations. Additionally, the program would output data to perhaps other unspecified elements of the data base to document all its operations. All of these procedures are compatible with real-time operations.

The program will interact with the data base storing and retrieving information throughout the experiment, and it will use the data base as a virtual memory that minimizes central memory requirements. Further, the program will respond in sophisticated ways to complex events by executing predetermined operational sequences. The program designed to execute the human problem solving experiments uses one element of the data base for a table specifying the operations to be per-

```
DATA BASE NAME = PROBLEM DATA BASE
ELEMENT NAME = TIMING PARAMETERS
ELEMENT NUMBER = 6
RECORD NUMBER = 1
```

POSITION	LENGTH	DATA ITEM	ALPHANUMERIC NAME
1/6/1/0/	1/	7	/MAXIMUM POSITION IN RECORD/
1/6/1/1/	1/	1	/TIMING PARAMETER IDENTIFICATION NUMBER/
1/6/1/2/	1/	60	/INTER-PROBLEM INTERVAL IN SECONDS/
1/6/1/3/	1/	2	/INTER-TRIAL INTERVAL IN SECONDS/
1/6/1/4/	1/	10	/DELAY BEFORE ISSUING PROMPT IN SECONDS/
1/6/1/5/	1/	5	/PRE-FEEDBACK DELAY IN SECONDS/
1/6/1/6/	1/	2	/FEEDBACK DELAY PERIOD/
1/6/1/7/	1/	3600	/MAXIMUM PROBLEM SOLVING TIME IN SECONDS/

Figure 6. Sample output from a single record of the timing parameter element.

formed during the experiment. It retrieves these operations and "executes" them. This table can be modified to allow the program to perform other experimental procedures without the modification of the program itself.

CONCLUSION

The SIM system has been used to conduct four studies of human problem solving, one on sentence evaluations, and one on figure recognition. The ease with which complex experimental paradigms involving different parameters and materials have been implemented has been a major benefit of the system. Further, the utility programs supporting the data base have provided detailed documentation of each of the experimental procedures and its associated parameters. The memory requirements of ACCESS are less than previously required by experimental program data input/output procedures. Further, this standardized information management system is easier to use than other input/output procedures.

Future work will center on expanding the SIM system for use with the modules of the CLIPR display terminal experiment station (Spear, Overgard, & Christian, 1975). The SIM system has demonstrated the usefulness of an information management system for use in on-line experimentation. It is hoped that readers can use this concept to implement similar systems in other laboratories. The potentials are just beginning to be recognized.

APPENDIX A

To provide for the efficient storage and retrieval of all information using the single routine ACCESS, special conventions for the calling arguments are used. First for the purpose of consistency throughout the SIM system, all interactions consist of six fields of information. These six fields apply to the six arguments of the calling sequence for the routine ACCESS. Both the input and output medium will use each of these same six fields as defined below:

Field	Contents
1	Operation flag
2	Element flag
3	Record number
4	Position number
5	Length of data items or number of data items to be stored or retrieved
6	Scalar or vector containing values to be stored or retrieved

Several special cases exist that make possible the input of the data base schema. For the storage of the data base name, the following conventions are used:

Field	Contents
2	0, (Zero)
3	0, (Zero)
4	0, (Zero)
5	Maximum number of records allocated

- 6 Alphanumeric name of the data base

For specification of elements within the data base, the following conventions are observed:

Field	Contents
2	Element number
3	0, (Zero)
4	0, (Zero)
5	Integer specifying the number of data items in each record of the element
6	Alphanumeric name of the element

For the specification of data items or aggregates within the records of an element of the data base, the following conventions are observed:

Field	Contents
2	Element number
3	0, (Zero)
4	Position of the data item or data aggregate
5	Number of sequential positions within the record to which this name will apply (Note: If number is greater than one, field is assumed to be alphanumeric.)
6	Alphanumeric name of the data item or data aggregate

REFERENCE NOTE

1. Polson, P. G., & Durrett, H. J. The direct measurement of strategies in complex concept identification tasks. Paper presented at the Psychonomic Society meeting, St. Louis, Missouri, November 1, 1973.

REFERENCES

- BAILEY, D. E., & POLSON, P. G. Real-time computers in psychology at the University of Colorado. *American Psychologist*, 1975, 212-218.
- DURRETT, H. J. A multiprogram approach to real time experimentation. *Behavior Research Methods & Instrumentation*, 1974, 6, 259-262.
- KADDEN, R. M. State notation and SKED: A general system for control and recording of behavior experiments. *Behavior Research Methods & Instrumentation*, 1974, 6, 167-170.
- MILLWARD, R. B., & SPOEHR, K. T. The direct measurement of hypothesis-sampling strategies. *Cognitive Psychology*, 1973, 4, 1-38.
- POLSON, P. G. SCAT: Design criterion and software. *Behavior Research Methods & Instrumentation*, 1973, 5, 241-244.
- RESTLE, R., & SHAFFER, W. D. Monitoring experimental results. *Behavior Research Methods & Instrumentation*, 1974, 6, 262-266.
- SPEAR, T. L., OVERGARD, D., & CHRISTIAN, I. W. The CLIPR display terminal experiment system. *Behavior Research Methods & Instrumentation*, 1975, 7, 107-112.
- U. S. DEPARTMENT OF COMMERCE, NATIONAL BUREAU OF STANDARDS. CODASYL Data Description Language. *Journal of Development*, June 1973, NBS Handbook 113, 2.5.