Proceedings of the 1987 Winter Simulation Conference
A. Thesen, H. Grant, W. David Kelton (eds.)

DATA COLLECTION WITH A PORTABLE MICROCOMPUTER

James E. Wilkinson
Gary E. Whitehouse, Ph.D., P.E.
University of Central Florida
College of Engineering
Industrial Engineering and Management Sciences
Orlando, Florida 32816-0450

ABSTRACT

Advances in simulation methodology, and the computer support systems to implement these methodologies, have led to the development of integrated simulation support environments (such as TESS). These environments, or collection of software tools, seek to aid the analyst in developing models, data management and analysis, and data presentation. While an integrated simulation environment provides a more organized structure for managing and performing simulation projects, and provides a database management structure for storing, manipulating, and analyzing data, they do not address the actual process of going out and obtaining the data. As a result, many of the common problems associated with poor problem and system definition and low quality model input data, may still occur. This paper examines the concept of a support-support system; a portable microcomputer with software tools designed to support collection of the data, both subjective and objective, required in a simulation study. To demonstrate how this concept can be implemented, a BASICA program which supports model input data collection is discussed. Using graphic display to define input data requirements and single key inputs, this program should maximize the time an analyst can spend observing the system and minimize the time he/she has to spend entering data.

1. INTRODUCTION

In recent years there have been numerous advances in the "art and science" of simulation and associated software. New simulation languages and methodologies have been developed that reflect a "shift from the program to the model view of the simulation process" (Nance 1983). There has also been an increased interest in the development of support software and simulation "environments" that comprise integrated collections of these software tools (Henricksen 1984). One of the more visible areas has been in the area of interactive simulation involving graphics and animation.

In regards to simulation environments, an integrated environment is a collection of software tools for designing, writing, and validating models; writing and verifying data; and designing and carrying out experiments with models. An important aspect of an integrated environment is that the user is shielded from being a computer technician, i.e., having to spend a significant amount of time managing files, making sure data are in correct formats, data are in correct formats, etc. (Henricksen 1984; Standridge and Walker 1983).

Additionally termed as integrated simulation support systems, they seek to "deal with ... other aspects of performing a simulation project" such as model building, user data management, data analysis, and data presentation (Standridge and Walker 1983). A primary attribute is integrated data management, which seeks to address the needs of newer simulation methodologies and the complexity of the systems now being modeled (Standridge 1983; Evers, Bachert, and Santucci 1981). Standridge (1985) states that the requirements for these systems have evolved to include: a framework which provides a more standardized way for performing a simulation project, a system which provides effective user interfaces, and a system which is integrated.

Such a system, coupled with a effective methodology, can alleviate many of the problems and failures of simulation projects. These problems include the need to effectively communicate model formulation and assumptions; poor communication, both with user and project team members; "failure to use modern tools and techniques to manage the development of a large, complex computer program"; and failure to adequately describe the system (Evers, Bachert, and Santucci 1981; Annino and Russell 1979; Huhn and Comer 1981).

Other just as common problems result from poor or inadequate data, where data is defined in its broadest context to include subjective data, such as information about the problem, objectives, and system to be modeled; as well as objective data such as that used to define variable/parameter inputs and to validate the model. Some of these problems include (Miller & Pare 1986; Evers et al. 1981; Shannon 1975):

1. A poor and incomplete definition of the problem.

2. Failure to identify the objectives of the model.

3. Failure to adequately identify the data needed to execute the model and how this data is to be collected.

4. Using statistical procedures that the data is independent and identically distributed when it is not.

5. Assuming observations in the data set are homogeneous when they are not.

While an integrated simulation environment provides a more organized structure for managing and performing simulation projects, and provides a database management structure for storing, manipulating, and analyzing data, they do not address the actual process of going out and obtaining the data. Thus many of the above mentioned problems may still occur.

This paper will discuss a concept for a system which could be called a support-support system. The objective or purpose of such a system would be to provide a more organized or disciplined approach and tool for the data collection phases of a simulation project and in turn support the data and information needs of an integrated support system, such as TESS. This would in turn reduce the chances of failure due to poor or inadequate data.

In approaching this problem it was felt the future is represented by integrated support systems and while data acquisition systems feeding the central data base of such a system is the ideal, this is not always possible. The next step down from this would be a portable computerized system which could be designed along the same lines as an integrated support system and could be used by the analyst to guide his data and information collection efforts. The results could then be ported to the main support system for final analysis and use in modeling the system and running the simulation. It is felt such a system can provide an organized and structured framework for the early phases of the project which in some cases would simplify the process and make it more efficient, and in all cases aid in avoiding some of the more common mistakes.

1.1 Basic Concept

The primary purpose of this portable support system will, therefore, be to support the data collection aspects of a simulation study. Detailed analysis of this data will occur in the primary support system. However, the proposed support system needs to provide for enough analytical ability so the analyst can adjust the collection effort as problems arise or new information becomes available.

In general, the portable support system is to help in:

1. organizing and guiding the collection effort;

2. organizing and archiving the data;

3. displaying the data for review, analysis, update, and validation; and

4. producing required reports.

The goal of the proposed system is to insure the data (both subjective and objective) necessary to perform a simulation project is gathered in as efficient and effective manner, and the problems common to many projects be avoided.

1.2 Major Functions

The proposed support system should provide help in four major areas and two minor areas as shown in Figure 1. The major functions consist of problem definition, system definition, input data collection, and project planning and management. The minor functions support the other four and are economic analysis and report generation. This section will briefly discuss these six functions.

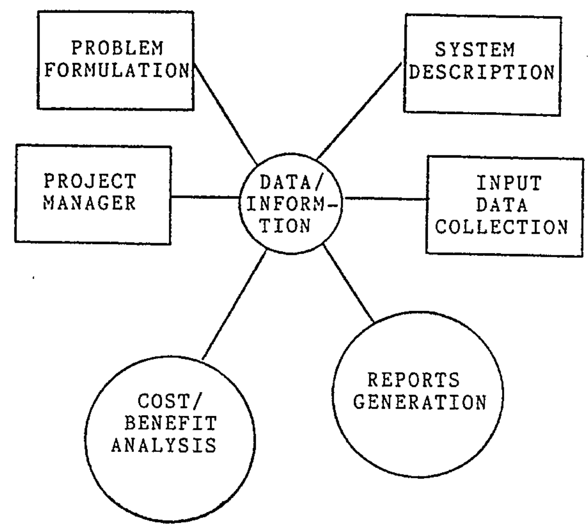

Figure 1: Support-Support System Architecture

Formulate Problem. The support system should aid in the orderly collection of data necessary to define the problem.

The problem definition will consist of four issues. The first is a workable definition of the problem as communicated to the analyst by the users. Often times the only thing people are sure about is the fact there is a problem. The analyst will start with a list of symptoms and have to identify the underlying causes. The second is to take into account the politics of the organization as it relates to the problem and its solution. That is why it's important to identify and interview all key people who can add to the understanding of the problem, and also affect the implementation of the solution. Third, is to determine the benefits that will results from solving the problem. This is closely related to the fourth issue. What will it take to solve the problem; both in terms of methodology, manpower, and money? Normally if the cost of solving the problem is more than the benefits, there is little reason to continue the project.

The data will be collected from numerous sources, but the primary method will be by interviewing key decision makers, managers, and system operators (experts). Other sources include observation, and researching applicable regulations, procedures, manuals, and policy statements. Of interest also, is whether any previous studies have been accomplished concerning the problem. These previous studies can provide valuable insight into both the problem and politics of solving the problem.

System Description. The system should aid in developing a thorough and accurate understanding and description of the system to be modeled. In doing this the system should provide various methods to schematically represent the system. This is necessary because different systems require different methods and different analysts may prefer one method over another. Also, a combination of methods may be required to adequately describe the system operation.

Whatever methods are used, it should be compatible with or translatable to the modeling representation used by the primary support system. However, the primary function is to obtain knowledge about the system and have it depicted and described in such a fashion that this knowledge can be understood and validated by the user's/sponsor.

Input Data Collection. The system should aid in developing a data collection plan and provide formats and means for collecting this data. The primary support system, through the modeling process, will define what data is to be collected, but the proposed support system should help in planning where, when, and how to collect it. It should also provide a means to archive this data and aid in inputting it to the primary support system. This system should provide for preliminary analysis of the data to insure it meets the statistical assumptions of independence and homogeneity. Finally, the system should provide a means to collect data by the observing system operation and/or by inputting data from documents.

Project Planner. The system should support the required planning tasks, both personal and for the project as a whole. The project planner must also help in accounting for resources expended to accomplish the project. The support system should also help in developing project cost estimates.

Report Generator. This function will support the other five. The system should provide help in developing all reports necessary during the initial phases of the project. Standardized formats should be developed which permit the data to be entered as it becomes available. However, the system must provide the flexibility to adjust to requirements of the particular project or analyst.

Economic Analysis. The system should help in analyzing the cost/benefits of solving the problem. Coupled with the project planner function, the system should help in making many of the trade-off decisions that arise during the course of a project. This includes the amount of data to be collected. Is there a point at which the cost of collecting more data/information overshadows the benefits?

1.3 General Support System Requirements

The support system should support the iterative nature of this phase of the process. While the simulation process as a whole is an iterative process, the early investigative phases are particularly so. Each source of information leads to another.

The support system should be able to adapt to the nature of the particular project. Each problem and system are different. While the support system calls for a more standardized and structured approach, it must be flexible.

In the same way, the support system should take into account the various approaches used by different analýsts. Whenever possible, the support system should provide multiple formats and techniques for gathering the information.

Data collected in support of one of the above listed functions but applicable to one or more of the other functions must be identified as such and cross-referenced. This is to prevent the analyst from collecting the same data twice.

Finally, the support system should insure the analyst considers all information and data requirements. Each project is different so some of the data requirements will not be applicable, but the analyst should make this decision and simply not overlook it. Also, information/data requirements which are applicable but not yet defined, should result in assumptions and hypothesis concerning the data. Then the analyst can continually review and update these assumptions as information/ data becomes available. These assumptions/ hypothesis can, in turn, help the analyst develop new questions and avenues in investigation.

1.4 Concept for Implementation

Though not hand-held, some of the newer portable microcomputers are small and light enough to be called "lap-top" and have capabilities equal to the desktop PCs; up to 640K of RAM, dual disk drives, full 80X25 character/line displays and full keyboards. These qualities make up for the limitations of the hand-held devices and can provide the capability to support all aspects of the data collection process.

One drawback at this time is the lack of readily usable software. Most of these computers use the 3 1/2 inch disks. They can hold up to 720K data but there is not as much software available in the 3 1/2" disk size. This should only be a temporary problem, though.

Ideally, to implement this system, a single software package which integrates all the required functions is necessary. As a starting point, however, this study suggests that the required application programs, consisting of standard formats, be developed using generic software, such as, spreadsheets, database managers, word processors, and processors, and programs written in BASIC. Whenever possible, utility programs should be written to integrate the different functions.

These software packages are readily available and most potential users are, at least, somewhat familiar with their operation. Also, they provide the user the maximum flexibility to adjust them to his/her own desires and/or the particular situation.

Application programs should be run in drive A and a data disk in drive B. To avoid mixing data, a data disk(s) should only contain data from one project. Also, the concept envisages the need for numerous data files. Using a 720K 3 1/2" disk, the user is limited to 112 files. If subdirectories are used then the number of files are limited only by the amount of memory on the disk. A tree type file directory is therefore recommended (Figure 2).

```
                              ROOT
   |           |          |         |          |          |
\PROBLEM  \SYSTEM  \INPUT  \REPORT  \PLAN  \ECON
   |           |          |         |          |          |
  DATA       DATA      DATA      DATA       DATA       DATA
  FILES      FILES     FILES     FILES      FILES      FILES
```

Figure 2: Data Disk Directory

The remainder of this section provides some brief suggestions on how each system function could be implemented.

Formulate Problem. Utilizing a word processor or data base manager, formats should be created to lead the analyst through the process of collecting data. A data base manager, dBASE III would appear to provide the most flexibility, because the programmable features of dBASE III would allow the analyst to input both the formats and the logic to guide him/her through the process. Database files could be created for each data type category.

To provide a basis for building the formats and logic, Balci and Nance (1985) offer a very good methodology which lends itself to computerization. As long as the problem can be categorized as descriptive or prescriptive, the analyst follows a list of questions/information requirements which, depending on the outcome of each step, guides the analyst to another appropriate question/information requirement. Refer to Balci and Nance (1985) for additional information. The article also describes a questionnaire type procedure for verifying the formulated problem.

Not included in this procedure are the data categories of assumptions and hypothesis. These would be added with the logic guiding the analyst to make assumptions and hypothesis regarding data that is not yet known. Then as information is collected, the analyst will update the assumptions. The goal here is to insure the analyst considers all the data requirements.

System Definition. This function will involve a verbal description of the system, plus a means to schematically depict it. Various formats for doing this should be available to the analyst. Shannon (1975) suggests process charts, flow diagrams, activity charts, and block and logic diagrams, as possible ways to help describe the system being modeled. Pictorial graphics is also on option. Whichever way is used, the system should have a word description file associated with each schematic, so as the schematic is created or analyzed, the details of its operation can also be entered/reviewed.

Project Planner. Numerous project planning software packages are available. The choice of which one to use would be up to the user.

As stated earlier, a simulation study is a project and must be managed. A project planner would help the analyst do this. It could be used to manage the personal time of the analyst and project staff. It could also be used to estimate and track the project resources: money, manpower, equipment, and time.

A standardized project format needs to be developed which will provide a starting point. The analyst can then adjust this format to suit the needs of the particular project. Also, each function should incorporate a utility which logs the time and other resources expended and updates the project planning files.

Economic Analysis. The primary focus of this function is to aid in conducting cost/benefit analysis studies of the project. Any of the engineering economy software packages will suffice. However, coupled with the project planner and a decision type model (Gray 1978) this function can help in making many of the trade-off decisions that arise during the course of a project.

Reports Generator. Standardized report formats can be developed using the programming portion of dBASE III or a similar data base manager. Then the applicable data can be merged from the appropriate files.

Input Data Collection. The system needs to support collection from at least two general sources: by observing the system, and from documents.

To aid in collecting data from documents, a program is needed to help the analyst build collection formats. Then the data can be entered into the computer as these documents are reviewed. The required formats can be determined during the investigative phase. The programmable features of a data base manager could be used. This would provide great flexibility, but the analyst would have to develop a program for each format. Thus, a program needs to be developed that will assist the analyst in building these formats.

Regarding the other source of input data, observing the system, there are many programs and collection devices that support time and motion studies that can be adapted to this function. MacMillan and Walker (1985) developed such a program that provides a conceptual framework for the input data collection function of this support system. The remainder of this paper will review this concept and outline a program that partially implements this specification.

1.5 Input Data Collection - Concept

Basically, a data collection device should be based on the concept that the analyst's primary function is to observe. Data entry should be done in such a way as to minimize the time the analyst spends entering data, as this distracts from his primary task of observing.

MacMillan and Walker (1985) developed a program, written in BASIC for use on a Radio Shack TRS-80 Model 100, and is recommended for a starting point. By using the computer clock, the time an event occurs can be logged. They have even developed a routine to access the computer timing crystal for greater accuracy. When an event occurs the operator hits enter and the time is logged. The operator is then prompted for an event number. This is, in turn, entered and logged. The operator must keep track of event or activity numbers and the logging of time and event identifiers requires multiple keystrokes. Despite these drawbacks, it does offer a good starting point.

1.6 Program Specification

Because the system specification developed earlier is rather general and conceptual in nature, this section will list some more definitive specifications.

1. The program should be menu driven and/or the user asked to respond to specific questions, i.e. easy to use.

2. Data entry should be limited to a single keystroke to the maximum extent possible. All information which describes the specific event or activity should be entered prior to beginning data collection. This is so the analyst can maximize the time spent observing, and thus maximize the number of events/activities he/she can track.

3. The computer display should define or display data to be collected and aid in tracking the collection process. Put another way, the display should somehow define what activities to observe, how to enter data for this activity, and what data has been entered.

4. All data file operations should be invisible to the user. The computer should manage the files, not the user.

5. Data files should be identified by project, scenario, study number and activity number. The project number is a name or number which ties the data to a particular simulation project. A scenario defines a specific set of activities or events. A study number defines one set of homogeneous data collected for a particular scenario. For example, if data are collected on a particular activity over a period of several days, each days data would be associated with a particular study number and kept in a separate file. This is so the data is not merged until it can be statistically analyzed to determine if it is homogeneous.

6. The program should also collect data and calculate statistics that supports model validation. While this is more a function of properly defining the proper data requirements (a function of the primary support system), the collection program can help by keeping track of queue size, etc.

7. Data generated in this program will be filed under the \INPUT directory.

1.7 Program Development and Operation

The program is predicated on the concept that the modeling process accomplished by the primary support system will determine what inputs are required to run and validate the simulation program. What the support-support system will do is help the analyst to plan how the data will be collected and a means to collect it (see Figure 3 for general logic diagram). The first step is to develop a plan.

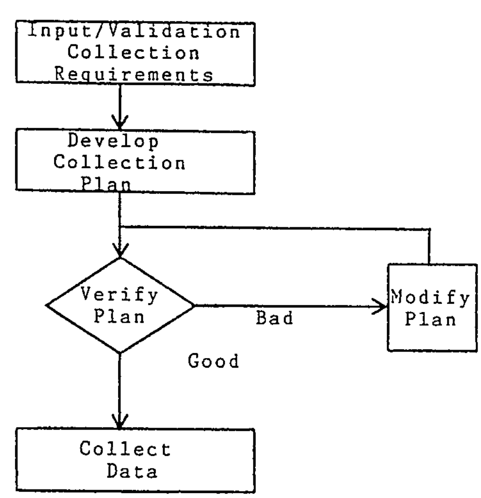

Figure 3: General Program Logic

1.8 Developing the Project Plan

The plan is developed by building the required collection scenarios. The scenarios, as defined above, are the activities/events an analyst will observe from one location. How many activities one analyst can observe and collect data on from one location depends on the physical layout of the system, the complexity of the activities/events, and the skill of the observer/collector.

This program allows the analyst to collect data for six distinct activities. If there are similar activities occurring at the same time (termed parallel activities in this program) the program will collect data on up to three parallel activities for each distinct activity. Thus, one scenario may actually collect data on up to 18 activities/events.

The collection plan is defined by a simple schematic composed of three types of activities: an arrival or departure event (circle), a delay or queue event (a D symbol), and an action or service event (a square) (Figure 4). While these symbols are geared to queuing systems, the action or service activity can represent any type of event or action, i.e., the time it takes to service a bank customer, or the time it takes to move a part from A to B, or the time it takes to perform a drilling operation, etc. [Caveat: This program is only seen as a starting point. The number or symbols and their layout/display can be changed and expanded in future program enhancements. The present layout was chosen to maximize the amount of data displayed on the schematic using normal text entry.]

To build this plan, you start the program and at the main menu (Figure 5) select option 1, Develop Collection Diagram. The result is an input format (Figure 6). An explanation of the inputs follows:

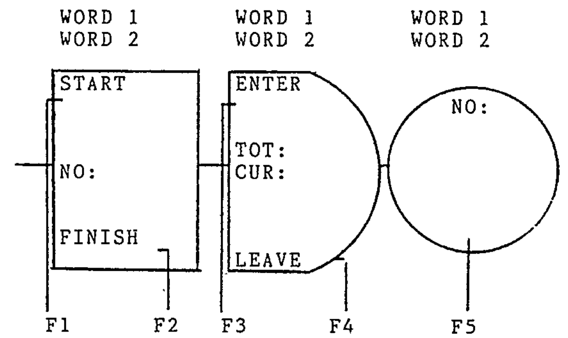

Figure 4: Activity/Event Symbols

MAIN MENU

1. DEVELOP COLLECTION DIAGRAM
2. MODIFY COLLECTION DIAGRAM
3. COLLECT DATA
4. DATA FILE MAINTENANCE
5. EXIT PROGRAM

ENTER SELECTION:? [

Figure 5: Main Menu

A. Project NO/Ident (8 characters) - Used to uniquely identify the project. Can be either letters or numbers.

B. Collection Scenario NO (2 characters) - Used to identify the scenario. Can be either letters or numbers.

MENU - ACTIVITY TYPE

1. ARRIVAL/DEPARTURE ACTIVITY (ONE COLLECTION POINT)
2. DELAY/QUEUE/WAIT ACTIVITY (TWO COLLECTION POINTS)
3. SERVICE/ACTION/ACTIVITY (TWO COLLECTION POINTS)

```
**************************************************

PROJECT NO/IDENT: ? A    *COLLECTION SCENARIO NO: B  *NO OF ACTIVITIES: C *
                    ACTIVITY         PARALLEL ACTIVITY NO:
TYPE OF ACTIVITY (SELECT FROM MENU): D *   PARALLEL ACT(Y/N): E *  NO F *
DESCRIPTION OF ACTIVITY (UP TO 66 CHARACTERS):
**                        G                                       **
SHORT DESCRIPTION TO BE USED TO LABEL SCHEMATIC (ONE OR TWO WORDS - LIMIT
   TO 8 CHARACTERS):
       FIRST WORD: H       *SECOND WORD: I[     **
```

Figure 6: Input Format to Build A Data Collection Schematic (Plan)

C. NO of activities - Used to identify the number of collection activities for the particular scenario. The current program limits the maximum number of activities to six.

For each activity the program request the following:

D. Type of Activity (Select from Menu) - The menu is located above the input format (Figure 6). Can enter one of three type of activities as discussed above.

E. Parallel ACT(Y/N) - As discussed above, if activities of the same type as that defined in D are going to be observed, you answer Y. This allows you to increase the number of activities you can observe and collect data on. If the answer is yes, enter Y. The cursor will go to F. If you enter N the cursor by-passes F and goes to G.

F. NO - The number of parallel activities associated with the current activity.

G. Description of Activity (66 characters) - Allows the user to enter a description of the activity/event. Should specify enough detail to sufficiently define the required data.

H. Short Description to be used to label schematic (one or two words, eight characters each) - On the collection plan schematic each activity diagram is labeled with these words. User should use key words to help in identifying the collection node. If there are parallel activities, a suggestion is to enter one of __ (like report page numbers) so you can identify which activity is being displayed.

The program will cycle through D-H for each activity. For parallel activities, the program will cycle G-H for each parallel activity/event. Thus each data collection point is individually defined and described.

When the last activity information is entered the program will display a schematic which the analyst will use to collect the data (Figure 7). The program asks the user if he/she wants to make any changes (not shown - currently limited to a change. Add a symbol and delete a symbol will have to be added to the program).

After all changes are made the program saves this information to a file. The program automatically develops a file name (Figure 8). The default disk drive for all data files is always drive B. The program then returns to the main menu.

```
 *                                   **
B: INPUT  P R _ _ _ _ _ _ . SDF

          First Four    Scenario
          Characters    Number
          Of Project
          Name
```

* Directory path not included in current program
** Simulation Definition File

Figure 8: Collection File Name

At the main menu the user can develop another collection scenario, modify an existing one, or retrieve a scenario to collect data. [Option 4, File Maintenance is not completed.]

1.8 Collecting Data

To collect data, option 3 is selected. The program then prompts the user for a project name, scenario number, and a study number (previously defined). The scenario collection plan is retrieved from the data disk. The program first displays a summary of the file information (Figure 9). Then the collection plan schematic is displayed (Figure 7).

When the plan was developed, an input key (function key) was automatically assigned and is displayed with the schematic. For arrival/departure nodes you collect one clock

PROJECT NO/IDENT: FACTORY   SCENARIO NO: 01   STUDY NO: 1
START TIME: 00:14:06   STOP TIME: MEMORY REMAINING:

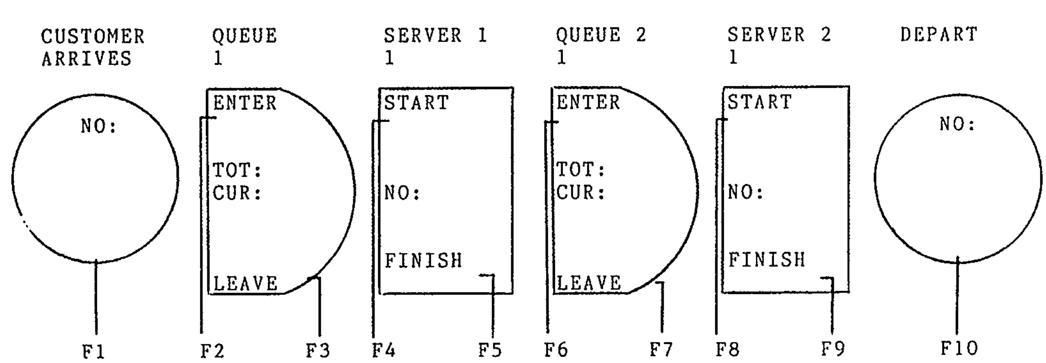

Figure 7: Data Collection Schematic (Plan)

PROJECT NO/IDENT: FACTORY   SCENARIO NO: 01   NO OF ACTIVITIES: 6

ACTIVITY #	TYPE ACTIVITY	PARALLEL ACTIVITY	SHORT ACTIVITY DESCRIPTION	F KEY(S)
1	1	1	CUSTOMER ARRIVES	F1
2	2	1 2	QUEUE 1 QUEUE 1 2	F2,F3
3	3	1 2	SERVER 1 1 SERVER 1 2	F4,F5
4	2	1 2 3	QUEUE 2 1 QUEUE 2 2 QUEUE 2 3	F6,F7
5	3	1	SERVER 2 1	F8,F9

PRESS ANY KEY TO CONTINUE

Figure 9: Data Collection Schematic (Plan) Summary)

time each time the event occurs. This can be used to calculate interarrival times. The delay and activity nodes have two collection points; an enter or begin time and a leave or end time. The program then calculates a duration for the event. [Caveat: as currently written the duration or average delay time will only be accurate for FIFO, first-in-first-out, queue priority. Also, a time in system can only be determined for single-queue, single-server systems with FIFO priority. However, the collection scenario will not always define the entire system. The program was designed to collect data on various parts of a system and the order in which the nodes appear should reflect the order in which events are to be observed, not the order in which entities flow through a system. This does limit some of the validation data and needs to be corrected in future research.] Also calculated and archived are maximum number in queue and average number in queue.

To enter data for a particular event the appropriate function key is pressed. The program calculates the time, places the time data into an array, calculates the number in queue or in a service activity, and updates the display. If there are parallel activities, the computer prompts the user for the appropriate parallel activity number. This number is entered by pressing one of the numeral keys. The enter key does not have to be pressed for data entry. Thus, the maximum number of key inputs is two. The user does not have to enter any other identifying data. This was already done in the scenario development phase.

The other information displayed on the screen during data collection is the name of the project, the scenario number, and study number. When the collection routine is first called, a start time is calculated and displayed. The same thing occurs when data collection is terminated. The amount of memory remaining is also displayed. If the machine has more than 256K of RAM, memory shouldn't be a major problem. The machine used to develop and run this program only had 256K which limits the number of collection entries per activity to between 50 and 75, and the number of parallel activities to three.

To terminate data collection, the Ctr-F10 keys are pressed simultaneously. When this is done, the data is automatically stored to files on the data disk. Each activity has its own file. Data Files are formatted as shown in Figure 10. Data stored includes the activity identifying data, the raw time data, the number of data entries, and for an arrival/departure event; interarrival times, maximum interarrival time and the average interarrival time. For the delay/queue activity, the raw enter/leave times are entered along with elapsed time, maximum number in queue, average number in queue, maximum delay time, and average delay time. For service/action activities, the raw enter/leave times are entered along with elapsed serviced times, maximum service times and average service times.

A file identifier is calculated as shown in Figure 11.

2. CONCLUSION

This paper has discussed a concept for a more organized and structured approach to the data/information gathering phases of a simulation project. It proposes using a portable microcomputer, with appropriate software, as a "support-support" system. When coupled with an integrated support system, such as TESS, this system could provide support for the entire simulation project.

To demonstrate how this concept can be implemented a program was discussed which supports the input data collection function. It's easy to use. File operation is invisible to the user, and a minimum number

ARRIVAL/DEPARTURE
ACTIVITY

Project Name, Scenario No, Program Counter

Activity Type, Parallel Activity No, Program Counter

Long Activity Description

Short Activity Description 1, Short Activity Description 2

Study No, Date, Start Time, Terminate Time

Number of Observations in This File

Number of Entities or Actions Completed in this Activity, 0

Observation Number, Event Time, Time Since Last Event (IAT)

⋮

Average Interarrival Time, Maximum Interarrival Time

DELAY OR SERVICE
ACTIVITY

First 7 entries the same as Arrival/Departure Event

⋮

Entry 8
Observation Number, Enter Time, Leave Time, Elapse Time

⋮

Average Elapse Time (Service Time), Maximum Elapse Time, Average Queue Length (For Delay Activities)

Figure 10: Data File Contents Format

B: INPUT* _ _ _ _ _ _ _ _ . SCF**

First 2 CHARACTERS Of PROJECT NAME | SCENARIO NO. | STUDY NO | ACTIVITY IDENTIFIER NUMBER

* Directory Path not included in current program.
** Simulation Collection File

Figure 11: Collection File Name

of keystrokes are required to enter data. The user only has to respond to input prompts to define a collection plan and a schematic collection format is proposed automatically.

The concept and program presented only represent a starting point, however. A great deal of research and work remains to be done. It does represent an attempt to take advantage of the capabilities of a portable microcomputer and how these devices can aid the simulation analyst in conducting a simulation study.

3. REFERENCES

Annino, J.S. and Russell, E.C. (1979). The ten most frequent causes of simulation analysis failure and how to avoid them. Simulation, 137-141.

Annino, J.S. and Russell, E.C. (1981). BASIC (users manual), IBM.

Balci, O. and Nance, R.E. (1985). Formulated problem verification as an explicit requirement of model credibility. Simulation, 76-86.

Banks, J. and Carson, J.S. (1984). Discrete Event System Simulation. Englewood Cliffs, Printice-Hall, Inc.

Barnes, R.M. (1968). Motion and Time Study, Design and Measurement of Work, Sixth Edition. John Wiley & Sons, Inc., New York.

Barnes, R.M. (1985). Catalog of Simulation Software. Simulation, 196-205.

Cobbin, P. (1987). A Tutorial on the SIMPLE-1 simulation environment. Sales literature received from Sierra Simulations and Software, Campbell, CA.

Collella, A.M., Carlino, D.J., and O'Sullivan, M.J. (1974). System Simulation Methods and Applications. D.C. Heath and Company, Lexington.

Dossett, R.J. (1986). Using data collectors with the IBM-PC for work measurement. Computers and Industrial Engineering 11, 476-480.

Evers, K.H.; Bachert, R.F.; and Santucci, P.R. (1981). SADT/SAINT: large scale analysis simulation methodology. 1981 Winter Simulation Conference Proceedings. 185-191.

Gray, P. (1978). Managing simulation projects. Proceedings of the 1978 Winter Simulation Conference. 63-68.

Henricksen, J.O. (1984). Discrete event simulation languages: current status and future directions. Proceedings of the 1984 Winter Simulation Conference. 83-88.

Hollocks, B. (1984). The impact of the microcomputer in simulation. Operational Research 1984 Proceedings of the Tenth International Conference.

Huhn, R.M. and Comer, E.R. (1981). applications of modern software techniques to modeling and simulation. 1981 Winter Simulation Conference Proceedings. 101-112.

Jones, E. (1985). Using dBASE III. McGraw-Hill, Berkeley.

Kelton, W.D. (1984). Input data collection and analysis. Proceedings of the 1984 Winter Simulation Conference. 91-95.

Ketcham, M.G.; Hogg, G.L.; and Shannon, R.E. (1985). Memory management, data structures, and the development of advanced simulation software in a microcomputer environment. Modeling and Simulation on Microcomputers: 1985. Society for Computer Simulation, 67-70.

MacMillan, J.S. and Walker, A.P. (1985). Portable microcomputer function as data controllers. Industrial Engineering. 18-26.

Martin, F.F. (1968). Computer Modeling and Simulation. John Wiley and Sons, New York.

Miller, A. and Pare, J. (1986). Managing simulation projects. 1986 Annual Simulation Symposium. 43-57.

Nance, R.E. (1983). A tutorial view of simulation model development. Proceedings of the 1983 Winter Simulation Conference. 16-20.

Nance, R.E. Chairman's corner - when you are hot and when you are not. Simuletter. 2-3.

Nance, R.E.; Mezaache, A.L.; and Overstreet, C.M. (1981). Simulation model management: resolving the technology gaps. 1981 Winter Simulation Conference Proceedings. 173-179.

Oren, T.I. and Zeigler, B.P. (1979). Concept for advanced simulation methodologies. Simulation. 69-81.

Pritsker, A.A. (1986). Introduction to Simulation and SLAM. John Wiley and Sons, New York.

Redman, S. (1985). User Friendly Guide to Lap Portables. McGraw-Hill, New York.

Schmidt, J.S. (1985). Introduction to system analysis, modeling and simulation. Proceedings of the 1985 Simulation Conference. 3-13.

Schruben, L. (1983). Modeling systems using discrete-event simulation. Proceedings of the 1983 Winter Simulation Conference. 101-107.

Shannon, R.E. (1975). System Simulation: The Art and Science. Printice-Hall, New Yersey.

Standridge, C.R. (1981). Using the simulation data language (SDL). Simulation. 73-81.

Standridge, C.R. (1981). The simulation data language (SDL): applications and examples. Simulation. 119-130.

Standridge, C.R. (1985). Performing simulation projects with the extended simulation system. Simulation. 283-291.

Standridge, C.R. and Walker, S.A. (1984). An introduction to simulation support software. Proceedings of the 1983 Winter Simulation Conference. 381-384.

Standridge, C.R. (1986). Animating simulation using TESS. Computers and Industrial Engineering 10. 121-134.

Standridge, C.R.; Vaughan, D.K.; and Sale, M.L. (1985). A tutorial on TESS: the extended simulation system. Proceedings of the 1985 Winter Simulation Conference. 73-79.

Waite, M. and Morgan, C.L. (1983). Graphics Primer for the IBM PC. McGraw-Hill, Berkeley.

Wilkerson, D.L. (1984). Computer aided standard data development for work measurement. Computerized Work Measurement. Industrial Engineering & Management Press, Norcross.

Wilkinson, J.E. (1987). The Use of a Portable Microcomputer as a Data Collection Tool to Support Integrated Simulation Environments: A Concept. Masters Thesis, University of Central Florida, Orlando, Florida.

Womack, J.W. and Marr, G.R. (1985). Desktop data bases for simulation development and documentation. Modeling and Simulation On Microcomputers. Society for Computer Simulation, 63-66.

Wygant, R.M. (1986). Data file management. Proceedings of the 8th Annual Conference on Computers and Industrial Engineering. 367-371.

AUTHORS' BIOGRAPHIES

JAMES E. WILKINSON is a Major in the United States Air Force. He is presently stationed at Scott Air Force Base, Illinois. His B.S. is from North Dakota State University. He recently received his M.S. in Industrial Engineering from University of Central Florida. This paper is a synopsis of Major Wilkenson's Master's Thesis.

Major James E. Wilkinson
2192-A Birchwood Terrace
Scott Air Force Base, Illinois 62225
(618) 256-2921

GARY E. WHITEHOUSE, professor of Industrial Engineering and Management Systems at the University of Central Florida, has served as consultant to industry and has published numerous books and articles on applications of operations research, computer applications, and network modeling. A Fellow of the Institute of Industrial Engineers, he is also editor of the Industrial Engineering's Mini/Micro Computer Column and IIE Micro Software Series. He is Director of ASEE's Engineering Management Division.

Gary E. Whitehouse, Ph.D., P.E.
University of Central Florida
College of Engineering - IEMS Department
Orlando, Florida 32816
(305) 281-5020