ESD-TDR-64-169

SR-125

THE ROLE OF SIMULATION AND DATA REDUCTION PROGRAMS IN THE DEVELOPMENT OF REAL-TIME SYSTEMS

TECHNICAL DOCUMENTARY REPORT NO. ESD-TDR-64-169

DECEMBER 1964

E. L. Lafferty

Prepared for

416L/M/N SYSTEM PROGRAM OFFICE
ELECTRONIC SYSTEMS DIVISION
AIR FORCE SYSTEMS COMMAND
UNITED STATES AIR FORCE
L. G. Hanscom Field, Bedford, Massachusetts

Project 416. 1

Prepared by

THE MITRE CORPORATION
Bedford, Massachusetts
Contract AF 19(628)-2390

ESD-TDR-64-169

SR-125

THE ROLE OF SIMULATION AND DATA REDUCTION PROGRAMS IN THE DEVELOPMENT OF REAL-TIME SYSTEMS

TECHNICAL DOCUMENTARY REPORT NO. ESD-TDR-64-169

DECEMBER 1964

E. L. Lafferty

Prepared for

416L/M/N SYSTEM PROGRAM OFFICE
ELECTRONIC SYSTEMS DIVISION
AIR FORCE SYSTEMS COMMAND
UNITED STATES AIR FORCE
L. G. Hanscom Field, Bedford, Massachusetts

Project 416.1

Prepared by

THE MITRE CORPORATION
Bedford, Massachusetts
Contract AF 19(628)-2390

Copies available at Office of Technical Services, Department of Commerce.

Qualified requesters may obtain copies from DDC. Orders will be expedited if placed through the librarian or other person designated to request documents from DDC.

When US Government drawings, specifications, or other data are used for any purpose other than a definitely related government procurement operation, the government thereby incurs no responsibility nor any obligation whatsoever; and the fact that the government may have formulated, furnished, or in any way supplied the said drawings, specifications, or other data is not to be regarded by implication or otherwise, as in any manner licensing the holder or any other person or corporation, or conveying any rights or permission to manufacture, use, or sell any patented invention that may in any way be related thereto.

Do not return this copy. Retain or destroy.

ESD-TDR-64-169 SR-125

THE ROLE OF SIMULATION AND DATA REDUCTION PROGRAMS IN THE DEVELOPMENT OF REAL-TIME SYSTEMS

TECHNICAL DOCUMENTARY REPORT NO. ESD-TDR-64-169

DECEMBER 1964

E. L. Lafferty

Prepared for

416L/M/N SYSTEM PROGRAM OFFICE
ELECTRONIC SYSTEMS DIVISION
AIR FORCE SYSTEMS COMMAND
UNITED STATES AIR FORCE
L. G. Hanscom Field, Bedford, Massachusetts

Project 416.1

Prepared by

THE MITRE CORPORATION
Bedford, Massachusetts
Contract AF 19(628)-2390

THE ROLE OF SIMULATION AND DATA REDUCTION PROGRAMS IN THE DEVELOPMENT OF REAL-TIME SYSTEMS

ABSTRACT

This report deals with the valuable use of simulation and data reduction computer programs in the acquisition and engineering of command and control systems. The value of simulations, especially in facilitating the learning process and in expediting system design, is described. Data reduction is shown to be an evolutionary process and the design of a data reduction system should be considered in the very early stages of system acquisition. Some model simulation and data reduction system software are examined and several considerations in their design are enumerated. The importance of the system engineer's recognition of the constantly changing nature of all his instrumentation is stressed as being all-important in the design of support systems which provide an overall effectiveness.

REVIEW AND APPROVAL

Publication of this technical documentary report does not constitute Air Force approval of the reports findings or conclusions. It is published only for the exchange and stimulation of ideas.

JOHN A. TRASK, COLONEL, USAF
System Program Director
416L/M/N System Program Office

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

SECTION I

INTRODUCTION

The purpose of this report is to describe the uses of simulation and data reduction computer programs in the acquisition of a command and control system. An attempt will be made to define some of the uses to which these programs have been put, to describe various types of tools and to show how programming techniques for producing these tools have been developed to achieve an over-all effectiveness.

SECTION II

GENERAL LOOK AT SUPPORT SYSTEMS

Ancillary to the operational program of a command and control system are many varied computer programs which are often overlooked. In this class of programs falls the utility, maintenance, simulation and data reduction programs. In many instances these support programs are prepared on a machine different from the operational computer, since one cannot usually assume that a prototype operational computer is available during the early stages of program production and when programs such as compilers, assemblers, experimental simulations and the like are required (Fig. 1).

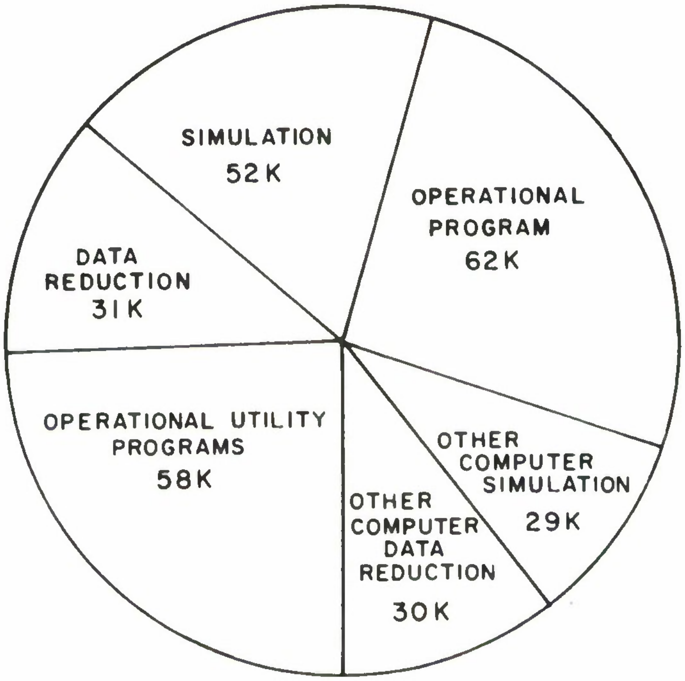

Fig. 1 Total System Software

The total size of the support programs exceeds that of the operational computer program significantly. Some of the support programs remain relatively fixed throughout the life of the system, as, for example, the utility programs. However, some programs, such as simulation and data reduction, may vary as often as, and in some instances more often than, the operational program itself. Such variation must be taken into account when discussing the size or relative size of a program. In the case of a fixed-size operational program such as SAGE or BUIC, experience has shown that the actual number of instructions programmed throughout a given time period may be many times the length of the actual program. This, of course, makes it difficult to cost programs in terms of dollars per instruction since, for example, a program of 30,000 instructions will increase in cost each year that the program remains operational and variable, but yet will remain the same in terms of fixed length, more or less. The same principles apply to some of the support subsystems.

SECTION III

SIMULATION AND ITS USES

For the purposes of this report, a model is defined as a program representation of a system or a subsystem, or its environment; the use of a model is called a simulation. Models for simulation have been built and used in many different areas of systems engineering for a variety of purposes. The most obvious use of a simulation is in the initial design stages of the command and control system where it is desired to experiment with new techniques or to improve old ones at a reasonable cost. A little later in the design cycle, when a particular design has been decided upon, it may be desired to verify this design with a larger sample or to optimize parameters for implementation. And finally, even after the acquisition of the system itself following live environment testing, it may be desired to evaluate the over-all system concept with a simulation. In this post-acquisition stage, it is assumed that the simulation has undergone some form of calibration such that it represents as closely as possible the operational system. This may be done by utilizing data collected during live environment tests to form the basic parameters of the simulation.

In this manner, it is then possible to extrapolate from a relatively small sample of data by the use of Monte Carlo methods and numerous replications. For example, many command and control systems have, as basic elements, destructable materials or environmental conditions too expensive to reproduce such as the firing of a missile or the conditions of a nuclear war. In some cases, small data samples may be collected, as, for example, a small number of missile shots can usually be expected during the Category II test phase. This data can then be used to calibrate a simulation so that literally thousands of simulated missiles can be fired to get a better estimate of the system capability.

Needless to say, the results obtained with a simulation are only as good as the success achieved in calibrating the model. It is, therefore, very important that accurate calibration be performed prior to collecting the large volume of data usually associated with an evaluation.

SECTION IV

SIMULATION DEVELOPMENT

The first step in simulation is to determine if a simulation is really needed and what it will do. The analysis plan must be formulated before embarking on a costly simulation program. After this is done and after it has been decided to simulate, the next step is an analysis of the structure of the system in order to determine which parameters are critical to the prediction and which are not. In some cases, parameters may be approximated or even eliminated as non-essential to the results expected. As an example, if one is trying to determine the accuracy necessary to guide a missile so as to achieve successful interception, it is of course necessary to represent the target tracking, missile tracking, missile guidance, as well as the missile performance. However, it may be permissible to ignore the effect of the human operator who passively monitors the interception since in very few cases can he or does he interfere.

After the actual models have been constructed and integrated into the simulated system, the calibration is now performed. Assume that the model is one of a SAGE-like tracking program and that some live track data has been collected, such as position and velocity throughout the life of the track and the radar data used to obtain these positions and velocities. The calibration is performed by describing to the radar model the flight path of the aircraft. The simulation is then run, and position and velocity information are extracted. These positions and velocities are compared with the observations and parameters in the tracking and/or radar models such as the smoothing constants, error deviations, etc., adjusted until a reasonable error exists. The average should closely represent the true target performance.

If it is found that, after sufficient repetitions of this process, it is impossible to duplicate the live environment closely enough, it may be necessary to examine the design of the model. Considerations are given to program errors, to loose approximations where more accurate ones are required, or to general design inconsistency with the true case. It should be noted that the calibration process can be time consuming, but is a necessary step in obtaining confidence in the simulated results; i. e., the results are only as good as the extent of calibration performed.

During the development of the SAGE system, simulations were used on a number of occasions to assist in the design process. The first full-scale simulation activity was started in order to develop the specifications for the integration of BOMARC into SAGE. A few early programs were produced to determine the necessary parameters for missile guidance. A simulation was produced to study the problems associated with the integration of Army weapons. A little later, when it was desired to integrate an airborne radar, modifications were made to these systems. After many modifications of these packages to enable special studies to be performed, it was decided to produce a modular all-purpose simulation program which was called STAPP.* The BOMARC and airborne radar simulations were modeled in the STAPP system and since then, a number of full-scale simulation activities have progressed.

A description of one of these activities will show the amount of effort applied and the results obtained. The primary goal was to determine the effects of varying environments on the performance of the BOMARC missile so that an intelligent employment could be made and the missile used effectively. The

* Initially this meant Single Thread All Purpose Program

simulation vehiele eonsisted of about 30, 000 words of storage on the 7090, ineluding tables. The model was a one-on-one simulation; i. e. , it involved a single target and a single weapon. An additional 10, 000 instructions were added to assist in ealibration. These additional units consisted primarily of data eolleetion and analysis programs. The programming effort eonsisted of approximately two man-years spread over a six-month period.

Following the produetion of the system, a calibration effort was begun whereby data colleeted during live environment testing was eompared with simulation runs. The ealibration effort was run eoneurrently with the live testing and, therefore, was dependent on the test schedule for its completion. The total manpower involved was approximately one-half a man-year over a one-year period. Computer time was used at a rate of about 20 hours per month. The evaluation task eonsisted of numerous replications with varying target environments and radar configurations. Some of the parameters examined were target speed, target altitude, traek erossing angle, radar registration, radar data quality and a selection of maneuvers at different times during the intereept. The probability of sueeessful intereept was computed over a series of Monte Carlo runs.

The evaluation aetivity lasted for approximately six months, ineluding the preparation of the final report. Computer time was used at a rate of about 80 hours per month, and the task was eondueted by one man. The most important information obtained from this aetivity eonsisted of the determination of numerieal values for what were formerly intuitive insights. For example, the output of the simulation provided a graph of probability of sueeess plotted against time-to-go intereept for a family of maneuvers. The same type of numerieal values were found for eritieal target speeds, target altitudes, data quality and the like.

SECTION V

ADVANTAGES AND DANGERS IN SIMULATING

It seems appropriate, at this time, to summarize some of the advantages and disadvantages of simulation.

(a) Simulation permits experimentation without the risks and costs involved in dealing with the real thing.

(b) Simulation often permits the demonstration of system operation before the hardware is built. It is also possible, at times, to use such a model for the training of operators for the real system or for setting up procedures to be used in the field.

(c) A large number of replications may be performed considerably faster than on comparable operational equipment.

(d) Environmental conditions, system parameters, and subsystem operating characteristics may be varied quickly and easily in many simulation models and at a lower cost.

(e) Simulation generally has beneficial "fallout;" i. e., quite often many questions are answered which are not really asked.

(f) A much higher degree of control over environmental conditions and time is possible in a simulation than in a real-world environment.

(g) The very task of constructing a simulation model gives additional and detailed insight into the system design and, as a byproduct, may produce more competently trained analysts.

As with any technique, simulation has its dangers as well as advantages. It is not always a faster, cheaper way of doing the job. Certain processes or tasks are much better handled analytically or by prototype testing. Hence, one of the primary dangers in simulation is, in fact, the use of simulation at all; i. e., its use in cases where other methods are indicated. Some specific pitfalls which can be encountered in the design of models are as follows:

(a) It is not always possible to foresee all the variables needed, thus resulting in either important omissions or in such a large number of variables that the analysis task is hopeless or simulation results worthless.

(b) It is possible to overdesign, concentrating too much on approximation of the real world and not enough on the problem to be solved.

(c) In many cases, there is too little emphasis placed on the effect on the man in the system or the adequate representation of his performance.

(d) Poor programming techniques, especially in the broad program design phase, may produce a simulation which is so limited and complex that minor perturbations may render it useless for analysis.

SECTION VI

SIMULATION TYPES

The world of simulation possesses many different categories of tools for performing different tasks. One way of classifying simulations is to state dichotomies as follows:

(a) Simulation of the environment versus simulation in a real environment; in this case, the total simulation internal to the computer, also facts of the environment, and the preparation of models of all environmental entities. In many cases, it is possible to simulate only the data input to the system while using real-world environmental entities to process them. An example of the latter is the simulation of intercept computations and interceptor performance against real-tracked target data.

(b) Models of non-computer systems constructed on a computer as opposed to models which use a computer in the way which is used in the real system. In the latter case, one may use the operational computer or a different computer to simulate the operational computer.

(c) Analytic versus real-time simulations. The analytic simulation is generally not time-sequenced or time is compressed. Simulations may also be classified by use or purpose such as for design experimentation, evaluation, training, or the study of human reactions. Other classifications are possible, such as deterministic versus Monte Carlo, a man-machine versus pure machine simulation, and the like. What is important, however, is to know what techniques are available and where they should be used. More difficult, but as important, is to determine to what level of sophistication one must duplicate the real world to achieve the desired result. The latter requires a careful analytic study of the problem and the model.

SECTION VII

DATA REDUCTION

Up to now, the discussion has centered on simulation and its uses. Throughout the entire acquisition phase of a command and control system, data reduction programs are required. At first they are required to support the simulation activity and, in many cases can, in fact, be a part of the simulation program itself, whilc later the data reduction is expanded to include the processing of data collected during a live environment test. In general, one can think of thc data reduction or analysis programs as being evolutionary for a given system; i. e., at first, when little is known about the system design or system techniques, onc is interested in looking at every minute detail of its operation. This type of program is generally unsophisticated in nature, and consists merely of the retrieval and organization of data in a form easily handled by analysts. As the command and control system evolves, so too does the data reduction in that the analysts are no longer interested in seeing all of the details but want them compressed and analyzed in the manner that the analysts themselves have used earlier. If this fact is realized, it can be used to great advantage in the design of the first generation of processing programs. Even more important is the fact that once data is analyzed in a certain way a number of times and a conclusion reached, it is quite unlikely that the data will continue to be processed in the same way. To put it simply, analysts want new programs often. If this is provided for in the design of the system, many future problems can be avoided.

The goals, then, toward which one aims in creating a system of data reduction programs are very similar to those for any large computer system. The trade-offs, however, are different. Whereas in a real-time system one aims for program efficiency, in post-test data reduction one aims for speed and accuracy of program preparation. This should, of course, include the ability to change the programs quickly and with a minimum of turnaround. The next most important consideration is reliability or confidence in the results--there is no more aggravating occurrence than discovering that an alleged "system error" was in fact an error in the instrumentation. Finally, efficiency of a sort must be considered, especially where the reduction of large volumes of data are involved--bearing in mind that reruns caused by unreliability are the most rapacious consumers of time in a limited use program and the tradeoff of reliability for efficiency can be dangerous.

Still on the subject of efficiency, a distinction between "design" efficiency and "code" efficiency should be made. In general, for any programming task, it is futile to apply extensive code optimization to a basically unsound or inefficient design. This trap is particularly dangerous in data reduction where many different analysis tasks may be performed on the same set of data. For example, rather than spending effort on code optimization, a method of reducing the number of passes through the data should be considered.

SECTION VIII

TECHNIQUES FOR SIMULATION AND DATA REDUCTION PROGRAMMING

Approximately eight years ago when the SAGE system was in its infancy, simulation and data analysis tools were produced as they were needed to perform the varying tasks necessary in the development. At one time, a count was made of the total number of instructions that were being maintained for use by analysts and designers. The number exceeded 100,000, and was growing each day. In addition, it was learned that much of the new programming work involved changes to the old programs. With the normal turnover of programming personnel, it became increasingly difficult to provide a staff of programmers that were intimately familiar with all of the programs and the inventory. The maintenance task became overwhelming and inefficiencies resulted because of program unreliability. The programs were all separate entities, each performing its own function and in no way linked with other programs in the inventory. This was wasteful since it is obvious that many programs can be combined during a single run in order to cut down tape-passage time (see Fig. 2).

Once some design goals have been established for an ideal simulation or data reduction system, it is necessary to examine the techniques for designing the basic system software. Data reduction and simulation have been common features. For example, they usually operate in non-real time and, therefore, the importance of code efficiency is less. Another similarity is that the total data base for these functions is usually large and, in many respects, quite similar in structure, the major difference being that the simulation data base is created by the program whereas the data reduction base is provided as an input.

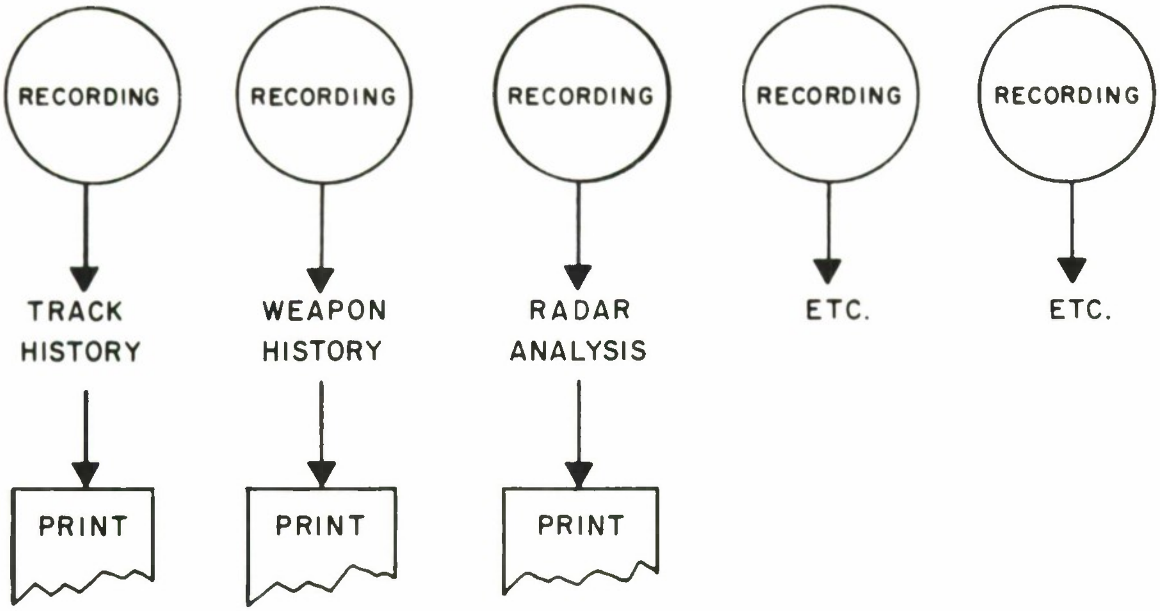

Fig. 2 Data Reduction Techniques— Old System

For both data reduction and simulation, the first requirement is a language in which programs can be written quickly and accurately, and, at the same time, can be linked structurally with other programs embodied in the system. A language similar to FORTRAN or JOVIAL and which contains the capability to specify data definitions externally is a must. In addition, the language should contain the necessary machinery for filing and retrieving common routines for repeated use. The retrieval process should be an integral part of the language such that the programmer does not always have to specify an operation to be performed if that operation, in fact, is implicit in the data specification.

For simulation programming, the most valuable feature which can be provided is the ability to specify in user language the necessary parameters which are to be varied from run to run. This language too may be the FORTRAN or JOVIAL type and must itself access the dictionary utilized in the compile process. In this way, the user can communicate with the simulation quickly and accurately.

In the simulation programming particularly, a feature which has been found most important is modularity. Early simulations were, for the most part, self-contained packages which intercommunicated within themselves but which were of little or no value in any other environment. The STAPP simulation system links modules together by means of an external data base specification and has provided the ability to utilize previously checked out units in new environments for new jobs. An example of this is in the BOMARC simulation where it was necessary to model SAGE radar inputs and SAGE tracking, and to combine these with a BOMARC guidance and missile model. When the second generation of BOMARC's, i. e. , the BOMARC B, had to be evaluated, it was possible to utilize all of the SAGE air surveillance programming in conjunction with a new guidance and missile model. Even some of the data reduction programming was useful (see Fig. 3).

For maximum effectiveness in data reduction, a system whereby multiple programs may access the recorded data file is desirable. Since the number and function of the programs may vary from time to time, the system should be designed such that programs can be selected from a library by the user. Some interface between the input data and the program is required. One way to simplify the programming task is to assume that the input data will be converted to a standard format and to program all routines for this format. In order to achieve this, one could prepare a separate conversion program for each new data input structure, or, on a more sophisticated level, design a general purpose conversion program which would access a structure specification and perform the conversion of any format of the input data.

Another important way to achieve over-all effectiveness in data reduction and simulation programming is to isolate certain general functions such as sorting and editing of output data and to perform these wherever possible by general

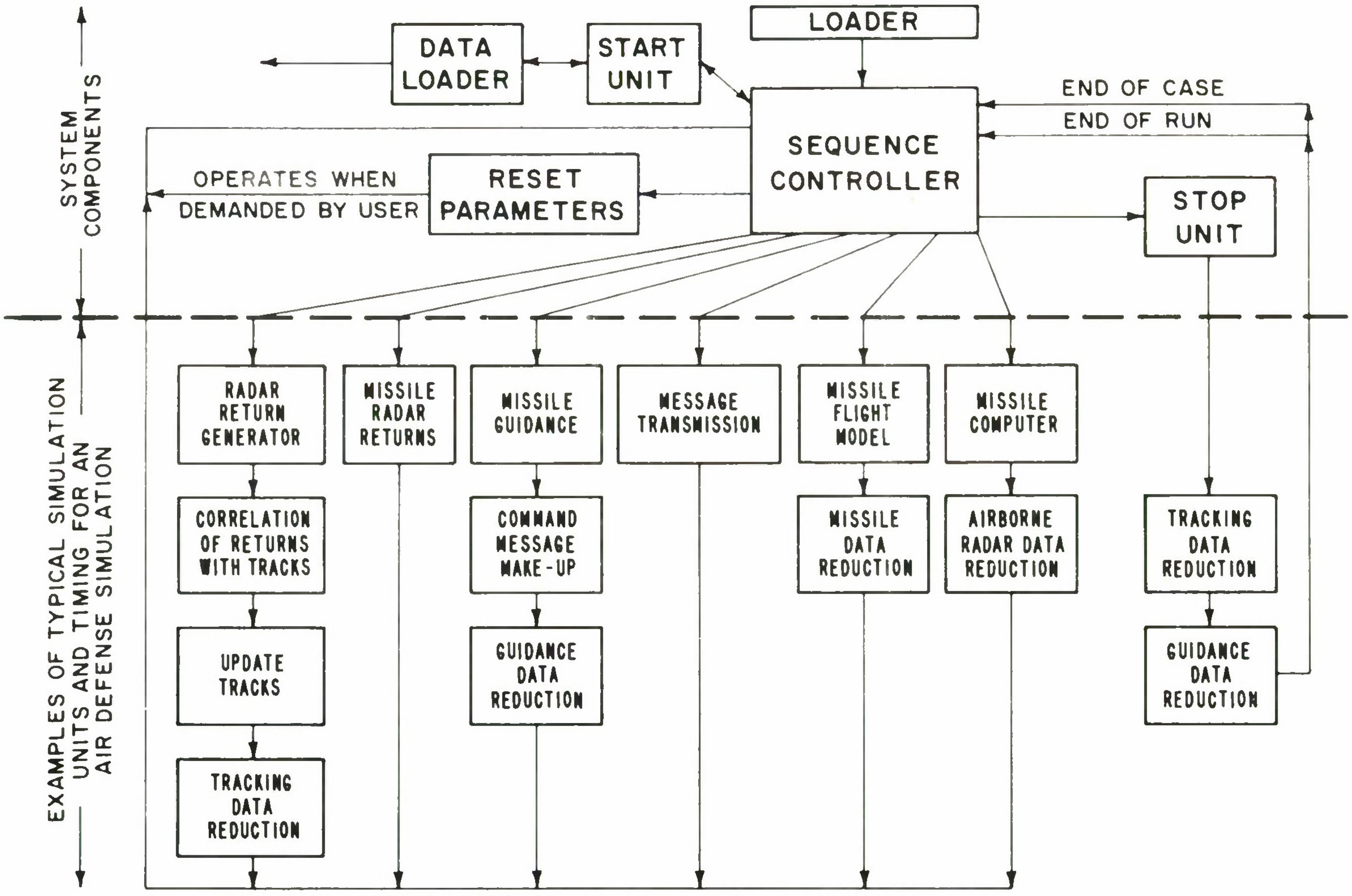

Fig. 3 Organization of STAPP System

purpose programs. This technique requires that the data manipulation programs prepare their outputs in formats acceptable to the general purpose sort and edit programs. The sort and edit programs should be code-optimized as much as possible since they will be used quite often and for large volumes of data. In addition, their design specifications should be flexible so that they can perform a wide range of functions. Other isolable functions, such as random number generation, sequence control, integration, differentiation, etc., may also be treated in this manner (see Fig. 4).

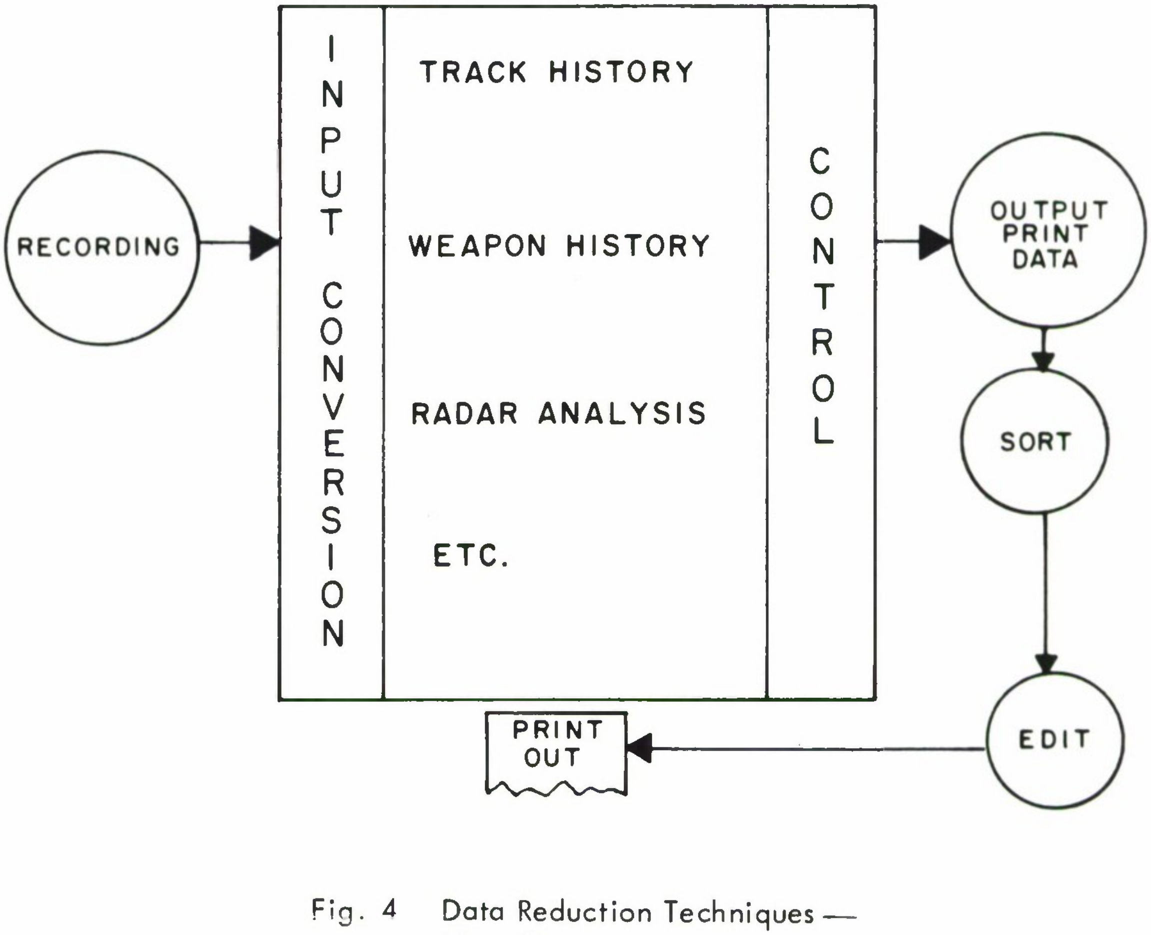

Fig. 4 Data Reduction Techniques — New System

SECTION IX

SUMMARY

The value of simulation and data reduction in support of system acquisition and engineering have been discussed. Data reduction is an evolutionary process, and the design of a data reduction system should be considered in the very early stages of system acquisition. It has been shown that simulations, in general, can be quite valuable, especially as amplification to the learning process and as a quick and effective method for system design, but that careful consideration is required before embarking on a simulation project to assure that its use is justified and that, in fact, the required results will be achieved. Some model simulation and data reduction system software have been described, and several considerations in their design have been enumerated. One last summation point which should be made is that the system designer should assume that all of his instrumentation, and, in particular, that discussed in this report, is subject to violent and constant change. This consideration is all-important in the design of these support systems if they are to provide effective and continuing aid to the system engineer.

Security Classification

DOCUMENT CONTROL DATA - R&D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

1. ORIGINATING ACTIVITY *(Corporate author)* The MITRE Corporation Bedford, Massachusetts	2a. REPORT SECURITY CLASSIFICATION Unclassified 2b. GROUP

3. REPORT TITLE

The Role of Simulation and Data Reduction Programs in the Development of Real-Time Systems

4. DESCRIPTIVE NOTES *(Type of report and inclusive dates)*

NA

5. AUTHOR(S) *(Last name, first name, initial)*

Lafferty, Edward L.

6. REPORT DATE December 1964	7a. TOTAL NO. OF PAGES 24	7b. NO. OF REFS 0

8a. CONTRACT OR GRANT NO. Contract AF 19(628)-2390 b. PROJECT NO. 416L c. d.	9a. ORIGINATOR'S REPORT NUMBER(S) ESD-TDR-64-169 9b. OTHER REPORT NO(S) *(Any other numbers that may be assigned this report)* SR-125

10. AVAILABILITY/LIMITATION NOTICES

Qualified requestors may obtain from DDC.
DDC release to OTS authorized

11. SUPPLEMENTARY NOTES	12. SPONSORING MILITARY ACTIVITY 416L/M/N System Program Office L. G. Hanscom Field Bedford, Massachusetts

13. ABSTRACT

This report deals with the valuable use of simulation and data reduction computer programs in the acquisition and engineering of command and control systems. The value of simulations, especially in facilitating the learning process and in expediting system design, is described. Data reduction is shown to be an evolutionary process and the design of a reduction system should be considered in the very early stages of system acquisition. Some model simulation and data reduction system software are examined and several considerations in their design are enumerated. The importance of the system engineer's recognition of the constantly changing nature of all his instrumentation is stressed as being all-important in the design of support systems which provide an overall effectiveness.

Security Classification

14. KEY WORDS	LINK A		LINK B		LINK C	
	ROLE	WT	ROLE	WT	ROLE	WT
Digital Computers						
Data Reduction						
Simulation						
Real-Time Systems, Development						

INSTRUCTIONS

1. ORIGINATING ACTIVITY: Enter the name and address of the contractor, subcontractor, grantee, Department of Defense activity or other organization *(corporate author)* issuing the report.

2a. REPORT SECURITY CLASSIFICATION: Enter the overall security classification of the report. Indicate whether "Restricted Data" is included. Marking is to be in accordance with appropriate security regulations.

2*b*. GROUP: Automatic downgrading is specified in DoD Directive 5200.10 and Armed Forces Industrial Manual. Enter the group number. Also, when applicable, show that optional markings have been used for Group 3 and Group 4 as authorized.

3. REPORT TITLE: Enter the complete report title in all capital letters. Titles in all cases should be unclassified. If a meaningful title cannot be selected without classification, show title classification in all capitals in parenthesis immediately following the title.

4. DESCRIPTIVE NOTES: If appropriate, enter the type of report, e.g., interim, progress, summary, annual, or final. Give the inclusive dates when a specific reporting period is covered.

5. AUTHOR(S): Enter the name(s) of author(s) as shown on or in the report. Enter last name, first name, middle initial. If military, show rank and branch of service. The name of the principal author is an absolute minimum requirement.

6. REPORT DATE: Enter the date of the report as day, month, year; or month, year. If more than one date appears on the report, use date of publication.

7*a*. TOTAL NUMBER OF PAGES: The total page count should follow normal pagination procedures, i.e., enter the number of pages containing information.

7*b*. NUMBER OF REFERENCES: Enter the total number of references cited in the report.

8*a*. CONTRACT OR GRANT NUMBER: If appropriate, enter the applicable number of the contract or grant under which the report was written.

8*b*, 8*c*, & 8*d*. PROJECT NUMBER: Enter the appropriate military department identification, such as project number, subproject number, system numbers, task number, etc.

9*a*. ORIGINATOR'S REPORT NUMBER(S): Enter the official report number by which the document will be identified and controlled by the originating activity. This number must be unique to this report.

9*b*. OTHER REPORT NUMBER(S): If the report has been assigned any other report numbers *(either by the originator or by the sponsor)*, also enter this number(s).

10. AVAILABILITY/LIMITATION NOTICES: Enter any limitations on further dissemination of the report, other than those imposed by security classification, using standard statements such as:

(1) "Qualified requesters may obtain copies of this report from DDC."

(2) "Foreign announcement and dissemination of this report by DDC is not authorized."

(3) "U. S. Government agencies may obtain copies of this report directly from DDC. Other qualified DDC users shall request through

________________________________ ."

(4) "U. S. military agencies may obtain copies of this report directly from DDC. Other qualified users shall request through

________________________________ ."

(5) "All distribution of this report is controlled. Qualified DDC users shall request through

________________________________ ."

If the report has been furnished to the Office of Technical Services, Department of Commerce, for sale to the public, indicate this fact and enter the price, if known.

11. SUPPLEMENTARY NOTES: Use for additional explanatory notes.

12. SPONSORING MILITARY ACTIVITY: Enter the name of the departmental project office or laboratory sponsoring *(paying for)* the research and development. Include address.

13. ABSTRACT: Enter an abstract giving a brief and factual summary of the document indicative of the report, even though it may also appear elsewhere in the body of the technical report. If additional space is required, a continuation sheet shall be attached.

It is highly desirable that the abstract of classified reports be unclassified. Each paragraph of the abstract shall end with an indication of the military security classification of the information in the paragraph, represented as *(TS)*, *(S)*, *(C)*, or *(U)*.

There is no limitation on the length of the abstract. However, the suggested length is from 150 to 225 words.

14. KEY WORDS: Key words are technically meaningful terms or short phrases that characterize a report and may be used as index entries for cataloging the report. Key words must be selected so that no security classification is required. Identifiers, such as equipment model designation, trade name, military project code name, geographic location, may be used as key words but will be followed by an indication of technical context. The assignment of links, rules, and weights is optional