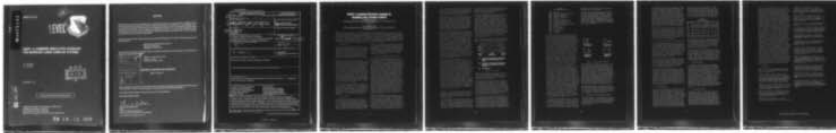


AD-A075 144

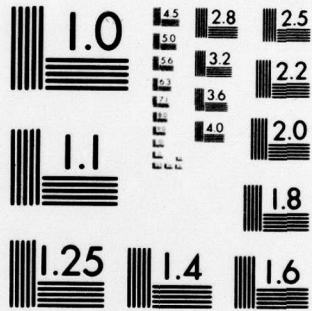
AEROSPACE MEDICAL RESEARCH LAB WRIGHT-PATTERSON AFB OH F/G 9/2  
SAINT: A COMBINED SIMULATION LANGUAGE FOR MODELLING LARGE, COMP--ETC(U)  
OCT 78 D J SEIFERT, G P CHUBB  
UNCLASSIFIED AMRL-TR-78-48

NL

| OF |  
AD  
A075144



END  
DATE  
FILMED  
11 -79  
DDC



MICROCOPY RESOLUTION TEST CHART  
NATIONAL BUREAU OF STANDARDS-1963-A

ADA075144

AMRL-TR-78-48

*RB*

**LEVEL**



**SAINT: A COMBINED SIMULATION LANGUAGE  
FOR MODELING LARGE COMPLEX SYSTEMS**

*D. J. SEIFERT  
G. P. CHUBB*

DDC  
RECEIVED  
OCT 17 1979  
A

OCTOBER 1978

DDC FILE COPY.

Approved for public release; distribution unlimited.

AEROSPACE MEDICAL RESEARCH LABORATORY  
AEROSPACE MEDICAL DIVISION  
AIR FORCE SYSTEMS COMMAND  
WRIGHT-PATTERSON AIR FORCE BASE, OHIO 45433

79 10 16 033

## NOTICES

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.

Please do not request copies of this report from Aerospace Medical Research Laboratory. Additional copies may be purchased from:

National Technical Information Service  
5285 Port Royal Road  
Springfield, Virginia 22161

Federal Government agencies and their contractors registered with Defense Documentation Center should direct requests for copies of this report to:

Accession for	
NTIS Grant	<input checked="" type="checkbox"/>
DDC TAB	<input type="checkbox"/>
Unannounced	<input type="checkbox"/>
Justification	
By _____	
Distribution/	
Availability Codes	
Dist.	Avail and/or special
A	

Defense Documentation Center  
Cameron Station  
Alexandria, Virginia 22314

### TECHNICAL REVIEW AND APPROVAL

ANRL-TR-78-48

This report has been reviewed by the Information Office (OI) and is releasable to the National Technical Information Service (NTIS). At NTIS, it will be available to the general public, including foreign nations.

This technical report has been reviewed and is approved for publication.

FOR THE COMMANDER



CHARLES BATES, JR.  
Chief  
Human Engineering Division  
Aerospace Medical Research Laboratory

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM										
1. REPORT NUMBER <b>AMRL-TR-78-48</b>	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER										
4. TITLE (and Subtitle) <b>SAINT: A COMBINED SIMULATION LANGUAGE FOR MODELLING LARGE, COMPLEX SYSTEMS.</b>	5. TYPE OF REPORT & PERIOD COVERED <b>Proceedings Paper</b>											
7. AUTHOR(s) <b>D. J. SEIFERT G. P. CHUBB</b>	8. CONTRACT OR GRANT NUMBER(s)											
9. PERFORMING ORGANIZATION NAME AND ADDRESS <b>Aerospace Medical Research Laboratory Aerospace Medical Div, Air Force Systems Command Wright-Patterson Air Force Base, Ohio 45433</b>	10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS <b>62202F, 7184</b>											
11. CONTROLLING OFFICE NAME AND ADDRESS <b>12) 82</b>	12. REPORT DATE <b>September 1978</b>											
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office) <b>11) Oct 78</b>	13. NUMBER OF PAGES <b>8</b>											
16. DISTRIBUTION STATEMENT (of this Report) <b>Approved for public release; distribution unlimited</b>		15. SECURITY CLASS. (of this report) <b>Unclassified</b>										
15a. DECLASSIFICATION/DOWNGRADING SCHEDULE												
17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)												
18. SUPPLEMENTARY NOTES <b>Presented at the Simulation Special Interest Group 1978 Conference, 4 - 6 September 1978, in Canberra, Australia.</b>												
19. KEY WORDS (Continue on reverse side if necessary and identify by block number) <table border="0"> <tr> <td>Combined simulation language</td> <td>Task analysis</td> </tr> <tr> <td>Discrete event simulation</td> <td>Manned systems modeling</td> </tr> <tr> <td>Continuous process simulation</td> <td>Crew performance modeling</td> </tr> <tr> <td>Monte Carlo simulation</td> <td>Human factors engineering</td> </tr> <tr> <td>Network models</td> <td>Human engineering evaluation</td> </tr> </table>			Combined simulation language	Task analysis	Discrete event simulation	Manned systems modeling	Continuous process simulation	Crew performance modeling	Monte Carlo simulation	Human factors engineering	Network models	Human engineering evaluation
Combined simulation language	Task analysis											
Discrete event simulation	Manned systems modeling											
Continuous process simulation	Crew performance modeling											
Monte Carlo simulation	Human factors engineering											
Network models	Human engineering evaluation											
20. ABSTRACT (Continue on reverse side if necessary and identify by block number) <b>SAINT is a FORTRAN based, network-oriented, combined simulation language for modeling large, complex systems with continuous and discrete event components that interact and possess time varying properties. The acronym stands for Systems Analysis of Integrated Networks of Tasks. The paper presents an overview of the basic features and concepts incorporated into SAINT and reviews some of the applications made to date. References are provided to the reports documenting SAINT and its applications.</b>												

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

009 850

set

# SAINT: A Combined Simulation Language for Modelling Large, Complex Systems

D. J. SEIFERT, B.S., M.S.

6570 Aerospace Medical Research Laboratory, Systems Research Branch, Wright-Patterson Air Force Base, Ohio, USA

and

G. P. CHUBB, B.S., M.A.

6570 Aerospace Medical Research Laboratory, Crew Station Integration Branch, Wright-Patterson Air Force Base, Ohio, USA

**SUMMARY** SAINT is an acronym for: Systems Analysis of Integrated Networks of Tasks. SAINT is a network modeling and simulation technique for design and analysis of complex man-machine systems. SAINT provides the conceptual framework for representing systems that consist of discrete task elements, continuous state variables, and interactions between them. It facilitates an assessment of the contribution system components make to overall system performance. A preprocessor routine is available to create DIMENSION statements for the source program that are tailored to the size requirements of the problem being treated. Provisions are made for structuring model outputs for post-run processing, using available statistical analysis packages. Extensive error checking routines are also provided in SAINT.

## 1 INTRODUCTION

SAINT is a computer simulation tool for modeling and analyzing man-machine systems. While SAINT was designed for modeling manned systems in which human performance is a major concern, it is potentially applicable to a broad class of problems--those in which discrete and continuous elements are to be portrayed and the behavior exhibits time varying properties. SAINT provides a mechanism for describing these dynamics so analyses can be performed.

SAINT evolved from two separate technologies. Task analysis and the Monte Carlo simulation of operator performance under workload stress as represented by Siegel and Wolf (1967), were the origin for the human factors development. Many of the features eventually incorporated into SAINT were identified as requirements based upon experience in applying this technology. The second origin of SAINT was in the GASP family of simulation techniques (Pritsker, 1974). The earliest version of SAINT was an incorporation of the Siegel-Wolf model in a modified PERT package (Pritsker, et al., 1974). The subsequent evolution of SAINT adapted features of GASP IV and allowed SAINT to become a flexible, sophisticated, combined modeling technique where networks of discrete events could be modeled along with the dynamics of continuous processes.

It is this ability to combine models of dynamics (e.g., aircraft equations of motion) with models of discrete activity sequences (e.g., operator actions) that permit the systems analyst to describe both hardware and human performance in the context of a single model. This affords the system engineer the opportunity to analyze system effectiveness and quantify the relative contributions of man and machine.

## 2 SAINT CONCEPTS

For the discrete event simulation, a graphical-network approach to modeling is taken, whereby a user of SAINT describes the system to be analyzed via a network model and auxiliary descriptions (e.g., equipment and operator performance parameters). (A symbol set has been devised for diagramming the discrete task network.) The SAINT computer simulation program accepts a description of the

network to be simulated and automatically performs an analysis to obtain statistical estimates of system performance. For the continuous process representation, the user is expected to provide FORTRAN statements of the relevant state equations to be solved. Mechanisms are provided for creating an interaction between the discrete and continuous components of the model.

### 2.1 Discrete Task-Oriented Model Component

The discrete task-oriented component of the SAINT model consists of nodes and branches, each node representing a task. Tasks are described by a set of characteristics (e.g., performance time duration, priority, resource requirements). Branches connecting the nodes indicate precedence relations and are used to model the sequencing and looping requirements among the tasks. Complex precedence relations have been designed into SAINT to allow predecessor-successor relationships which are deterministic, probabilistic, and conditional. Resources, either human operators or hardware equipment, perform the tasks in accordance with the prescribed precedence relations, subject to resource availability. The precedence relations also indicate the flow of information through the network. Information is organized into packets with each information packet containing attributes that characterize the information being processed. The information packet can characterize items flowing through the network, signals being processed by the network, or any other concept related to network flow. When a task is completed, the information packet residing at the task is transmitted along each precedence branch selected. Information attribute values can be assigned or modified at any task in the network and can influence both task performance times and task branching relations.

Resources perform tasks either individually or in groups. Each resource included in a SAINT model is described by a set of attributes. These attributes are also organized into packets, with each packet characterizing a particular resource. Examples of operator attributes include such things as level of training, age, height, etc. Machine reliability is an example of an equipment attribute. Resource attributes are used in conjunction with the task descriptions in order to make a general network model resource-specific. The initial values of these

user-defined resource attributes are assigned prior to the start of the simulation. The values may be dynamically changed at any task in the network and can be used as parameters in determining both task performance times and precedence relations.

In many instances it may be desirable to specify attributes which are not directly applicable to an information-oriented or resource-oriented characterization. These attributes are global in nature and do not flow through or move about the network as information and resource packets do. Temperature and time remaining in a mission are examples of model parameters which may be characterized as system attributes. Just as with information and resource attributes, system attributes may influence the task network performance and flow.

Each task in a SAINT network has two requirements which must be satisfied before the task can be performed. First, a specified number of predecessor tasks must be completed before the task is released. Second, once the task has been released, the resources required to perform the task must be available, (that is, not be busy performing other tasks). All tasks which have been released (all predecessor requirements have been satisfied) but whose required resources are not available are ranked in a queue according to their priority. Task priority may be assigned at the start of the simulation and may change dynamically as a function of system parameters and contingencies. When the required resources become available with task completions, the tasks in the waiting queue are started. The time to perform a task may be specified as a random variable defined by a probability distribution. SAINT supplies the user with 11 different distributions (Normal, Gamma, Beta, Weibull, etc.; see Hastings and Peacock, 1975).

Frequently the task performance time is also a function of the type of task, the resource or resources performing the task, the status of the system, or the condition of the environment at the time the task is executed. SAINT provides for the specification of factors which influence task performance via user-written moderator functions. It is presumed the modeler can describe (e.g., by least squares techniques) the functional relationships between a set of conditions and a performance parameter or attribute of interest. For example, one might hypothesize that fatigue affects operator performance such that the average task time increases as a function of mission duration. Research data must be obtained to postulate the functional form of this relationship and fit a curve to these results. This empirically derived relationship can then be implemented in SAINT as a moderator function to determine the possible impact fatigue could have on operator performance. In addition to moderator functions, user-written functions can be developed for specifying attribute assignments. Both types of functions are written in FORTRAN or a FORTRAN-compatible language.

Contingencies, decision making, and emergency conditions can be represented via SAINT's flexible attribute assignment and branching logic. SAINT provides two additional mechanisms for modeling system performance.

The first of these is termed task modification. This feature enables the user to modify task parameters as a function of ongoing system events. For example, consider a task which may require repetition due to a possibility of failure on the first attempt. The second time the task is performed the performance time may be significantly smaller than

the initial execution. SAINT provides for the modification of the task time distribution after the initial attempt. Other task parameters can be modified in a similar fashion.

The second SAINT modeling construct of interest is "clearing." Both tasks and resources can be cleared. "Task" clearing halts a specified task in progress, contingent on the completion of another task. "Resource" clearing halts whatever task the specified resource is performing. Both types of clearing may specify an additional task to be signaled. As an example, consider the simulation of an emergency condition in which all operators must stop their ongoing activities to assist in the emergency operations. This situation is best modeled in SAINT with resource clearing. The onset of the unexpected event would "free-up" (clear) the operators. Concurrently, emergency handling tasks would be signaled for initiation (and release if all other precedents were satisfied). Task and resource clearing provide dynamic realism in man-machine simulation modeling.

For those wishing to diagram the network, the symbol used to model a task in SAINT is illustrated in Figure 1. The input side of the node reflects the

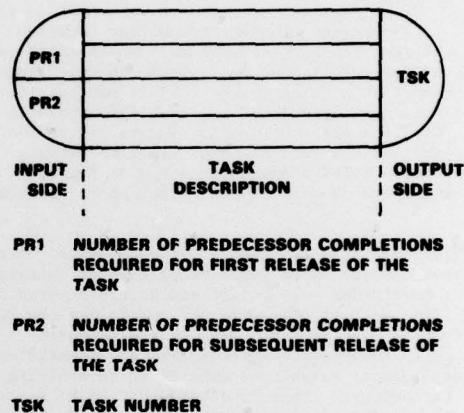


Figure 1 Task Symbol

precedence requirements for releasing a task. The number of requirements for releasing a task the first time is on the top (PR1), and the number of requirements for releasing a task on subsequent times is on the bottom (PR2).

The center portion of the task symbol contains all task description information, such as performance time characteristics, statistics to be collected, and attributes to be assigned. It is subdivided into rows, with each row containing a specific type of descriptive information about the task. Further, each row is divided into two parts. The left-hand part contains the task description code. It is used to identify the type of information that appears in the right-hand part of the row, and can be any of the 17 available codes shown in Table I.

The LABEL permits an eight character identifier to be associated with this node to depict the nature of the task/activity represented. The TIME characteristics, when specified, indicate the distribution type and parameter values for the characterization of task duration. If activity times are known to be a function of specifiable factors (e.g., task, system, or information attributes), a moderator function (MODF) may be employed (as a FORTRAN subroutine)

TABLE I  
TASK DESCRIPTION CODES

LABL	task label
TIME	performance time characteristics
MODF	moderator functions
DMOD	distribution modification
RESR	resource requirements
PRTY	priority
INCM	information choice mode
DIFF	different predecessor option
PREC	completion precedence
UTCH	user-defined task characteristics
ATAS	attribute assignments
STAT	statistics
MARK	mark information
TCLR	task clearing
RCLR	resource clearing
SWIT	switching
REGL	regulation

to generate the activity duration instead of generating a time value by Monte Carlo methods. If Monte Carlo methods are employed (via TIME specification), a modification can be effected during model execution by using the DMOD feature to identify an alternate distribution and/or parameter set when specified event conditions prevail. RESR may be used to specify the type and quantity of resources and whether multiple resources imply substitution ("or") rather than conjoint ("and") requirements. If priority (PRTY) is a concern, it can be specified a priori and subsequently manipulated dynamically during model execution. Since information packets can arrive at a task from several sources, but only one will exit, it is necessary to specify which incoming packet will be passed along, INCM. The default condition for processing information packets is to simply pass the last one arriving at the node. If different predecessor completions are required in order for the task to be released, the DIFF option must be specified. Otherwise the multiple occurrence of any predecessor may cause the task to be prematurely released. When two or more tasks have identical completion times, it is necessary to specify which will take precedence (PREC) over the others. User-defined task characteristics (UTCH) permit the user to specify additional attributes of a task (e.g., difficulty, complexity, etc.), and these attributes can be modified upon task execution. Information, resource, and system attributes can be assigned or updated (ATAS) upon task release, start, or completion as required. The statistics to be collected (STAT) are described in subsequent discussion. A particular task can be used to mark the start point (MARK) for timing how long it takes to traverse a path to some other task of interest. The MARK feature allows elapsed time computations within the network (e.g., time between events). Task and resource clearing operations are established by specifying the appropriate parameters associated with the TCLR and RCLR mnemonics. If a discrete task actually affects some binary state or there is a need to set a flag for some subsequent processing option, SWIT allows a switch to be set (or reset) that can subsequently be examined. The REGL mnemonic is used as a device for regulating values employed in the continuous process model, where a task is permitted to alter a state variable, for example.

By selectively using these description codes, only the information necessary to describe a task need be shown on the task symbol. In this manner, any or all of the task description codes can be specified for a particular task. If more than the four rows provided are required for a complete description, the user simply adds the necessary number of

additional rows to the bottom of the task description portion of the task symbol.

The output side of the node contains the task number (TSK). In addition, the shape of the output side indicates the branching operation to be performed upon task completion. It specifies the process to be employed in selecting the successor tasks whose precedence requirements should be reduced by one. The four branching types included in SAINT are deterministic, probabilistic, conditional take-first, and conditional take-all. Their shapes are depicted in Figure 2.

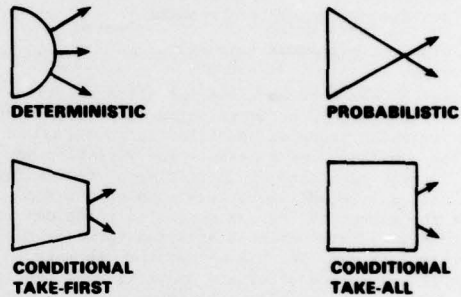


Figure 2 Task Branching Symbolism

When deterministic branching is specified, the number of requirements for all successor tasks is reduced by one. For probabilistic branching, each branch emanating from the task has an associated probability of selection. These probabilities may be specified directly or obtained from information, operator, or system attributes. Only a single successor task is selected. For conditional take-first branching, each branch has an associated condition, and the branches are ordered. Each condition is tested in the prescribed order, and the first branch whose condition is satisfied is selected. Conditional take-all branching operates in the same manner, but selects all branches whose conditions are satisfied. Conditions may be based on task completions, simulated time, or attribute values.

The above discussion only included the basic task node symbology. Additional symbolism is available for task modification, task signaling as a result of task or resource clearing, task signaling resulting from a threshold crossing, and state variable monitors (Wortman et al. 1977(a) and (b)).

## 2.2 Continuous State Variable Model Component

The second component of a SAINT model is the state variable description. The SAINT user defines these state variables by writing the algebraic, difference, or differential equations that govern their time-dependent behavior. The use of state variables in SAINT is optional.

The SAINT user writes the state variable equations in a FORTRAN subroutine (subroutine STATE). State variables represented by algebraic or difference equations are defined in subroutine STATE as SS(.) variables. Those represented by differential equations are written in terms of DD(.) variables. SAINT employs a Runge-Kutta-England (RKE) numerical algorithm to integrate the equations of subroutine STATE written in terms of the DD(.) variables. The RKE algorithm obtains a solution to a set of simultaneous first order ordinary differential equations. Higher order differential equations can be modeled by placing the equations in canonical form. Subroutine STATE can be used to model state variables using a combination of DD(.) and SS(.) variables.

In SAINT, simulated time is advanced in accordance with the type of system being modeled. If no state variables are included, simulated time is advanced from one task completion to the next. When state variables are included in the model, time is also incremented in steps between scheduled task completions for the purpose of updating the values of the state variables. The step size is a function of user-specified accuracy requirements.

### 2.3 Discrete and Continuous Component Interactions

The interactions between tasks and state variables are initiated either by tasks being completed or by state variables crossing specified threshold values. Upon the completion of a task, state variables may be discretely regulated by increasing or decreasing their values. In addition, task completions can change the values of logical variables which can be used to alter state variable equation forms or the network structure. In this manner the discrete task-oriented component of the model affects the continuous state variable component.

Threshold crossings by state variables can signal or initiate tasks. Thus the values of state variables can influence task performance characteristics and precedence relations. Threshold crossings can also change the values of logical variables which, in turn, can be used to alter equation forms or change task precedence.

As an example of discrete and continuous component interactions, consider a system in which a pilot must keep the aircraft altitude within specified constraints. The pilot's inputs might be modeled as discrete tasks and the aircraft dynamics as continuous state variables. When the altitude state variable crosses the allowable threshold value, the corresponding discrete pilot correction task is signaled. The pilot makes the appropriate input and regulates the state variable(s) which determine altitude. Thus, through this component interaction, the aircraft altitude is brought back within acceptable limits.

### 3 STATISTICAL OUTPUT

Once the model has been built, the modeler can impose a data collection structure to obtain information about his description of the system as it is exercised. A variety of data can be obtained; these fall into three major categories. The first type of output is a statistical description of the execution of specific nodes or collections of nodes. Table II provides some insight into the variety of possible measures one can create using the built-in features of SAINT. Since users can create their own functions for updating attributes and for moderating network parameters, it has been necessary to allow the user to collect his own statistics on those parts of the model which cannot be predefined

because the user creates them himself. SAINT supplies statistical subroutines for collecting data on user-supplied parts of the model. Tabular summaries of the computed descriptive statistics can then be generated to portray the results of a single iteration, a set of iterations, or a series of iterated runs showing the trends induced by some systematic variation of run conditions.

TABLE II  
STATISTICS CODES

What	When			
	Release	Start	Completion	Clearing
First	FIR REL	FIR STA	FIR COM	FIR CLE
All	ALL REL	ALL STA	ALL COM	ALL CLE
Between	BET REL	BET STA	BET COM	BET CLE
Number	NUM REL	NUM STA	NUM COM	NUM CLE

The second type of output is a graphic portrayal of the probability and cumulative density functions for a distributed variable. They provide a quick look at the shape of the data. An experienced user can store the actual values on an external device; later, the data can be fed to a plotting package for reproducible drawings.

The third type of output are time traces of the state variables. Up to 10 variables can be plotted on the same graph with user specified scale factors and plotting symbols. Multiple graphs can be generated. Tabled values of the variables can also be obtained. Here again, an experienced programmer can save these data for more precise plots. The tabulated plot provided by SAINT equips the user to quickly examine the results of his simulation run.

### 4 THE SAINT PROGRAM

Development of the SAINT simulation package has been completed and is fully documented (Wortman, et al., 1977a, b and c and Duket, et al., 1977). SAINT was developed in ANSI standard FORTRAN and consequently, is machine-independent. The user, however, must supply his own system-specific random number generator. The task network data is punched on cards in free-form. SAINT includes an extensive input error-check feature to assist users in debugging their models. For production runs, users can select a more efficient non-error-checking version of SAINT. A separate FORTRAN program has been devised to create a source module with the COMMON blocks sized to the problem being run. SAINT also includes provisions that allow formatting model outputs so they can be processed by available statistical analysis packages (Wortman, et al., 1977c).

### 5 APPLICATIONS OF SAINT

SAINT has been used to analyze a wide variety of man-machine systems. It is gaining a wide and enthusiastic acceptance by systems modelers and analysts of many disciplines. The following is a list of completed or ongoing modeling and simulation efforts involving the use of SAINT: SAINT has been used by the Aerospace Medical Research Laboratory (AMRL) to evaluate alternatives for a Vehicle/Drone Control Facility (RPV/DCF) in which operators monitor and control the flight of RPV's through the use of visual (CRT) displays (Wortman,

et al., 1976). SAINT was used, also, to provide flight control performance predictions for the Digital Avionics Information System (DAIS) cockpit configuration in which dedicated instruments, displays, and subsystem status displays have been replaced with interactive multipurpose displays and multifunction keyboard switching (Kuperman and Seifert, 1975). SAINT is currently being used by AMRL to provide cost trade off design analyses of three proposed configurations for the UPD-X All Weather Wide Area Surveillance ground exploitation station. AMRL plans to utilize SAINT to analyze design proposals in a B-52 strategic navigation system involving complex crew activities and task management (Chubb and Berisford, 1977). SAINT has been used by the Air Force Human Resources Laboratory to explore the feasibility of employing computer simulation for evaluating human effects on nuclear systems safety in a missile loading operation (Askren, et al., 1976). SAINT was used by Air Force Weapons Laboratory to examine workload sharing and nuclear radiation effects on pilot performance in an air-to-air refueling mission (Wortman, et al., 1975). SAINT has been used by Purdue University researchers to investigate the effect of higher degrees of automation, different capacities of process limiting operations, and alternative task allocations on the operator's idle times in a hot strip mill (Maltas and Buck, 1975). SAINT has been used by the U.S. Department of Commerce office of Telecommunications to analyze communication frequency utilization in a railroad switching yard. SAINT has been used by New Mexico State to compare theoretical human performance predictions with empirically derived performance data (Teichner, 1976). SAINT is being employed by Pritsker and Associates in support of the Army Research Institute to analyze human system performance in an AN/TSQ-73 guided missile air defense system operation. Since this work is in progress at this time, it has yet to be documented. SAINT is also being utilized by several universities both in the classroom and for research activities. Among these are Purdue, Iowa State, and North Carolina State.

## 6 REFERENCES

- Anonymous (1978). UPD-X Exploitation Computer Models. System Development Corporation, Dayton, Ohio, TM-HU-505/000/01.
- Askren, William B., Campbell, Wendy Bohnett, Seifert, Deborah J., Hall, Thomas T., Johnson, Robert C., Sulzen, Robert H. (1976). Feasibility of a Computer Simulation Method for Evaluating Human Effects on Nuclear Systems Safety, Air Force Human Resources Laboratory, Air Force Systems Command, Brooks Air Force Base, AFHRL-TR-76-18.
- Chubb, Gerald P. and Berisford, Kathleen M. (1977). Manned System Modeling: SAINT Applied to Strategic Navigation, Record of Proceedings, Tenth Annual Simulation Symposium, pp. 321-352.
- Duket, S.D., Wortman, D.B., Seifert, D.J., Hann, R.L., Chubb, G.P. (1977). Documentation for the SAINT Simulation Program, Aerospace Medical Research Laboratory, Wright-Patterson Air Force Base, Ohio, AMRL-TR-77-63.
- Hastings, N.A.J. and Peacock, J.B. (1975). Statistical Distributions, Australia, Butterworths.
- Kuperman, G.G. and Seifert, D.J. (1975). Development of a Computer Simulation Model for Evaluating DATS Concepts. Proceedings: Human Factors Society 19th Annual Meeting, Dallas, Texas, Oct, pp. 347-353.
- Maltas, Kenneth L., Buck, James R. (1975). Simulation of a Large Man/Machine Process Control System in the Steel Industry. Proceedings: Human Factors Society 19th Annual Meeting, Dallas, Texas, pp. 193-205.
- Pritsker, A.A.B. (1974). The GASP IV Simulation Language. New York, John Wiley & Sons, Inc.
- Pritsker, A.A.B., Wortman, D.B., Seum, C.S., Chubb, G.P., and Seifert, D.J. (1974). SAINT: Volume I, Systems Analysis of Integrated Networks of Tasks. Aerospace Medical Research Laboratory, Wright-Patterson Air Force Base, Ohio, AMRL-TR-78-126.
- Siegel, Arthur I., and Wolf, J. Jay (1967). Man-Machine Simulation Models: Performance and Psychosocial Interaction. New York, John Wiley & Sons, Inc.
- Teichner, Warren H. (1976). Human Performance Simulation, Annual Report. AFOSR Contract No. F44620-76-C-0013, Bolling Air Force Base, Washington, D.C.
- Wortman, David B., Sigal, C. Elliot, Pritsker, A. Alan B., and Seifert, Deborah J. (1975). New SAINT Concepts and the SAINT II Simulation Program, Section VI, Aerospace Medical Research Laboratory, Wright-Patterson Air Force Base, Ohio, AMRL-TR-74-119.
- Wortman, D.B., Duket, S.D., and Seifert, D.J. (1976). SAINT Simulation of a Remotely Piloted Vehicle/Drone Control Facility: Model Development and Analysis. Aerospace Medical Research Laboratory, Wright-Patterson Air Force Base, Ohio, AMRL-TR-75-118.
- Wortman, D.B., Duket, S.D., Seifert, D.J., Hann, R.L., Chubb, G.P. (1977a). Simulation Using SAINT: A User-Oriented Instruction Manual. Aerospace Medical Research Laboratory, Wright-Patterson Air Force Base, Ohio, AMRL-TR-77-61.
- Wortman, D.B., Duket, S.D., Seifert, D.J., Hann, R.L., Chubb, G.P. (1977b). The SAINT User's Manual. Aerospace Medical Research Laboratory, Wright-Patterson Air Force Base, Ohio, AMRL-TR-77-62.
- Wortman, D.B., Duket, S.D., Seifert, D.J., Hann, R.L., Chubb, G.P. (1977c). An Example to Illustrate the Use of SPSS for the Analysis of a SAINT Model. Aerospace Medical Research Laboratory Wright-Patterson Air Force Base, Ohio, AMRL-TR-77-64.