MODELING AND ANALYSIS USING SAINT: A COMBINED DISCRETE/CONTINUOUS NETWORK SIMULATION LANGUAGE

David B. Wortman and Steven D. Duket

Pritsker & Associates, Inc.
West Lafayette, Indiana 47906

Deborah J. Seifert

Aerospace Medical Research Laboratory
Wright-Patterson AFB, Ohio 45433

ABSTRACT

SAINT, Systems Analysis of Integrated Networks of Tasks, is a network modeling and simulation technique developed to assist in the design and analysis of complex man-machine systems.* SAINT provides the concepts necessary to model systems that consist of tasks (discrete elements), state variables (continuous elements), and interactions between them. It facilitates the assessment of the contribution that system components make to overall system performance.

INTRODUCTION

SAINT is a unique and powerful technique that is both a systems modeling vehicle and a computer analysis tool. It is the only available technique that allows engineers and human factors specialists to develop system models in which men, machines, and environmental conditions are represented as elements of a network. SAINT models permit the analyst to investigate the impact of component characteristics on overall system performance without a major investment in equipment and time and without necessitating a commitment to prototype hardware development.

SAINT has been designed, developed, and used for modeling and analyzing systems in which resources (men and machines) perform tasks subject to physical and environmental constraints. It satisfies the need for a graphical approach to the modeling and analysis of systems which contain procedural, risk and random elements. For engineers and human factors specialists, SAINT provides modeling capabilities similar to those provided by circuit diagrams for electrical engineers, signal flow graphs and block diagrams for systems analysts, and PERT/CPM networks for project managers. Further, it provides automatic model analysis capabilities via the SAINT simulation program.

*This research has been supported by the United States Air Force under contract no. F33615-76-C-5012.

APPLICATIONS OF SAINT

The application of SAINT is extending into many diverse areas. It is gaining a wide and enthusiastic acceptance by systems modelers and analysts of many disciplines. For example, Pritsker & Associates, under contract to the Aerospace Medical Research Laboratory, Human Engineering Division, has effectively used SAINT to model and analyze a Remotely Piloted Vehicle/Drone Control Facility (2,9,10,11). In this system, pilotless aircraft are directed to a target area by operators stationed at remote terminals in an underground facility. Air Force personnel were interested in the feasibility of this system, the required flight characteristics of the remotely piloted vehicles, the effect of human characteristics on system performance, and the effect of environmental and threat factors on mission performance.

In a non-military application, SAINT was employed by Pritsker & Associates, under contract to the U.S. Department of Commerce, Office of Telecommunications, to model and analyze communication frequency utilization in a railroad switching yard. In that model, tasks represented physical operations and communications performed during the switching process. The available communication frequencies were modeled as resources. Frequency utilization was the prime measure of system performance. The model was used to evaluate the effect of changing communication frequency assignments in the railroad industry on yard operations.

The following is a list of other completed or ongoing modeling and analysis efforts involving the use of SAINT:

- SAINT has been used by the Human Resources Laboratory at Wright-Patterson Air Force Base to determine the feasibility of integrating human resources data and maintenance task data with a computer simulation technique to form a computer-based tool for performing safety analyses of nuclear systems (1).

- SAINT has been used by the Human Engineering Division of the Aerospace Medical Research Laboratory at Wright-Patterson Air Force Base to model and further investigate psychological theory (5).
- SAINT has been used by the Human Engineering Division of the Aerospace Medical Research Laboratory at Wright-Patterson Air Force Base to analyze multi-function switching and multi-purpose display concepts in support of the Digital Avionics Information System (DAIS) advanced development program (6).
- 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 operator idle time in the steel industry (7).
- SAINT has been used by the Air Force Weapons Laboratory at Kirtland Air Force Base to model in-flight aircraft refueling operations (12).
- SAINT has been used by the Marshall Space Flight Center to study the scheduling of experiments for the Spacelab.
- SAINT has been used by the School of Aerospace Medicine at Brooks Air Force Base to determine crew survivability/vulnerability in a nuclear environment.
- SAINT is currently being used by the Human Engineering Division of the Aerospace Medical Research Laboratory to study navigation and electronics warfare officer performance in B-52 missions.

NETWORK MODELING AND ANALYSIS

The SAINT philosophy is to separate the modeling process from the analysis process. A graphical approach to modeling is taken in which the system to be analyzed is represented by a network model. The network model facilitates communication regarding the characteristics of the system and also serves as the basis for subsequent system analysis. The SAINT approach to network modeling and analysis is depicted in Figure 1.

A SAINT network model is developed using symbols contained in the SAINT symbol set. The fundamental elements of SAINT networks are tasks, resources (personnel and/or equipment) required to perform tasks, relationships among tasks, and system status variables referred to as state variables. System performance is related to which tasks are performed, the manner in which they are initiated, utilization of the system resources, and the extent to which certain states of the system are achieved

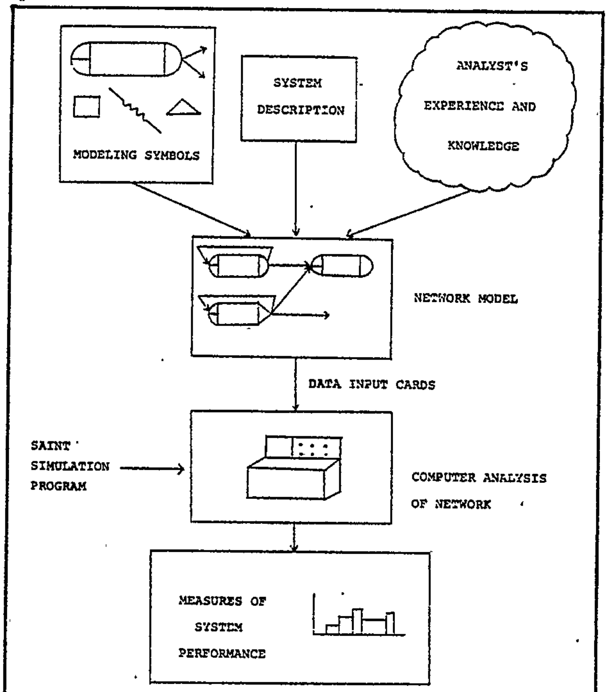

Figure 1

or maintained. The SAINT symbol set (13,14) provides the tools required to build models of systems in which resources perform tasks to accomplish system objectives.

In addition to providing a flexible set of symbols which are integrated to form a network model of the system under study, the SAINT modeling approach allows for the specification of the conditions and constraints under which the system operates. These conditions and constraints may include such factors as time constraints on resources and the environment within which the tasks must be performed. By providing the means for specifying such conditions and constraints, SAINT allows the analyst to depict system performance in a variety of situations.

An important factor in the SAINT modeling approach is the analyst's experience and knowledge of the system. It is the analyst's responsibility to integrate SAINT symbols to form a network model of the system; by identifying each task to be performed, the resources required to perform them, the required task sequences and priorities, the state variable relationships, the interactions between the model components, and the environmental conditions under which the system operates. Thus, while SAINT provides the necessary tools for developing system models, it is the analyst who builds the models by utilizing the tools.

Once the analyst has developed a SAINT model, the SAINT simulation program will automatically generate system performance estimates, i.e., it will perform an analysis of the network model. The SAINT simulation program contains all of the instructions necessary to interpret any combination of SAINT symbols. The input to the SAINT simulation program is an alphanumeric representation of the network model. It processes the input and performs a simulation of the network model to obtain estimates of system performance.

The level of detail at which a system or system segment should be modeled cannot be specified a priori. It is the analyst's responsibility to determine the level of detail to be included in the network model based upon the nature of the problem he is trying to solve and an analysis of the task components and their interrelationships. He must decide if it is sufficient to model a task as a single unit, or if it is necessary to model each component individually. Detailed network models of specific tasks may be included in an overall system model. Further, the output derived from simulating a detailed model may be used as the description of a particular task in an overall system model. The concept of hierarchial modeling, where the output of one simulation is used as input to another, is an inherent aspect of the SAINT modeling and analysis approach.

SAINT MODELING CONCEPTS

A SAINT model contains two basic components. The discrete component of the model is a network consisting of nodes, branches, resources, and information. It is so named because the SAINT simulation program uses discrete event simulation procedures to analyze this model component. The continuous component is comprised of the mathematical descriptions of system variables whose values change continuously over time. The SAINT simulation program uses continuous simulation procedures to analyze this model component. In addition, SAINT provides concepts for modeling and analyzing interactions between the discrete and continuous components of the model.

Discrete Model Component

The discrete component of a SAINT model is prepared in network form. The network consists of nodes and branches, where each node represents a task. Tasks are described in SAINT by a set of characteristics (e.g., performance time, priority, statistics to be collected, resource requirements). Additional characteristics can be defined by the user to satisfy any special requirements. Branches connecting the nodes indicate precedence relations and are used to model the sequencing requirements among the tasks. Complex precedence relations have been designed into SAINT to allow predecessor-successor relationships which are deterministic, probabilistic, and conditional. Resources perform the tasks in accordance with the prescribed precedence relations, subject to resource availability. Both task characteristics and the basic network flow defined by the precedence relations may be altered as a function of system status.

Tasks can have special roles within the network. Tasks can be at the beginning of the network, i.e., source tasks. Source tasks are automatically released at the start of the simulation. Tasks can also be at the end of the network, i.e., sink tasks. Sink tasks signify the possible end points for the network. When a prescribed number of sink task completions have occurred, the simulation is terminated. In addition, it may be desirable to collect statistics at intermediate points within the network. Such tasks are referred to as statistics tasks. SAINT provides numerous alternatives in the specification of the type of statistics to be collected at these tasks. It is important to note that the characterization of a task as a source, sink, or statistics task is the means by which the modeler superimposes an information system on the physical processes being modeled. In using SAINT, it must be remembered that a model of a system is being developed as well as the means for obtaining performance statistics about the operation of the model. Although this can add some complexity to the modeling process, it forces the modeler to consider the objective of his effort at the same time he is developing the model.

Each task in a SAINT network has two requirements which must be satisfied before it 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. If the resources are available, the task is started. If the resources are not available, i.e., they are busy performing other tasks, the task cannot be started. All tasks which have been released (all predecessor requirements have been satisfied) but whose required resources are not available are ranked in a queue of tasks waiting to be performed according to a user-defined priority scheme. When the required resources are made available due to task completions, tasks in the waiting queue may be started if their resource requirements can be satisfied using the newly-available resources.

The baseline time required to perform a task can be a constant or a sample drawn from any one of eleven probability distributions available in SAINT. It may also be a function of the type of task, the resources performing the task, the status of the system at the time the task is started, or the current condition of the environment. In SAINT, any function that specifies an effect on baseline task performance is defined as a moderator function. Moderator functions are programmed by the user in a special SAINT subprogram.

In addition to identifying the predecessor-successor relationships among tasks, precedence relations indicate the flow of information through the network. Information is organized into packets, with each information packet containing user-defined attributes that characterize the information being processed. When a task is completed, the information packet residing at the task is transmitted along each branch selected according to the predecessor-successor relationship specified by the user.

Resources perform tasks either individually or in groups. Statistics on resource utilization are automatically provided by the SAINT simulation program. Each resource included in the SAINT model is described by a set of attributes. These attributes are also organized into packets with each packet characterizing a particular resource. Resource attributes are used in conjunction with the task descriptions in order to make a general network model resource-specific.

In many situations, it may be desirable to specify attributes which are not directly applicable to an information-oriented or resource-oriented characterization. These attributes, being global in nature, do not flow through or move about the network as information and resource packets do. These attributes are characterized as system attributes, with one set of system attributes being associated with a SAINT model.

Information, resource, and system attribute values may be assigned or changed at any task in the network. They can be constants or samples drawn from any of the eleven probability distributions available in SAINT. Further, attribute values may be a function of system status or environmental conditions. If so, the SAINT user can program the necessary relations in a special SAINT subprogram.

In developing user-written subprograms to be included in a SAINT model, local variables and special relationships may be defined. SAINT includes a number of user-callable subprograms that facilitate the manipulation of input and output information relating to these variables and relationships. Subprograms are available to read user-defined input data, generate statistical summaries based on observations, generate statistical summaries of time-persistent variables, prepare histograms, and develop time-dependent plots. Subprograms for initializing statistical information storage arrays and for generating user-required output reports are also provided.

Continuous Model Component

The second component of a SAINT model consists of the descriptions of state variables. While variables describing the discrete model component, such as resource busy/idle status, change values at discrete points in time, state variables change values continuously over time. The SAINT user defines these state variables by writing the algebraic, differential, or difference equations that govern their time-dependent behavior. The incorporation of a continuous component in a SAINT model is optional. The capability for modeling and analyzing state variables is similar to that provided by GASP IV (8).

The SAINT user is required to write the governing state variable equations in a special user-written FORTRAN subprogram. State variables represented by algebraic or difference equations are defined as SS(.) variables. Those represented by differential equations are written in terms of DD(.) variables. SAINT uses a variable step size Runge-Kutta-England (RKE) algorithm to integrate the DD(.) variable equations. The RKE algorithm is used to obtain a solution to a set of simultaneous first order ordinary differential equations. Higher order differential equations can be modeled by putting the equations in canonical form. Any combination of DD(.) and SS(.) variables may be modeled in SAINT.

Information concerning the time-dependent performance of state variables may be output in three forms. The output may consist of plots of the values of state variables versus time, tables of the values of state variables versus time, and time-integrated statistical summaries of state variable values. The user may specify any combination of the three types of output for each state variable included in the model.

Discrete and Continuous Component Interactions

SAINT provides the capability of discrete event simulation with the incorporation of the discrete model component. With the provision of methods for modeling state variables, SAINT offers those capabilities most often associated with analog or continuous simulation. In addition to these, SAINT provides a number of concepts for modeling and analyzing interactions between the discrete and continuous model components.

The interactions between tasks and state variables are initiated either by tasks being completed or by state variables crossing specified threshold values. Upon completion of a task, state variables may be discretely regulated by increasing or decreasing their values. In addition, task completions can cause the values of logical variables to be changed. State variable equation forms or the structure of the task network can be altered as a function of the values of these logical variables. In this manner, the discrete component of the model affects the continuous component.

Threshold crossings by state variables can cause task predecessor requirements to be reduced. In this way, the values of state variables can affect the initiation of task performance and/or the precedence relations between tasks. Threshold crossings can also cause the values of logical variables to be changed. State variable equation forms or task precedence relations can be altered as a function of logical variable values.

THE SAINT SIMULATION PROGRAM

The SAINT simulation program is designed to simulate SAINT models consisting of any combination of SAINT symbols. Data cards that describe the model serve as input to the program. The data cards are prepared in a special free format, with input fields separated by commas. An extensive input error-checking feature is included to assist users in debugging their models. For production runs, a more efficient non-error-checking option may be selected.

The SAINT simulation program is written in ANSI Standard FORTRAN IV and can be run on any machine that has the necessary compiler. The only machine-specific subprogram required is a pseudo-random number generator.

The modular design of the SAINT simulation program allows efficient use of core storage by providing the means to develop a storage-saving overlay structure. In its unoverlayed form, the program requires approximately 55,000 decimal words of core storage. This figure can be altered significantly by increasing or decreasing program dimensions to satisfy model requirements.

DOCUMENTATION, AVAILABILITY, AND SUPPORT

During the four year SAINT development process, numerous technical documents were prepared. However, all SAINT documentation has now been incorporated in four technical reports:

> The symbolism and terminology required for modeling systems using SAINT are presented in Simulation Using SAINT: A User-Oriented Instruction Manual (13).
>
> The procedures for using the SAINT simulation program to analyze SAINT models are described in The SAINT User's Manual (14).
>
> The overall structure and individual FORTRAN subprograms of the SAINT simulation program are described in Documentation for the SAINT Simulation Program (3).
>
> The use of an external statistical analysis package to analyze SAINT output is described in Analyzing SAINT Output Using SPSS (4).

Information on how to obtain SAINT documentation, the SAINT simulation program, or support in SAINT modeling efforts is available from the authors.

SUMMARY

Decomposing an overall system into its components and then providing a vehicle to integrate the components into system performance measures is in the true spirit of the system approach to problem-solving. SAINT provides this capability. It is continuing to prove to be a useful, powerful, and flexible tool for modeling and analyzing complex systems.

ACKNOWLEDGEMENT

In developing the SAINT technique, the authors received guidance and assistance from many individuals, especially the staff members of Pritsker & Associates and the Aerospace Medical Research Laboratory. We appreciate their efforts and acknowledge their contributions.

BIBLIOGRAPHY

1. Askren, W.B., W.B. Campbell, D.J. Seifert, T.H. Hall, R.C. Johnson, and R.H. Sulzen, Feasibility of a Computer Simulation Method for Evaluating Human Effects on Nuclear Systems Safety, AFWL-TR-76-15, Air Force Weapons Laboratory, Kirtland Air Force Base, New Mexico, May 1976.

2. Duket, S.D., D.B. Wortman, and D.J. Seifert, SAINT Simulation of a Remotely Piloted Vehicle/Drone Control Facility: Technical Documentation, AMRL-TR-75-119, Aerospace Medical Research Laboratory, Wright-Patterson Air Force Base, Ohio.

3. Duket, S.D., D.B. Wortman, R.L. Hann, G.P. Chubb, and D.J. Seifert, Documentation for the SAINT Simulation Program, AMRL-TR-77-63, Aerospace Medical Research Laboratory, Wright-Patterson Air Force Base, Ohio.

4. Duket, S.D., D.B. Wortman, R.L. Hann, G.P. Chubb, and D.J. Seifert, Analyzing SAINT Output Using SPSS, AMRL-TR-77-64, Aerospace Medical Research Laboratory, Wright-Patterson Air Force Base, Ohio.

5. Hann, R.L., and G.G. Kuperman, "SAINT Model of a Choice Reaction Time Paradign", Proceedings of the 19th Annual Meeting of the Human Factors Society, Dallas, October 1975.

6. Kuperman, G.G., and D.J. Seifert, "Development of Computer Simulation Model for Evaluating DAIS Display Concepts", Proceedings of the 19th Annual Meeting of the Human Factors Society, Dallas, October 1975.

7. Maltas, K.L., and J.R. Buck, "Simulation of a Large Man/Machine Process Control System in the Steel Industry", Proceedings of the 19th Annual Meeting of the Human Factors Society, Dallas, October 1975.

8. Pritsker, A.A.B., The GASP IV Simulation Language, John Wiley & Sons, New York, N.Y., 1974.

9. Wortman, D.B., S.D. Duket, and D.J. Seifert, SAINT Simulation of a Remotely Piloted Vehicle/Drone Control Facility: Model Development and Analysis, AMRL-TR-75-118, Aerospace Medical Research Laboratory, Wright-Patterson Air Force Base, Ohio.

10. Wortman, D.B., S.D. Duket, and D.J. Seifert, "Simulation of a Remotely Piloted Vehicle/Drone Control Facility Using SAINT", Proceedings of the 1975 Summer Computer Simulation Conference, San Francisco, July 1974.

11. Wortman, D.B., S.D. Duket, and D.J. Seifert, "SAINT Simulation of a Remotely Piloted Vehicle/Drone Control Facility", Proceedings of the 19th Annual Meeting of the Human Factors Society, Dallas, October 1975.

12. Wortman, D.B., C.E. Sigal, A.A.B. Pritsker, and D.J. Seifert, New SAINT Concepts and the SAINT II Simulation Program, AMRL-TR-74-119, Aerospace Medical Research Laboratory, Wright-Patterson Air Force Base, Ohio, April 1975.

13. Wortman, D.B., S.D. Duket, R.L. Hann, G.P. Chubb, and D.J. Seifert, Simulation Using SAINT: A User-Oriented Instruction Manual, AMRL-TR-77-61, Aerospace Medical Research Laboratory, Wright-Patterson Air Force Base, Ohio.

14. Wortman, D.B., S.D. Duket, R.L. Hann, G.P. Chubb, and D.J. Seifert, The SAINT User's Manual, AMRL-TR-77-62, Aerospace Medical Research Laboratory, Wright-Patterson Air Force Base, Ohio.