014524-10-T
SEL 137

# *Performability Modeling with Continuous Accomplishment Sets*

J. F. MEYER

July 1979

Prepared for
National Aeronautics and Space Administration
Langley Research Center
Hampton, Virginia 23365

G. E. Migneault, Technical Officer

NASA Grant NSG 1306

DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING
SYSTEMS ENGINEERING LABORATORY
THE UNIVERSITY OF MICHIGAN, ANN ARBOR

014524-10-T
SEL 137

# PERFORMABILITY MODELING WITH CONTINUOUS ACCOMPLISHMENT SETS

J. F. Meyer

Prepared For
National Aeronautics and Space Administration
Langley Research Center
Hampton, Virginia 23365
G. E. Migneault, NASA Technical Officer
NASA Grant NSG 1306

John F. Meyer, Principal Investigator
Department of Electrical and Computer Engineering
Systems Engineering Laboratory
The University of Michigan
Ann Arbor, Michigan 48109

July 1979

014524-10-T
SEL 137

# PERFORMABILITY MODELING WITH CONTINUOUS ACCOMPLISHMENT SETS

J. F. Meyer

Prepared For
National Aeronautics and Space Administration
Langley Research Center
Hampton, Virginia 23365
G. E. Migneault, NASA Technical Officer
NASA Grant NSG 1306

John F. Meyer, Principal Investigator
Department of Electrical and Computer Engineering
Systems Engineering Laboratory
The University of Michigan
Ann Arbor, Michigan 48109

July 1979

## TABLE OF CONTENTS

# I. INTRODUCTION

The intent of this report is to make explicit the general applicability of performability modeling concepts and techniques. Earlier reports and publications on the subject have emphasized applications where the performance variable takes the values in a discrete (and usually finite) accomplishment set A. Although we have alluded to the fact that A may also be continuous (see [1], for example), the definitions of basic concepts such as performability and capability have heretofore been formulated for the discrete case only.

In the discussion that follows, the continuous case is treated explicitly, where we find that certain issues must be dealt with more carefully to insure that a given base model $X_S$ and capability function $\gamma_S$ comprise a legitimate performability model $(X_S, \gamma_S)$.

The development of these ideas follows the format of [1] and much of the commentary there regarding motivation and justification has been retained in this report. The report thus serves as a self-contained description of the basic concepts and constructs of performability modeling. This includes discussion of a hierarchical modeling framework, whose purpose is to aid the solution process. On the other hand, solution techniques per se, are not discussed. The latter is a subject in itself (in both the continuous and discrete cases) and constitutes an important area of ongoing research. Traditionally, modeling of computer performance (see [1] - [3], for example) has stressed the need to represent the probabilistic nature of user demands (workload) and internal state behavior, under the assumption that the computer's structure is fixed, that is, there are no permanent changes in structure due to faults. On the other hand, modeling of computer reliability (beginning with the pioneering work of Bouricius, Carter, and Schneider [4]) has stressed representation of the probabilistic nature of structural changes caused by transient and permanent faults of the computer.

In the face of these traditional modeling distinctions, we consider an important class of computing systems wherein system performance is "degradable," that is, depending on the history of the computer's structure, internal state, and environment during some specified "utilization period" T , the system can exhibit one of several worthwhile levels of performance (as viewed by the user throughout T ). In this case we find that performance evaluations (of the fault-free system) will generally not suffice since structural changes, due to faults, may be the cause of degraded performance. By the same token, traditional views of reliability (probability of success, mean time to failure, etc.) no longer suffice since "success" can take on various meanings and, in particular, it need not be identified with "absence of system failure."

Modeling needs for (gracefully) degradable systems were first investigated by Borgerson and Frietas [5] in connection with their analysis of the PRIME system [6]. Although they recognized the need to formulate the probability of each possible level of performance, that is, the probability of k "crashes" during T for k = 0, 1, 2, ..., their evaluation effort dealt mainly with the question of reliability (the probability of no crashes during T ). Other studies employing Markov models have likewise emphasized the evaluation of reliability oriented measures (see [7] - [9], for example).

Some recent investigations, on the other hand, have dealt with measures aimed at quantifying performance as well as reliability. In particular, Beaudry [10] has introduced a number of computation related measures for degradable computing systems and has shown how to formulate these measures in terms of a transformed Markov model. In examining reconfiguration strategies for degradable systems, Troy [11] has distinguished levels of performance according to "workpower" and has formulated system effectiveness (referred to as "operational efficiency") as expected workpower. In another recent study, Losq [12] has investigated degradable systems in terms of degradable resources, where each resource is modeled by an irreducible, recurrent, finite-state Markov process.

In the discussion that follows, we describe a general modeling framework that permits the definition, formulation, and evaluation of a unified performance-reliability measure referred to as "performability." It is shown that performability relates directly to system effectiveness and is a proper generalization of both performance and reliability. A critical step in performability modeling is the introduction of the concept of a "capability function" which relates low-level system behavior to user-oriented levels of performance. A hierarchical modeling scheme is used to formulate the capability function, and capability is used, in turn, to evaluate performability.

## II. SYSTEM MODELS

A computing system, as it operates in its use environment, may be viewed at several levels. At a low level, there is a detailed view of how various components of the computer's hardware and software behave throughout the utilization period. At this level, there is also a detailed view of the behavior of the computer's "environment," where by this term we mean both man-made components (user input, peripheral subsystems, etc.) and natural components (radiation, weather, etc.) which can influence the computer's effectiveness. The computer, together with its environment, will be referred to as the "total system." A second view of the total system is the user's view of how the system behaves during utilization, that is, what the system accomplishes for the user during the utilization period. A third, even less detailed view, is the economic benefit derived from using the system, that is, the computing system's worth (as measured, say in dollars) when operated in its use environment.

To formalize these views, we postulate the existence of a probability space $(\Omega, E, P)$ that underlies the total system, where $\Omega$ is the sample space, $E$ is a set of events (measurable subsets of $\Omega$), and $P$: $E \rightarrow [0,1]$ is the probability measure (see [13], for example). This probability space represents all that needs to be known about the total system in order to describe the probabilistic nature of its behavior at the various levels described above. It thus provides a hypothetical basis for defining higher level (i.e., less detailed) models. In general, however, it will neither be possible nor desirable to completely specify $\Omega$, $E$ and $P$.

In the discussion that follows, let S denote the total system, where S is comprised of a computing system C and its environment E. At the most detailed level, the behavior of S is formally viewed as a stochastic process [14], [15]

$$X_S = \{X_t | t \in T\} \tag{1}$$

where

T = a set of real numbers (observation times) called the utilization period

and, for all $t \in T$, $X_t$ is a random variable

$$X_t : \Omega \to Q$$

defined on the underlying description space and taking values in the state space Q of the total system. Depending on the application, the utilization period T may be discrete (countable) or continuous and, in cases where one is interested in the long-run behavior, it may be unbounded (e.g., $T = \mathbb{R}_+ = [0,\infty)$). The state space Q embodies the state sets of both the computer and its environment, i.e.,

$$Q = Q_C \times Q_E$$

where $Q_C$ and $Q_E$ can, in turn, be decomposed to represent the local state sets of computer and environmental subsystems. For our purposes, it suffices to assume that Q is discrete and, hence, for all $t \in T$ and $q \in Q$, "$X_t = q$" has a probability (i.e., $\{\omega | X_t(\omega) = q\} \in E$). The stochastic process $X_S$ is referred to as the base model of S. An instance of the base model's behavior for a fixed $\omega \in \Omega$ is a state trajectory $u_\omega : T \to Q$ where

$$u_\omega(t) = X_t(\omega), \quad \forall t \in T. \tag{2}$$

(The term "state trajectory" derives from modern usage in the theory of modeling [16]; synonyms in the more specific context of stochastic processes are "sample function," "sample path," and "realization" [14], [15].) Thus, corresponding to an underlying outcome $\omega\varepsilon\Omega$, $u_\omega$ describes how the state of the total system changes as a function of time throughout the utilization period T. Accordingly, the "description space" for the base model is the set

$$U = \{u_\omega | \omega\varepsilon\Omega\} \tag{3}$$

which is referred to as the (state) <u>trajectory space</u> of S.

As generally defined, the concept of a base model thus includes the type of queueing models used in computer performance evaluation [3] and the kind of Markov or semi-Markov models employed in reliability evaluation [7]-[9]. The intent of the definition, however, is the inclusion of less restricted base models which can represent simultaneous variations in the computer's structure and internal state (via the state set $Q_C$) and environment (via the state set $Q_E$). In other words, the emphasis here is on the modeling of degradable computing systems where changes in structure, internal state, and environment can all have an influence on the system's ability to perform. Accordingly, these base models may be regarded as generalized performance models, where structure is allowed to vary, or equivalently as generalized reliability models where variations in internal state and/or the computational environment are taken into account.

In formal terms, the user-oriented view of system behavior is likewise defined in terms of the underlying probability space $(\Omega, E, P)$. Here we assume that the user is interested in distinguishing a number of different "levels of accomplishment" when judging how well the system has performed throughout the utilization period (one such level may be total system failure). The user's "description space" is thus identified with an <u>accomplishment set</u> A whose elements are referred to alternatively as <u>accomplishment levels</u> or (user-visible) <u>performance levels</u>. A may be finite, countably infinite, or uncountable (in the last case, A is assumed to be an interval of real numbers). Thus, for example, the accomplishment set associated with a nondegradable system is $A = \{a_0, a_1\}$ where $a_0$ = "system success" and $a_1$ = "system failure." In their modeling of the PRIME system, Borgerson and Freitas [5] viewed accomplishment as the set $A = \{a_0, a_1, a_2, \ldots\}$ where $a_k$ = "k crashes during the utilization period T." If the user is primarily concerned with system "throughput," a continuous accomplishment set might be appropriate, i.e, $A = \mathbb{R}_+ = [0,\infty)$, where a number $a \epsilon A$ is the "throughput averaged over the utilization period T."

In terms of the accomplishment set, <u>system performance</u> is formally viewed as a random variable

$$Y_S : \Omega \to A \tag{4}$$

where $Y_S(\omega)$ is the accomplishment level corresponding to outcome $\omega$ in the underlying description space. Similarly, assuming that the economic gain (or loss) derived from using the system is represented by a real number r (interpreted, say, as r dollars), <u>system worth</u> is a random variable defined as

$$W_S : \Omega \to \mathbb{R} \text{ (the real numbers)} \tag{5}$$

where $W_S(\omega)$ is the worth associated with outcome $\omega$.

The terminology and notation defined above is summarized in Figure 1. Note that, at this point in the development, there are no implied relationships between these three views; their only common bond so far is that they are representations of the same system or, more formally, that they are defined on the same underlying probability space. To be useful, however, the base model $X_S$ should support the performance variable $Y_S$ in an appropriate manner (indeed, the term "base" is suggestive of this need) and, in turn, $Y_S$ should support the worth variable $W_S$. The precise nature of these connections, as they relate to the system's effectiveness, is developed in the section that follows.

## III. EFFECTIVENESS, PERFORMABILITY AND CAPABILITY

When applied to computing systems, "system effectiveness" (see [2], for example) is a measure of the extent to which the user can expect to benefit from the tasks accomplished by a computer in its use environment. More precisely, if benefit is identified with the worth $W_S$ of the system then system effectiveness is expected worth, i.e., the expectation (expected value) of the random variable $W_S$; in short

$$\mathrm{Eff}(S) = E[W_S]. \qquad (6)$$

(An implicit assumption here is that $W_S$ is defined such that $E[W_S]$ exists; see [13], for example.) Because a direct evaluation of Eff(S), using the definition of $W_S$, is generally not feasible (cf. our earlier remarks concerning the hypothetical nature of the underlying probability space), we wish to establish connections among the base model $X_S$, system performance $Y_S$ and system worth $W_S$ which can be used in the process of evaluating Eff(S).

To express system effectiveness in terms of system performance, the user's view of system worth must be compatible with system performance to the extent that $W_S$ can be formulated as a function of $Y_S$. More precisely, we assume there exists a worth function $w\colon A \to \mathbb{R}$ such that, for all $\omega \epsilon \Omega$,

$$W_S(\omega) = w(Y_S(\omega)). \tag{7}$$

If $a \epsilon A$, $w(a)$ is interpreted as the "worth of performance level a." As for the performance variable $Y_S$, a natural measure that quantifies both system performance and reliability (i.e., the ability to perform) is the probability measure induced by $Y_S$. We refer to this unified performance-reliability measure as the "performability of S" which, in terms of our modeling framework, can be generally defined as follows:

Definition 1: If S is a total system with performance $Y_S$ taking values in accomplishment set A, then the performability of S is the function $p_S$ where, for each measurable set B of accomplishment levels ($B \subseteq A$),

$$p_S(B) = P(\{\omega | Y_S(\omega) \epsilon B\}).$$

Since P is the probability measure of the underlying probability space, the interpretation of performability is straightforward, that is, for a designated set B of accomplishment levels, $p_S(B)$ is the probability that S performs at a level in B. The requirement that B be "measurable" says simply that the corresponding event $\{\omega | Y_S(\omega) \epsilon B\}$ must lie in the underlying event space, insuring that the right-hand probability is defined.

If the performance variable $Y_S$ is continuous then A must be continuous and, hence (by an earlier assumption), A is some interval of real numbers, including the possibility that $A = \mathbb{R} = (-\infty, \infty)$. In this case (or if $Y_S$ happens to be discrete and yet real-valued), we know from probability theory that the induced measure $p_S$ is uniquely determined by the probability distribution function of $Y_S$ (see [13], for example), i.e., by the function $F_{Y_S}$ (which we write simply as $F_S$) where, for all $b \varepsilon A$,

$$F_S(b) = P(\{\omega | Y_S(\omega) \leq b\}). \tag{8}$$

Moreover, $p_S$ can then be expressed as the Lebesgue-Stieltjes measure induced by $F_S$ (cf. [13], Sec. 4.5), that is, for any (measurable) set B of accomplishment levels, the performability value of B is given by

$$p_S(B) = \int_B dF_S(b). \tag{9}$$

In particular, if B is a single interval $B = (b_0, b_1]$ where $b_0 < b_1$, then

$$p_S(B) = F_S(b_1) - F_S(b_0).$$

This special case has practical significance since it quantifies the ability of S to perform within the specified limits $b_0$ and $b_1$.

If, on the other hand, $Y_S$ is a discrete random variable then each singleton set $B = \{a\}$ $(a \varepsilon A)$is measurable and $p_S$ is uniquely determined by the probability distribution of $Y_S$, i.e., by the set of values

$$\{p_S(a) | a \varepsilon A\} \tag{10}$$

where $p_S(a)$ denotes $p_S(\{a\})$. Given this distribution, if B is a set of accomplishment levels then $p_S(B)$ can be written as the sum

$$p_S(B) = \sum_{b \varepsilon B} p_S(b). \tag{11}$$

Hence, the probability distribution of $Y_S$ or, equivalently, the restriction of $p_S$ to single accomplishment levels, suffices to determine the performability. For this reason, when $Y_S$ is discrete the performability of S can be alternatively defined as follows:

<u>Definition 1a</u>: If S is a total system with performance $Y_S$ taking values in accomplishment set A and, moreover, $Y_S$ is a discrete random variable, then the <u>performability of S</u> is the function $p_S$ where, for each accomplishment level a ($a \varepsilon A$),

$$p_S(a) = P(\{\omega | Y_S(\omega) = a\}).$$

Note that Definition 1a is essentially the restriction of Definition 1 to single accomplishment levels (which have probabilities defined when $X_S$ is discrete). Conversely, given Definition 1a, its extension to Definition 1 can be obtained (at least conceptually) by applications of equation (11).

To justify the notion of performability in the context of system effectiveness, if we assume the existence of a worth function w (see (7)), then the real-valued random variable $W_S$ is a function w of the random variable $Y_S$. Moreover, we know that w is a "measurable" function (e.g., see [13], Sec. 3.8) since, prior to (7), we assumed that $W_S$ was a random variable. (Indeed, condition (7) is actually stronger than needed; its advantages, however, are its simplicity and the fact that it serves the purpose of the present discussion.) Hence, we are able to appeal to the well developed theory of functions of a random variable and, particularly, expectations where, again, it is convenient to distinguish two cases. If the performance variable $Y_S$ is continuous (and thus real-valued) with probability distribution function $F_S$ (8) then (cf. [13], Sec. 5.1).

$$E[w(Y_S)] = \int_A w(a) dF_S(a) \tag{12}$$

where the integral is a Lebesgue-Stieltjes integral. In case $Y_S$ is discrete, then (12) still applies provided the levels in A are represented by real numbers. However, independent of whether $Y_S$ is real-valued, a simpler and more familiar formulation holds in this case where, if $p_S(a)$ is as defined in (10), then

$$E[w(Y_S)] = \sum_{a \in A} w(a)p_S(a) \cdot \qquad (13)$$

By the definition of system effectiveness (6) and the fact that $W_S = w(Y_S)$ (7), we have

$$Eff(S) = E[W_S] = E[w(Y_S)]$$

and, accordingly, (12) and (13) are formulations of the effectiveness of S. Moreover, we see that each formula involves the worth function w and the performability $p_S$ (although $p_S$ does not occur explicitly in (12), recall that $F_S$ characterizes $p_S$ (9)). In other words, relative to the system-user interface delineated by the accomplishment set A, effectiveness evaluation may be decomposed into worth evaluation (on the user side of the interface) and performability evaluation (on the system side). Consequently, looking in toward the system, performability emerges as a key measure with regard to evaluations of system effectiveness.

To further justify the concept of performability, we note that traditional evaluations of computer performance and computer reliability are concerned with special types of performability. Performance evaluation is concerned with evaluating $p_S$ under the assumption that the computer part of S is fixed (i.e., its structure does not change as the consequence of internal faults). Reliability evaluation is concerned with evaluating $p_S(B)$ where B is a designated subset of accomplishment levels associated with system "success." If A is finite, a performability evaluation can alternatively be regarded as $|A|$ reliability evaluations, one for each singleton success set $B = \{a\}$, and the evaluation may actually be carried out in this manner. As this process is generally more complex than a typical reliability evaluation procedure (in particular, it involves distinguishing all the performance levels as well as determining their probabilities), we reserve the term "reliability evaluation" to mean the evaluation of "probability of success" for some specified success criterion B. Moreover, even when $|A| = 2$, we find (as discussed later in this section) that performability models represent a proper extension of models typically employed in reliability evaluation.

As a final remark regarding justification, we have found that when system performance is not degradable (as in the case, for example, with fault-tolerant architectures which employ standby sparing [4], N modular redundancy [17], or combinations thereof), it is possible to treat performance and reliability as separate issues in the process of evaluating system effectiveness. On the other hand, if performance is degradable, it can be shown (see [18]) that the more general concept of performability must typically be invoked (as in equations (12) and (13), for example) when evaluating system effectiveness.

With performability established as the object of the evaluation process, we are now in a position to specify how the base model process $X_S$ (1) must relate to the performance variable $Y_S$ (4) if it is to support an evaluation of the performability $p_S$. To precisely state this relationship, we suppose that $X_S$ is specified by its finite-dimensional probability distributions (or by information that determines these distributions) and we let Pr denote the probability measure (defined on a $\sigma$-algebra of subsets of U) which is uniquely determined by these finite-dimensional distributions (see [14], for example). If Pr is defined for a trajectory set V ($V \subseteq U$) then, relative to the underlying measure P,

$$Pr(V) = P(\{\omega | u_\omega \varepsilon V\}), \qquad (14)$$

i.e., Pr(V) is the probability that an observed state trajectory $u_\omega$ (see (2)) lies in the set V. In practice, however, Pr(V) will be calculated directly from the finite-dimensional distributions that determine Pr. The measure Pr thus serves to formally describe the probabilistic nature of the base model $X_S$.

For $X_S$ to support $Y_S$, we now impose the following restrictions. We assume first that the base model is refined enough to distinguish the levels of accomplishment perceived by the user, that is, for all $\omega,\omega' \varepsilon \Omega$,

$$Y_S(\omega) \neq Y_S(\omega') \text{ implies } u_\omega \neq u_{\omega'} \qquad (15)$$

where $u_\omega$ and $u_{\omega'}$ are the state trajectories associated with outcomes $\omega$ and $\omega'$. This implies that each trajectory $u \varepsilon U$ is related to a unique accomplishment level $a \varepsilon A$. In addition, we assume that the probabilistic nature of $Y_S$ is determinable from that of $X_S$. More precisely, if B is a measurable set of accomplishment levels, i.e., the set $\{\omega | Y_S(\omega) \varepsilon B\}$ is in the domain of the underlying measure P, then we require that the corresponding trajectory set

$$U_B = \{u_\omega | Y_S(\omega) \varepsilon B\}$$

lie in the domain of the base model measure Pr; in short

If B is measurable then Pr is defined for $U_B$. (16)

Given that conditions (15) and (16) are satisfied, we can establish a link between $X_S$ and $Y_S$ which, in the context of effectiveness modeling [18], is generally referred to as the "capability" of S. Adopting this terminology, we have

<u>Definition 2</u>: If S is a system with trajectory space U and accomplishment set A then the <u>capability function of S</u> is the function $\gamma_S: U \to A$ where $\gamma_S(u)$ is the level of accomplishment resulting from state trajectory u, that is,

$$\gamma_S(u) = a \text{ if, for some } \omega \varepsilon \Omega,\ u_\omega = u \text{ and } Y_S(\omega) = a.$$

Condition (15) insures that the capability function $\gamma_S$ is well-defined (i.e., it deserves the name "function"), for if $u_\omega = u_{\omega'}$ then

$\gamma_S(u_\omega) = \gamma_S(u_{\omega'})$. Condition (16) guarantees that the inverse $\gamma_S^{-1}$ of the capability function ($\gamma_S^{-1}$ is a relation between A and U but generally not a function) will carry sets that are measurable with respect to $Y_S$ into sets that are measurable with respect to $X_S$. To substantiate this fact, suppose that B is a measurable set of accomplishment levels. Then the inverse image of B is the trajectory set

$$\gamma_S^{-1}(B) = \{u|\gamma_S(u)\varepsilon B\}$$

or, equivalently, by the definition of $\gamma_S$ (Definition 2),

$$\gamma_S^{-1}(B) = \{u_\omega|Y_S(\omega)\varepsilon B\}$$
$$= U_B.$$

But condition (16) insures that Pr is defined for $U_B$ and, hence, $\gamma_S^{-1}(B)$ is measurable where

$$Pr(\gamma_S^{-1}(B)) = Pr(U_B). \tag{17}$$

In effect, therefore, conditions (15) and (16) say that $\gamma_S$ can be viewed as a random variable defined on the probability space (with measure Pr) induced by the base model $X_S$. Of more practical significance, however, is the fact that, under these conditions, $X_S$ and $\gamma_S$ suffice to determine the performability of S. To argue the latter, if B is measurable then, by (14),

$$Pr(U_B) = P(\{\omega|u_\omega\varepsilon U_B\})$$
$$= P(\{\omega|Y_S(\omega)\varepsilon B\})$$

which, by Definition 1, is just the performability of S for accomplishment levels B, i.e.,

$$Pr(U_B) = p_S(B). \tag{18}$$

Combining equations (17) and (18), we conclude that

$$p_S(B) = Pr(\gamma^{-1}(B)), \tag{19}$$

substantiating the fact that $X_S$ and $\gamma_S$ (which determine $P_r$ and $\gamma_S^{-1}$, respectively) suffice to support an evaluation of the performability $p_S$.

In view of what has just been observed, if $X_S$ and $Y_S$ admit to the definition of a capability function $\gamma_S$ (in which case we presume that conditions (15) and (16) are satisfied), the pair $(X_S, \gamma_S)$ is said to constitute a performability model of S. If B is a (measurable) set of accomplishment levels, the inverse image $\gamma_S^{-1}(B) = U_B$ is referred to as the trajectory set of B, where its determination requires an analysis of how levels in B relate back down via $\gamma_S^{-1}$ to trajectories of the base model. $p_S(B)$ is then determined by a probability analysis of $\gamma_S^{-1}(B)$. In case $Y_S$ is discrete (Def. 1a), it suffices to consider inverse images of the form $\gamma_S^{-1}(a)$ where $a \varepsilon A$. Methods of implementing this process in the discrete case are discussed in Section IV.

The role of a capability function in performability evaluation is similar to that of a "structure function" [19] in reliability evlauation. However, even when performability is restricted to reliability, the concept of a capability function is more general. The special class which corresponds to the use of structure functions in "phased mission" analysis (see [20], for example) may be characterized as follows. Let S be a system where Q is the state space of the base model and $A = \{0,1\}$ is the accomplishment set (here, 1 denotes "success" and 0 denotes "failure"). Then a capability function $\gamma_S$ is structure-based if there exists a decomposition of T into k consecutive time periods $T_1, T_2, \ldots, T_k$ and there exist functions $\varphi_1, \varphi_2, \ldots, \varphi_k$ with $\varphi_i: Q \rightarrow \{0,1\}$ such that, for all $u \varepsilon U$,

$$\gamma_S(u) = 1 \text{ iff } \varphi_i(u(t)) = 1, \qquad (20)$$

for all $i \in \{1, 2, \ldots, k\}$ and for all $t \in T_i$. In the context of "phased mission" analysis, $T_i$ is referred to as the ith phase (of the mission) and $\varphi_i$ is the structure function of the ith phase. For each function $\varphi_i$, the inverse image $\varphi_i^{-1}(1)$ can be interpreted as the set of "success states" of the ith phase and, accordingly, (20) says that S performs successfully ($\gamma_S(u) = 1$) if and only if $u(t)$ is a success state throughout each phase. Thus, the advantage of a structure-based formulation is that each phase may be treated independently when determining the set $\gamma_S^{-1}(1)$ of all successful state trajectories.

If system success is viewed in structural terms, as is the case in most reliability studies, a structure-based capability function will usually suffice. On the other hand, when success relates to system performance we find that capability may no longer be expressible in terms of locally defined success criteria as specified by the structure functions $\varphi_i$. The following example serves to demonstrate this fact.

Let $S = (C,E)$ where C represents a distributed computer comprised of n subsystems, and E represents the computer's workload. Suppose further that system "throughput" (i.e., the user-visible work rate of C in E) varies as a function of the number of faulty subsystems. For our purposes here, it suffices to assume that the workload E is constant and, hence, the operational states of S can be represented by the state space $Q = \{q_0, q_1, \ldots, q_n\}$ where state $q_i$ corresponds to "i faulty subsystems." The variation in throughput is described by a function $\tau: Q \to \mathbb{R}_+$ where $\tau(i)$ = the throughput of S in state $q_i$.

Assuming S is used continuously throughout a utilization period $T = [0,h]$ of duration $h > 0$, the base model of S is a stochastic process $X_S = \{X_t | t\varepsilon[0,h]\}$ where each $X_t$ is a random variable taking values in Q. (The probabilistic nature of $X_S$ is not an issue here.) As for performance, suppose that the user is interested in the average throughput of the system, where the average is taken over the utilization period T. Suppose further that system "success" is identified with a minimum average throughput $\overline{\tau}$. Then the capability function of S is the function $\gamma_S: U \to \{0,1\}$ where

$$\gamma_S(u) = \begin{cases} 1 \text{ if } \frac{1}{h}\int_0^h \tau(u(t))dt \geq \overline{\tau} \\ 0 \text{ otherwise.} \end{cases} \qquad (21)$$

Due to the inherent memory of the integration operation, we find that $\gamma_S$ does not admit to a structure-based formulation. To verify this fact with a simple 2-state example, suppose $Q = \{q_0,q_1\}$, $\tau(q_0) > \tau(q_1)$, and $\overline{\tau} = (\tau(q_0)-\tau(q_1))/2$. Then, according to (21), $\gamma_S(u) = 1$ iff the total time for which $u(t) = q_0$ is at least $h/2$. In particular, this says that more than one trajectory results in success, i.e., $|\gamma_S^{-1}(1)| > 1$. To prove that $\gamma_S$ is not structure-based let us suppose to the contrary, that is, there exist phases $T_1, T_2, \ldots, T_k$ and structure functions $\varphi_1, \varphi_2, \ldots, \varphi_k$ such that (20) is satisfied. If we let $R_i$ denote the success states of phase i, i.e., $R_i = \varphi_i^{-1}(1)$, then $R_i \neq \phi$, for all i, or otherwise no trajectory results in success. It must also be the case that $R_i \neq \{q_0,q_1\}$ for all i, for if $R_i = \{q_0,q_1\}$ (all states are success states during phase i) then the condition $\varphi_i(u(t)) = 1$, $\forall t\varepsilon T_i$ (see

(20)) is always satisfied, that is, phase i can be ignored when determining whether u spends at least half its time in state 0. This is clearly impossible if the duration of $T_i$ is at least h/2. If the duration of $T_i$ is less than h/2, trajectories u and v can be found such that $u(t) = v(t)$, $\forall t \in (T-T_i)$, and yet $\gamma_S(u) \neq \gamma_S(v)$ contradicting the ability to ignore phase i. The only remaining alternative is that $|R_i| = 1$, for all i, that is, each phase has exactly one success state which, in turn, implies that there is exactly one success trajectory u. This contradicts our initial observation that $|\gamma_S^{-1}(1)| > 1$ and proves that $\gamma_S$ is not structure-based.

We can conclude, therefore, that even in the case of two accomplishment levels, the concept of a capability function (Def. 2) represents a proper extension of relations between state behavior and system performance that are typically assumed in the theory of reliability. Moreover, we have found that this extension permits the phases of a utilization period to be "functionally dependent" in a precisely defined sense, whereas the phases associated with a structure-based capability function must be functionally independent. The reader is referred to [21] for a more complete discussion of functional dependence and its implications.

## IV. PERFORMABILITY EVALUATION

As established in the previous section (see (19)), if $(X_S, \gamma_S)$ is a performability model then the performability of S for a set of accomplishment levels B may be expressed as $p_S(B) = Pr(\gamma_S^{-1}(B))$. Accordingly, one method of evaluating a particular $p_S(B)$ is to (i) determine $\gamma_S^{-1}(B)$ and then (ii) evaluate $Pr(\gamma_S^{-1}(B))$. Since the "distance" between the base model $X_S$ and the accomplishment set A may be considerable, step (i) can be facilitated by introducing additional models between $X_S$ and A.

In general, each intermediate model is defined in a manner similar to that of the base model. More precisely, if there are m+1 levels in the hierarchy, the level-i model (i = 0, 1, ..., m, where level-0 is the least detailed model at the "top" of the hierarchy) is a stochastic process

$$X^i = \{X_t^i | t \in T^i\}, \; T^i \subseteq T$$

where, for a fixed $t \in T^i$, $X_t^i$ is a random variable taking values in a set $Q^i$, the state space of $X^i$. The state space $Q^i$ is generally composed of two components, i.e.,

$$Q^i = Q_c^i \times Q_b^i$$

where $Q_c^i$ is the composite state set and $Q_b^i$ is the basic state set (at level-i). States in the composite part $Q_c^i$ represent a less detailed view of the operational status of the system than do states in $Q^{i+1}$, such that the state behavior at level-(i+1) uniquely determines the composite state behavior at level-i (this will be made more precise in a moment). States in $Q_b^i$, on the other hand, represent basic information not conveyed by states in $Q^{i+1}$, i.e., $Q_b^i$ is a coordinate set of the base model state s ce Q. In case there is no composite (alternatively, basic) part at level-i, $Q_c^i(Q_b^i)$ is simply deleted, that is, $Q^i = Q_b^i$ $(Q^i = Q_c^i)$. In particular, the above definition precludes a composite state set at level-m (the "bottom" level of the hierarchy) and, hence, $Q^m = Q_b^m$.

In specifying the model hierarchy, it is convenient to view $X^i$ as a pair of processes which determine the projections on $Q_c^i$ and $Q_b^i$, respectively. (If one of $Q_c^i$ or $Q_b^i$ does not exist, this pair reduces to a single process.) More precisely, given $Q_c^i$, the <u>composite process</u> (at level-i) is the stochastic process

$$X_c^i = \{X_{c,t}^i \mid t\varepsilon T_c^i\},\ T_c^i \subseteq T^i$$

where the random variables $X_{c,t}^i$ take values in $Q_c^i$. For a fixed outcome $\omega$ in the underlying sample space $\Omega$, a <u>composite state trajectory</u> is a function $u_{c,\omega}:T_c^i \to Q_c^i$ where $u_{c,\omega}(t) = X_{c,t}^i(\omega)$; the <u>composite trajectory space</u> is the set $U_c^i = \{u_{c,\omega} \mid \omega\varepsilon\Omega\}$. Similar definitions, terminology, and notation apply to the <u>basic process</u> $X_b^i$. To permit extension of either $X_c^i$ or $X_b^i$ to larger time bases, a fictitious state ¢ is adjoined to each of $Q_c^i$ and $Q_b^i$ so that if $t \not\in T_c^i$ (similar remarks apply to the basic part) then $X_{c,t}^i$ is defined to be a degenerate random variable that always assumes the value ¢, i.e.,

$$X_{c,t}^i(\omega) = ¢,\ \text{for all } \omega\varepsilon\Omega. \qquad (22)$$

If $X_c^i$ and $X_b^i$ are so extended to $T^i$, and we take $X^i$ to be the process whose projections on $Q_c^i$ and $Q_b^i$ are $X_c^i$ and $X_b^i$, respectively, then $X^i$ is uniquely determined by $X_c^i$ and $X_b^i$. (Note that the processes $X_c^i$ and $X_b^i$ may be statistically dependent.)

By the above observation, we can alternatively regard the level-i model as the pair of processes

$$X^i = (X^i_c, X^i_b)$$

which is a convenient view for the purpose of specifying interlevel relationships. With this identification, a state trajectory of $X^i$ is viewed as a pair of trajectories, i.e., the <u>trajectory space</u> $U^i$ (at level-i) is taken to be the set $U^i_c \otimes U^i_b$ where

$$U^i_c \otimes U^i_b = \{(u_{c,\omega}, u_{b,\omega}) | \omega \varepsilon \Omega\}.$$

In case there is no composite (alternatively, basic) state set at level -i, the above representations of $X^i$ and $U^i$ are understood to be their appropriate single component versions.

The required relationship of these models to the base model, the accomplishment set, and the capability function is prescribed by the following definition.

<u>Definition 3</u>: If S is a total system with base model $X_S$ and capability function $\gamma_S$, the collection $\{X^0, X^1, \ldots, X^m\}$ of level-0 to level-m models is a <u>model hierarchy for S</u> if the following conditions are satisfied.

(i) $X^m = X^m_b$, that is, all variables of the bottom model are basic.

(ii) If each model $X^i$ is extended to the utilization period T, the base model $X_S$ is the stochastic process $X_S = \{X_t | t\varepsilon T\}$ where $X_S = (X^m_{b,t}, X^{m-1}_{b,t}, \ldots, X^0_{b,t})$. Accordingly, the state space of $X_S$ is $Q = Q^m_b \times Q^{m-1}_b \times \ldots \times Q^0_b$ and the trajectory space U is represented by the set $U^m_b \otimes U^{m-1}_b \otimes \ldots \otimes U^0_b$.

(iii) For each level i, there exists an <u>interlevel translation</u> $\kappa_i$ where

$$\kappa_0: U_c^0 \otimes U_b^0 \to A$$

$$\kappa_i: U_c^i \otimes U_b^i \to U_c^{i-1} \quad (1<i<m)$$

$$\kappa_m: U_b^m \to U_c^{m-1}$$

such that the capability function $\gamma_S$ can be decomposed as follows. If $u \varepsilon U$ where $u = (u_m, u_{m-1}, \ldots, u_0)$ with $u_i \varepsilon U_b^i$, then

$$\gamma_S(u) = \kappa_0(\kappa_1(\ldots \kappa_{m-1}(\kappa_m(u_m), u_{m-1})\ldots), u_0). \tag{23}$$

The terminology and notation of Definition 3 is summarized in Figure 2 where a) is the original model and b) is the hierarchical model.

A model hierarchy thus provides a step-by-step formulation of the capability function in terms of interlevel translations of state trajectories, beginning with a translation of the bottom model. It also permits the expression of capability relative to higher level (less detailed) views of total system behavior. More precisely, let $\overline{U}^i$ denote the level-i trajectory space, along with all the basic trajectory spaces of higher level models, i.e.,

$$\overline{U}^i = U^i \otimes U_b^{i-1} \otimes \ldots \otimes U_b^0.$$

(Note that, at the extremes, $\overline{U}^0 = U^0$ and $\overline{U}^m = U$.) Then the <u>level-i based capability function</u> is the function

$$\gamma_i: \overline{U}^i \to A$$

defined inductively as follows. If $i = 0$ and $u \varepsilon \overline{U}^0$, then

$$\gamma_0(u) = \kappa_0(u). \tag{24}$$

If $i > 0$ and $(u,u') \varepsilon \overline{U}^i$ where $u \varepsilon U^i$ and $u' \varepsilon U_b^{i-1} \otimes \ldots \otimes U_b^0$, then

$$\gamma_i(u,u') = \gamma_{i-1}(\kappa_i(u),u'). \tag{25}$$

It is easily shown that $\gamma_i$ has its intended interpretation, i.e., if u and u' correspond to a base model trajectory v then $\gamma_i(u,u') = \gamma_S(v)$. In particular, if i = m then $\gamma_m = \gamma_S$.

The practical significance of the model hierarchy, however, is the ability to formulate the inverse of $\gamma_S$ via the inverses of the $\gamma_i$, thereby providing a step-by-step, top-down method of elaborating a set of accomplishment levels B. Beginning with level-0-based capability, by (24) we have

$$\gamma_0^{-1}(B) = \kappa_0^{-1}(B). \tag{26}$$

Assuming that $\gamma_{i-1}^{-1}(B)$ has been determined, by (25) it follows that

$$\gamma_i^{-1}(B) = \bigcup_{(u,u')\varepsilon\gamma_{i-1}^{-1}(B)} (\kappa_i^{-1}(u),u'), \tag{27}$$

where $(\kappa_i^{-1}(u),u') = \{(v,u') | \kappa_i(v) = u\}$. This process is iterated until i = m, yielding $\gamma_m^{-1}(B) = \gamma_S^{-1}(B)$.

## V. CONCLUSION

In conclusion, we see that performability modeling with continuous accomplishment sets can proceed in much the same manner as modeling with discrete accomplishment sets. The essential differences are that measurability conditions must be observed more closely in the continuous case when defining performability (Definition 1) and capability (see condition 1b)). Also methods of actually specifying $\gamma_S$ or, in case a hierarchy is used, the inter-level translations $\kappa_i$, differ in the two cases: in the continuous case, values of these functions must be specified by formulas for computing their values; in the discrete case there are situations (e.g., when A is finite) where $\gamma_S$ or the $\kappa_i$ can be tabulated completely.

Finally, regarding solutions (although our subject here is not solution techniques, per se), if A is continuous it may often be the case that a complete knowledge of performability, i.e., the ability to compute $p_S(B)$ for <u>every</u> measurable set B, is not required. For example, intervals of the form

(i) $B = (b_o,b_1]$, $(b_o<b_1)$

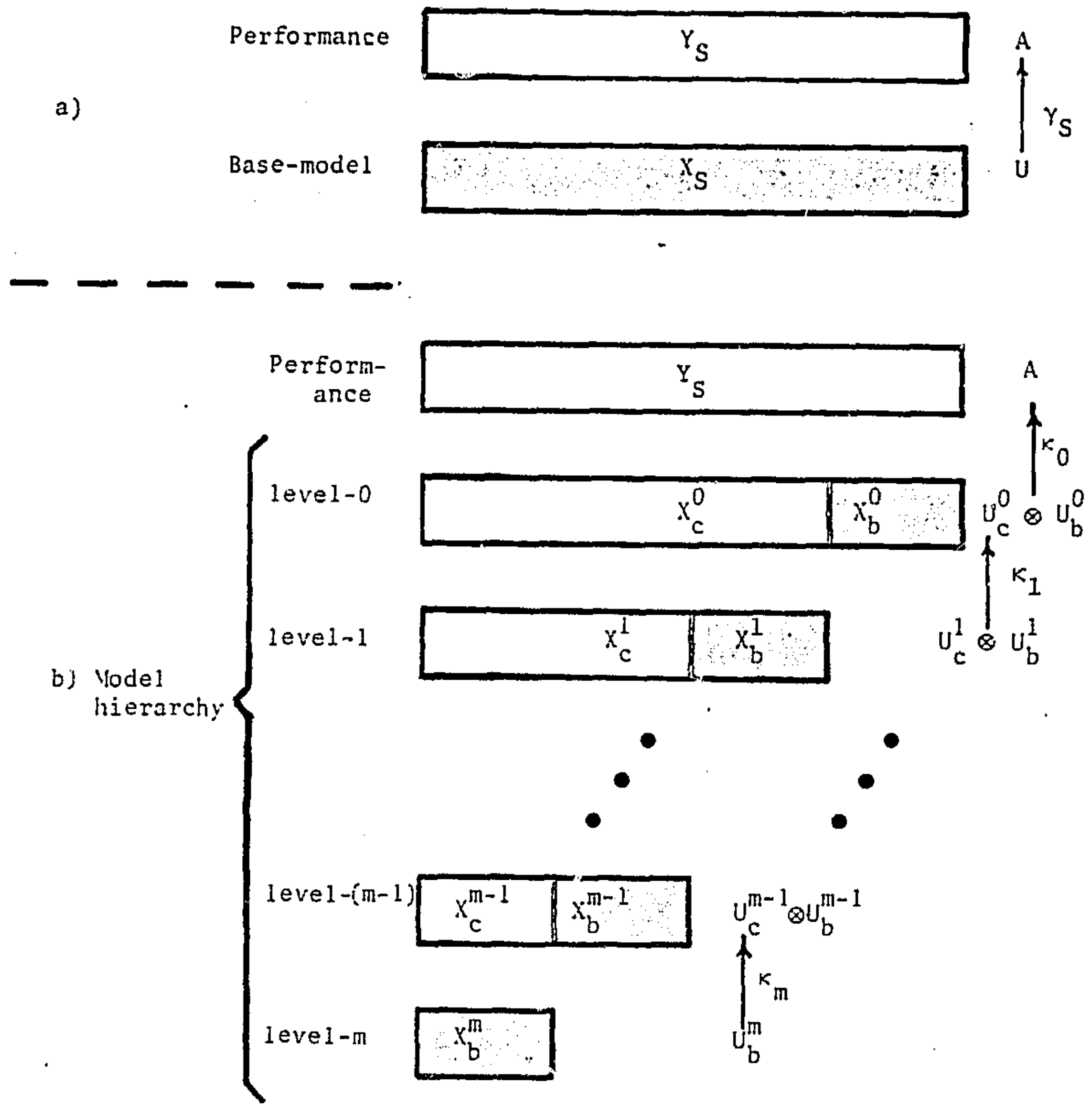
or

(ii) $B = (B_o,\infty)$

may suffice where, in case (i), $p_S(B)$ is the probability that S performs within specified limits $b_o$ and $b_1$ and, in case (ii) $p_S(B)$ is the probability that the performance of S is greater than a specified minimum $b_o$. In other cases, even less information may satisfy the evaluation needs of the user, e.g., the mean $E[Y_S]$ amd variance var $[Y_S]$ of the performance variable $Y_S$. Although these are not performability measures in the strict sense of Definition 1, they are "performability-based" in the sense that the performability $p_S$ uniquely determines $E[Y_S]$ and var $[Y_S]$. Moreover, the latter should be easier to evaluate since generally, as experienced in the performance evaluation of fault-free systems [2], [3], full knowledge of $p_S$ is not required in the process of determining $E[Y_S]$ and var $[Y_S]$.

| Model | Description Space |
|---|---|
| Base model $X_S$ | Trajectory space $U$ |
| System performance $Y_S$ | Accomplishment set $A$ |
| System worth $W_S$ | The real numbers $\mathbb{R}$ |

Figure 1



Performance
$Y_S$
a)
$A$
$Y_S$
$U$
Base-model
$X_S$
Perform-ance
$Y_S$
$A$
$\kappa_0$
level-0
$X_c^0$
$X_b^0$
$U_c^0 \otimes U_b^0$
$\kappa_1$
level-1
$X_c^1$
$X_b^1$
$U_c^1 \otimes U_b^1$
b) Model hierarchy
level-(m-1)
$X_c^{m-1}$
$X_b^{m-1}$
$U_c^{m-1} \otimes U_b^{m-1}$
$\kappa_m$
level-m
$X_b^m$
$U_b^m$

Figure 2

## REFERENCES

[1] J. F. Meyer, "On Evaluation the Performability of Degradable Computing Systems," Proc. 1978 Int'l Symp. on Fault-Tolerant Computing, Toulouse, France, pp. 44-49, June, 1978.

[2] L. Svobodova, Computer Performance Measurement and Evaluation Methods: Analysis and Applications. New York, NY: Elsevier, 1976.

[3] H. Kobayashi, Modeling and Analysis: An Introduction to System Performance Evaluation Methodology. Reading, MA: Addison-Wesley, 1978.

[4] W. G. Bouricius, W. C. Carter and P. R. Schneider, "Reliability modeling techniques for self-repairing computer systems," Proc. 24th ACM Nat'l Conf., pp. 295-305, August 1969.

[5] B. R. Borgerson and R. F. Freitas, "A reliability model for gracefully degrading and standby-sparing systems," IEEE Trans. Computers, vol. C-24, pp. 517-525, May 1975.

[6] H. B. Baskin, B. R. Borgerson and R. Roberts, "PRIME-A modular architecture for terminal oriented systems," 1972 Spring Joint Computer Conf., AFIPS Conf. Proc., vol. 40, pp. 431-437, 1972.

[7] J. C. Laprie, "Reliability and availability of repairable structures," Dig. 1975 Int'l Symp. on Fault-Tolerant Computing, Paris, FR., pp. 87-92, June 1975.

[8] Y. W. Ng and A. Avizienis, "A reliability model for gracefully degrading and repairable fault-tolerant systems," Proc. 1977 Int'l Symp. on Fault-Tolerant Computing, Los Angeles, CA, pp. 22-28, June 1977.

[9] A. Costes, C. Landrault, and J. C. Laprie, "Reliability and availability models for maintained systems featuring hardware failures and design faults," IEEE Trans. Comput., vol. C-27, pp. 548-560, June 1978.

[10] M. D. Beaudry, "Performance related reliability measures for computing systems," IEEE Trans. Comput., vol. C-27, pp. 540-547, June 1978.

[11] R. Troy, "Dynamic reconfiguration: An algorithm and its efficiency evaluation," Proc. 1977 Int'l Symp. on Fault-Tolerant Computing, Los Angeles, CA, pp. 44-49, June 1977.

[12] J. Losq, "Effects of failures on gracefully degradable systems," Proc. 1977 Int'l Symp. on Fault-Tolerant Computing, Los Angeles, CA, pp. 29-34, June 1977.

[13] P. E. Pfeiffer, Concepts of Probability Theory. New York, NY: McGraw-Hill, 1965.

[14] E. Wong, Stochastic Processes in Information and Dynamical Systems. New York, NY: McGraw-Hill, 1971.

[15] E. Cinlar, Introduction to Stochastic Processes. Englewood Cliffs, NJ: Prentice-Hall, Inc., 1975.

[16] B. P. Zeigler, Theory of Modelling and Simulation. New York, NY: John Wiley & Sons, 1976.

[17] F. P. Mathur and A. Avizienis, "Reliability analysis and architecture of a hybrid-redundant digital system: Generalized triple modular redundancy with self-repair," Proc. 1970 Spring Joint Computer Conf., AFIPS Conf., Vol. 36, pp. 375-383, 1970.

[18] J. F. Meyer, "Models and techniques for evaluating the effectiveness of aircraft computing systems," Semi-Annual Status Report Number 3, NASA Report CR158992, January 1978.

[19] Z. W. Birnbaum, J. F. Esary and S. C. Saunders, "Multicomponent systems and structures and their reliability," Technometrics, Vol. 3, pp. 55-57, February 1961.

[20] J. D. Esary and H. Ziehms, "Reliability of phased missions," Reliability and Fault Free Analysis, SIAM, Philadelphia, PA, pp. 213-236, 1975.

[21] R. A. Ballance and J. F. Meyer, "Functional dependence and its application to system evaluation," Proc. 1978 Conf. on Info. Sciences and systems, The Johns Hopkins Univ., Baltimore, MD, pp. 280-285, March 1978.

[22] J. F. Meyer, "Models and techniques for evaluating the effectiveness of aircraft computing systems," Semi-Annual Status Report Number 4, NASA Report CR158993, July 1978.

[23] A. L. Hopkins, Jr., et al., "FTMP: A highly reliable fault-tolerant multiprocessor for aircraft," Proc. of the IEEE, vol. 66, no. 10, pp. 1221-1239, Oct. 1978.

[24] J. H. Wensley, et al., "SIFT: Design and analysis of a fault-tolerant computer for aircraft control," Proc. of the IEEE, vol. 66, no. 10, pp. 1240-1255, Oct. 1978.

[25] D. G. Furchtgott and J. F. Meyer, "Performability evaluation of fault-tolerant multiprocessors," Digest 1978 Government Microcircuit Applications Conf., Monterey, CA, Nov. 1978.

[26] J. F. Meyer, D. G. Furghtgott, and L. T. Wu, "Performability evaluation of the SIFT computer," Proc. 1979 Int'l Symp. on Fault-Tolerant Computing, Madison, WI, pp. 43-50, June 1979.

[27] L. T. Wu and J. F. Meyer, "Phased models for evaluating the performability of computing systems," Proc. of the 1979 Johns Hopkins Conference on Information Sciences and Systems, Balitmore, MD, pp. 426-431, March 1979.