From W. H. Sanders and J. F. Meyer, "A Unified Approach for Specifying Measures of Performance, Dependability, and Performability," in Dependable Computing for Critical Applications, Vol 4: of Dependable Computing and Fault-Tolerant Systems (ed., A. Avizienis and J. Laprie), Springer-Verlag, 1991, pp. 215-237.

A UNIFIED APPROACH FOR SPECIFYING MEASURES OF PERFORMANCE, DEPENDABILITY, AND PERFORMABILITY*

W. H. Sanders† and J. F. Meyer‡

†The University of Arizona
Computer Engineering Research Laboratory
Department of Electrical and Computer Engineering
Tucson, AZ 85721
whs@ece.arizona.edu
and
‡The University of Michigan
Computing Research Laboratory
Department of Electrical Engineering and Computer Science
Ann Arbor, MI 48109
jfm@eecs.umich.edu

ABSTRACT

Methods for evaluating system performance, dependability, and performability are becoming increasingly more important, particularly in the case of critical applications. Central to the evaluation process is the definition of specific measures of system behavior that are of interest to a user. This paper presents a unified approach to the specification of measures of performance, dependability, and performability. The unification is achieved by 1) using a model class well suited for representation of all three aspects of system behavior, and 2) defining a variable class which allows for the specification of a wide range of measures of system behavior. The resulting approach permits the specification of many non-traditional as well as traditional measures of system performance, dependability, and performability in a unified manner. Example instantiations of variables within this class are given and their relationships to variables used in traditional performance and dependability evaluations are illustrated.

Keywords: Performance Evaluation, Dependability Evaluation, Performability Evaluation, Reward Models, Stochastic Petri Nets.

*This work was supported in part by the Digital Equipment Corporation Faculty Program: Incentives for Excellence and the Office of Naval Research under Contract No. N00014-85-K-0531.

I Introduction

With growth in the complexity of computing systems and their applications, means of system evaluation are becoming increasingly more complex and difficult. One source of this difficulty is the dependence of what is to be evaluated, as reflected by the measures employed, on the specific nature of the system's application. This is particularly so in the case of critical applications since, here, the measures used must indeed capture what the user perceives as crucial to successful operation. Development of methods for the evaluation of system performance (see [5, 12, 14, 28], for example), dependability (see [13, 2], for example), and performability [17, 18] has thus become an activity of recognized importance. Central to this activity is the definition of specific measures of performance, dependability, and performability that are of interest to a user. Many different measures in these categories have been proposed. Typically, however, the measure definition is linked to a particular class of models for that measure, and different measures require different model classes. This paper presents a unified approach to the specification of measures of performance, dependability, and performability. The unification is achieved by 1) using a model class well suited for representation of all three aspects of system behavior, and 2) defining a variable class which allows for the specification of a wide range of measures of system behavior.

The model class used for system representation is a stochastic extension of Petri nets known as "stochastic activity networks." Stochastic activity networks (SANs) [21, 24] were developed to facilitate unified performance/dependability evaluation and have features which permit the representation of parallelism, timeliness, fault tolerance, and degradable performance [20]. Through the introduction of several new primitives (relative to Petri nets), they allow a model to be specified in a convenient way, as evidenced in applications of SANs to computer networks (e.g., [1, 16, 22]), computer systems (e.g., [23]), and automated manufacturing systems, while providing the formal structure necessary for analytic solutions [25]. When model characteristics preclude analytical evaluation, performability (as well as performance and dependability) can be evaluated via simulation.

However, before methods to do this can be developed, it is necessary to specify the range of measures (i.e. "types" of variables) that may be considered. To some extent, this range is determined by the choice of representation scheme. For example, if queueing networks are employed as the representation scheme, one is typically limited to asking questions

regarding server utilizations, queue lengths, waiting times, and service times. If stochastic activity networks are used, the class is larger, due to the lower-level nature of the model primitives.

It is therefore useful to formally categorize measures of system behavior in a manner that suggests methods which may be used to obtain their solution. Previous work done regarding "reward models" [10] (and associated "reward variables") provides an instructive example in this regard. Informally, a reward model consists of a stochastic process and a "reward structure". The reward structure relates possible behaviors of the process to a specified performance variable. Typically, this is done by associating a "reward rate" with each state, the interpretation being that this rate is the rate at which reward accumulates while the process is in the state. The performance variable in this case is then taken to be the reward accumulated over some utilization interval (either finite or infinite). By associating different reward rates to states, one can construct performance variables with many different interpretations.

We take a similar approach in this paper, but develop reward structures that quantify behaviors at the stochastic activity network level, instead of the state level. This approach has several distinct advantages over the state-level approach outlined above. First, the assignment of rewards and interpretation of solutions is more natural, since it is done at the level at which the modeler thinks. Second, since rewards are assigned at the network level, they can be used in the construction procedure (i.e. the procedure by which a stochastic process or simulation program is generated from the network representation and performance variable).

The remainder of this paper is organized as follows. In the next section, the basic definitions and concepts concerning stochastic activity networks are reviewed. Traditional reward models and variables are then reviewed and a general framework for classifying reward variables based on the "type" of their reward structure is given. This framework is then used to generate particular variable types that will be considered. Variables based on a particular type of reward structure which captures information regarding activity completions and numbers of tokens in places are then investigated. Finally, example instantiations of variables of this type are given and their relationships to variables used in traditional performance and dependability evaluations are illustrated.

II Stochastic Activity Networks

Stochastic activity networks (SANs) [21, 24] incorporate features of both stochastic Petri nets and queueing models. Structurally, SANs have primitives consisting of *activities, places, input gates, and output gates.* Activities ("transitions" in Petri net terminology) are of two types, *timed* and *instantaneous.* Timed activities represent activities of the modeled system whose durations impact the system's ability to perform. Instantaneous activities, on the other hand, represent system activities which, relative to the performance variable in question, complete in a negligible amount of time. Cases associated with activities permit the realization of two types of spatial uncertainty. Uncertainty about which activities are enabled in a certain state is realized by cases associated with intervening instantaneous activities. Uncertainty about the next state assumed upon completion of a timed activity is realized by cases associated with that activity. Places are as in Petri nets. Gates are introduced to permit greater flexibility in defining enabling and completion rules.

The stochastic nature of the nets is realized by associating an *activity time distribution function* with each timed activity and a *probability distribution* with each set of cases. Generally, both distributions can depend on the global marking of the network. A *reactivation function* [21] is also associated with each timed activity. This function specifies, for each marking, a set of *reactivation markings.* Informally, given that an activity is activated in a specific marking, the activity is *reactivated* whenever any marking in the set of reactivation markings is reached. This provides a mechanism for restarting activities that have been activated, either with the same or different distribution. This decision is made on a per activity basis (based on the reactivation function), and is not a net-wide execution policy.

The execution of stochastic activity networks is discussed in detail in several places, including [25]. Informally, SANs execute in time through completions of activities that result in changes in markings. More specifically, an activity is chosen to *complete* in the current marking based on the relative priority among activities (instantaneous activities have priority over timed activities) and the activity time distributions of *enabled* activities. A case of an activity chosen to complete is then selected based on the probability distribution for that set of cases. These two choices determine uniquely the next marking of the network, which is then obtained by executing the input gates connected to the input of the activity chosen and the output gates connected to the chosen case. This procedure is repeated by considering the activities enabled in the new marking.

Stochastic activity networks can be solved by both analysis and simulation, depending on system characteristics. Informally, SANs can be solved via analytic methods when all activity time distributions are exponential and activities are reactivated often enough to ensure that their rates depend only on the current state. When this is the case, stochastic processes exist that can be used to obtain analytic solutions for a wide class of variables characterizing both activity and marking related behavior. If this is not the case, simulation can be used to evaluate system behavior.

In order to be effectively applied to realistic systems, model construction and solution techniques require machine implementation. Both the complexity of the construction procedures and the typical sizes of resulting base models make this a necessity. To fill this need an extensive software package, called METASAN[1] [26], has been developed specifically for the construction and solution of SAN-based performability models.

METASAN, developed at the Industrial Technology Institute, was written using UNIX tools (C, Yacc, Lex, and Csh) and contains some 37,000 lines of source code. Models consist of two parts: a description of the structure of the net, and a description of the desired performance variables and solution method to be used in the evaluation process. Solution options include analytical techniques as well as both terminating and steady-state simulation.

III Measure Specification

As stated in the introduction, a reward model consists of a stochastic process and a reward structure. The stochastic process represents the dynamics of the system and can be constructed by hand or, automatically, from some network level description. The reward structure is typically a set of one or more functions defined on the states or transitions between states in the process. In all cases known to the authors, the interpretation given to each function is either that it is a *rate* at which reward is accumulated or that it is an *impulse* of reward that is obtained at the time of some "event" of the process. These events are typically either entrances to states, exits from states, or transitions between pairs of states. If the interpretation is of the first type we say that the reward is *rate-based*; reward functions with the second interpretation are said to be *impulse-based*. Performance variables can then be written in terms of the reward structure.

[1]METASAN is a registered Trademark of the Industrial Technology Institute.

As with the reward structure itself, the manner in which this is done varies greatly in the literature. Variables can be written in terms of the state of the process at a particular time, during an interval of time, or during a time-averaged interval of time. In the first case, the variable typically represents the "status" of the modeled system at some time t and is said to be an *instant-of-time* variable. In the second case, the variable typically represents accumulated benefit derived from operating the system for some interval of time and is said to be an *interval-of-time* variable. If the reward accumulated during some interval is divided by the length of the interval, one obtains a variable which represents the (time-averaged) rate at which reward is accumulated during the interval. Variables of this type are called *time-averaged interval-of-time* variables.

An excellent early exposition of a general reward structure and variable class is given by Howard [10]. In [10], Howard postulates a reward structure on semi-Markov processes that consists of both "yield rates" and "bonuses". In the terminology introduced above, the "yield rates" specify rates at which reward is accumulated and the "bonuses" specify impulses of reward that are obtained at state changes. More precisely, *yield rates* are associated with pairs of states, the interpretation being that for a pair of states i and j, $y_{i,j}(\alpha)$ is rate at which reward is accumulated in state i α time units after i was entered when the successor state is j. Furthermore, *bonuses* are associated with state transitions, where $b_{i,j}(\tau)$ is the reward awarded upon exit from i and subsequent entry into j given that the holding time in i was τ time units. The bonuses paid at state transitions depend both on the transition made and the holding time in the state preceding the transition. The generality of this structure is difficult to fully exploit, due to the complexity of the resulting solution. The analysis required is simplified if one considers reward rates that are constant during the occupancy of each state and bonuses that do not depend on the holding time in the previously occupied state. In this case,

$$y_{i,j}(\alpha) = y_{i,j} \text{ and } b_{i,j}(\tau) = b_{i,j}.$$

Howard then considers the solution for the expected value of an interval-of-time variable written in terms of reward structures of this type.

Further work focused on developing solution methods for reward models and did not make use of reward structure types that were as general as those considered by Howard. In particular, most researchers have limited their attention to a reward structure type with a single function that is rate-based. For acyclic systems, two general approaches have

emerged. The first is a time-domain approach. Examples of work that take this approach include Meyer [19] (rate-based time-averaged interval-of-time variable, specific two-processor system), Furchtgott and Meyer [6] (rate-based interval-of-time variable, acyclic nonrecoverable [32] system), and Goyal and Tantawi [9] (rate-based interval-of-time-variable, acyclic nonrecoverable system). The second approach is to use transform techniques. For example, see Donatiello and Iyer [4] (rate-based interval-of-time variable, acyclic system), Iyer *et al.* [11] (rate-based interval-of-time variable, acyclic system), and Ciciani and Grassi [3] (rate-based interval-of-time variable, acyclic system).

Later work considered systems that were cyclic, as well as more general reward variables. Notable here is the work of Trivedi *et al.* [31] (rate-based instant-of-time and interval-of-time variable, cyclic and acyclic system), Smith *et al.* [29] (rate-based instant-of-time, interval-of-time, and time-averaged interval-of-time variable, cyclic and acyclic system), and de Souza e Silva and Gail [30] (rate and impulse based interval-of-time variable, cyclic and acyclic system).

While each of these efforts extended known solution techniques for reward models, they did little to extend the generality of reward structure types and hence performance variables that could be considered. Except for the work by Howard and de Souza e Silva, little use has been made of impulse rewards. In addition, the utility of these methods has been limited by having all rewards assigned at the state level. While reasonable for state spaces that are small or have a high degree of regularity, it is often difficult to assign meaningful rewards to large numbers of states. We address both these issues by 1) constructing general reward structures at the network level and 2) systematically generating variables from these reward structure types.

The variables that we consider are systematically organized according to reward structure type, category within a reward structure type, and variable type within a category. The manner in which we do this is outlined in Figure 1. As depicted in this figure, categories of variables are distinguished at the highest level by the choice of a reward structure type. By type we mean one or more classes of functions that have a particular interpretation in terms of the networks. For a given reward structure type, variables can be further distinguished by the interval of time that they depend on. Three categories of variables are distinguished at this level, as was discussed earlier in this chapter. The first category, instant-of-time variables, represents the status of the SAN at either a particular time t or in steady state, as shown in Figure 1. Interval-of-time and time-averaged interval-of-time variables will also

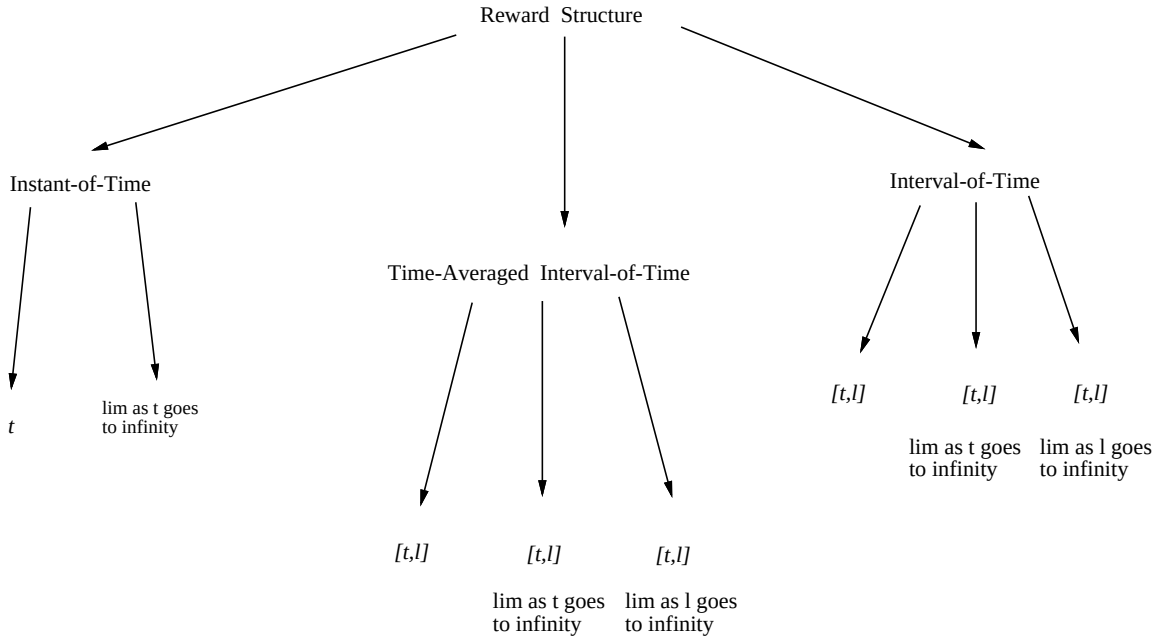


Figure 1: Types of Variable Considered

be considered.

Within each of the other two categories, the interval-of-time variables and time-averaged interval-of-time variables, three types of variables are considered. The first type represents the total or time-averaged reward (relative to a particular reward structure) accumulated during some interval $[t, t+l]$. The second type corresponds to an interval of length l as t goes to infinity, and is useful in representing the reward that is accumulated during some interval of finite length in steady-state. The final variable type corresponds to the total or time-averaged reward accumulated during an interval starting at t and of length l as $l \to \infty$. Thus, as can be seen in Figure 1, we consider eight variable types for each reward structure type.

We now consider a reward structure type that quantifies benefits associated with activity completions and particular numbers of tokens in places. By associating impulse rewards with activity completions, as well as reward rates with particular numbers of tokens in places, we greatly extend the measures of performability that can be considered. Variables based on a reward structure of this type can be used to determine many traditional and non-traditional measures of performance, including queueing time, queue length, processor

utilization, steady-state and interval availability, reliability, and productivity. In addition, if some high-level measure of "worth" is defined, this can be expressed as a particular reward structure of this type.

A Structure and Variable Definitions

We define an "activity-marking oriented reward structure" as follows:

Definition 1 *An* activity-marking oriented reward structure *of a stochastic activity network with places P and activities A is a pair of functions:*

$\mathcal{C} : A \to \mathbb{R}$ *where for $a \in A$, $\mathcal{C}(a)$ is the reward obtained due to completion of activity a, and*

$\mathcal{R} : \mathcal{P}(P, \mathbb{N}) \to \mathbb{R}$ *where for $\nu \in \mathcal{P}(P, \mathbb{N})$, $\mathcal{R}(\nu)$ is the rate of reward obtained when for each $(p, n) \in \nu$, there are n tokens in place p,*

where $\mathbb{N}$ is the set of natural numbers and $\mathcal{P}(P, \mathbb{N})$ is the set of all partial functions between P and $\mathbb{N}$.

Informally, impulse rewards are associated with activity completions (via $\mathcal{C}$) and rates of reward are associated with numbers of tokens in sets of places (via $\mathcal{R}$). An element $\nu \in \mathcal{P}(P, \mathbb{N})$ is referred to as a *partial marking*. The marking is partial in the sense that natural numbers are assigned some subset of P, namely the domain of the partial function ν. This assignment is made in a manner identical to the way a (total) marking assigns natural numbers to all the places in the set P. Although $\mathcal{R}$ has a countably infinite domain, the number of elements ν that are of interest to the modeler and, hence, deserving of a non-zero reward assignment will generally be small compared to, say, the number of reachable stable markings of the SAN. Similarly, it will usually be the case that only a fraction of the SAN's activities will have non-zero rewards associated with their completions. We thus use the convention, in practice, that rewards associated with activity completions and partial markings are are taken to be zero if not explicitly assigned otherwise.

Given a SAN with a reward structure of this kind, there are a variety of ways of defining different types of performance (reward) variables, as suggested in the previous section. In particular, we consider two variable types in the instant of time category. The first of these quantifies the behavior of a stochastic activity network at a particular time t. More

precisely, if we let V_t denote this variable type then

$$V_t = \sum_{\nu \in \mathcal{P}(P, \mathbb{N})} \mathcal{R}(\nu) \cdot I_t^{\nu} \; + \; \sum_{a \in A} \mathcal{C}(a) \cdot I_t^{a},$$

where

I_t^{ν} is an indicator random variable representing the event that the SAN is in a marking such that for each $(p, n) \in \nu$, there are n tokens in p at time t, and

I_t^{a} is an indicator random variable representing the event that activity a is the activity that completed most recently at time t.

This variable expresses the total reward (according to the reward structure defined above) associated with a SAN's status at an instant of time t. Depending on the instantiation of the reward structure, the variable can represent a variety of things including queue length and component status (e.g. idle, busy, blocked, failed, functioning). In view of our above observations concerning typical reward structures, and since zero values of $\mathcal{R}(\nu)$ can be ignored in the summation, the number of elements which must be accounted for in this sum is again relatively small.

Depending on the nature of the stochastic activity network in question, I_t^{ν} and I_t^{a} may converge in distribution for all ν and a with non-zero rewards as t approaches ∞. When this happens, the "steady-state" reward obtained at an instant of time can be studied. If we denote the random variable with this steady-state distribution as $V_{t \to \infty}$, its value can be expressed as

$$V_{t \to \infty} = \sum_{\nu \in \mathcal{P}(P, \mathbb{N})} \mathcal{R}(\nu) \cdot I_{t \to \infty}^{\nu} \; + \; \sum_{a \in A} \mathcal{C}(a) \cdot I_{t \to \infty}^{a},$$

where

$I_{t \to \infty}^{\nu}$ is an indicator random variable representing the event that the SAN is in a marking such that for each $(p, n) \in \nu$, there are n tokens in p in steady-state, and

$I_{t \to \infty}^{a}$ is an indicator random variable representing the event that activity a is the activity that completed most recently in steady-state.

Variables of the interval and time-averaged-interval categories can also be considered. In these cases, the reward accumulated is related both to the number of times each activity

completes and time spent in particular markings during an interval. As was discussed in the previous section, we consider three variable types in each of these categories corresponding to an interval of length l starting at time t ($[t, t+l]$), an interval of length l as $t \to \infty$ ($[t, t+l], t \to \infty$), and an interval starting at t as $l \to \infty$ ($[t, t+l], l \to \infty$). In the following, variable types of the interval category are denoted by "Y" while variables types of the time-averaged category are denoted by "W", each with the appropriate subscript. In particular, let

$$Y_{[t,t+l]} = \sum_{\nu \in \mathcal{P}(P, \mathbb{N})} \mathcal{R}(\nu) \cdot J^{\nu}_{[t,t+l]} \; + \; \sum_{a \in A} \mathcal{C}(a) \cdot N^{a}_{[t,t+l]}, \text{ and}$$

$$W_{[t,t+l]} = \frac{Y_{[t,t+l]}}{l}$$

where

$J^{\nu}_{[t,t+l]}$ is a random variable representing the total time that the SAN is in a marking such that for each $(p, n) \in \nu$, there are n tokens in p during $[t, t+l]$, and

$N^{a}_{[t,t+l]}$ is a random variable representing the number of completions of activity a during $[t, t+l]$.

If $J^{\nu}_{[t,t+l]}$ and $N^{a}_{[t,t+l]}$ converge in distribution as $t \to \infty$ for all ν and a that have non-zero reward assignments, the time-averaged reward accumulated and total reward accumulated during some interval of length l in steady-state can be studied. If we denote the random variables with these steady-state distribution as $Y_{[t,t+l],t\to\infty}$ and $W_{[t,t+l],t\to\infty}$ then

$$Y_{[t,t+l],t\to\infty} = \sum_{\nu \in \mathcal{P}(P, \mathbb{N})} \mathcal{R}(\nu) \cdot J^{\nu}_{[t,t+l],t\to\infty} \; + \; \sum_{a \in A} \mathcal{C}(a) \cdot N^{a}_{[t,t+l],t\to\infty}, \text{ and}$$

$$W_{[t,t+l],t\to\infty} = \frac{Y_{[t,t+l],t\to\infty}}{l},$$

where

$J^{\nu}_{[t,t+l],t\to\infty}$ is a random variable representing the total time that the SAN is in a marking such that for each $(p, n) \in \nu$, there are n tokens in p during a interval of length l in steady-state, and

$N^{a}_{[t,t+l],t\to\infty}$ is a random variable representing the number of completions of activity a during an interval of length l in steady-state.

Similarly, if $J^{\nu}_{[t,t+l]}$ and $N^{a}_{[t,t+l]}$ converge in distribution as $l \to \infty$ for all ν and a that have non-zero reward assignments, the total reward and time-averaged reward accumulated during an infinite interval starting at time t can be expressed as

$$Y_{[t,t+l],l\to\infty} = \sum_{\nu \in \mathcal{P}(P,\mathbb{N})} \mathcal{R}(\nu) \cdot J^{\nu}_{[t,t+l],l\to\infty} \; + \; \sum_{a \in A} \mathcal{C}(a) \cdot N^{a}_{[t,t+l],l\to\infty}, \text{ and}$$

$$W_{[t,t+l],l\to\infty} = \lim_{l\to\infty} \frac{Y_{[t,t+l]}}{l}$$

where

$J^{\nu}_{[t,t+l],l\to\infty}$ is a random variable representing the total time that the SAN is in a marking such that for each $(p,n) \in \nu$, there are n tokens in p during $[t,\infty)$, and

$N^{a}_{[t,t+l],l\to\infty}$ is a random variable representing the number of completions of activity a during $[t,\infty)$.

IV Example Variable Instantiations

Traditional measures of dependability and performance as well as more general performability measures can be specified easily using the performance variables just discussed and particular instances of the activity-marking oriented reward structure. To illustrate this, we consider a simple multiprocessor system where all processors service tasks from a single degradable buffer. The normal, fault-free operation of the system is as follows. Tasks arrive as a Poisson process with rate α. If the buffer is full, they are rejected. If not, they are placed in the buffer to be served by the first available processor in a FIFO manner. In addition, processing times are independent and exponentially distributed with each processor having a processing rate β.

Faults can occur both due to a failure of a buffer stage and due to a failure of a processor. In each case, the fault may be covered (i.e. the system degrades successfully to a less productive structure state) or it may result in a total loss of processing capability (i.e. total system failure). Additionally, certain processor failures are repairable. Repairs are performed on one processor at a time, with an exponentially distributed repair time with rate ζ. We assume further that faults in both a buffer stage and a processor occur as Poisson processes with rates λ and γ, respectively.

A stochastic activity network representing changes in the structure of the multiprocessor due to faults is given in Figure 2. Since our intent is to illustrate the specification of traditional dependability and performance variables, the model is kept simple. System resources (i.e. processors and buffers) are represented by tokens in places. Place A represents the number of processors queued for repair, place B represents the number of fault-free processors, and place C represents the number of fault-free buffer stages. Activities *processor_failure* and *buffer_failure* represent the occurrence of faults in the processors and buffer stages, respectively. Three types of processor faults are possible, corresponding to the three cases associated with activity *processor_failure*. Case 1 represents the occurrence of a fault that is repairable. Case 2 represents a total system failure, and case 3 represents the occurrence of a non-repairable fault. Cases for *buffer_failure* are similar, except that buffer stages may not be repaired. Here case 1 represents the occurrence of a non-repairable fault and case 2 represents total system failure. Processor repairs are represented by activity *processor_repair*.

Before dependability measures can be formulated, a definition of "system failure" must be given. In this regard, we say that the system has failed if all processors have failed in a manner such that they cannot be repaired. Then, if we define a reward structure such that

$$\mathcal{C}(a) = 0, \quad \forall a \in A$$

$$\mathcal{R}(\nu) = \begin{cases} 0 & \text{if } \nu = \{(A,0),(B,0)\} \\ 1 & \text{otherwise,} \end{cases}$$

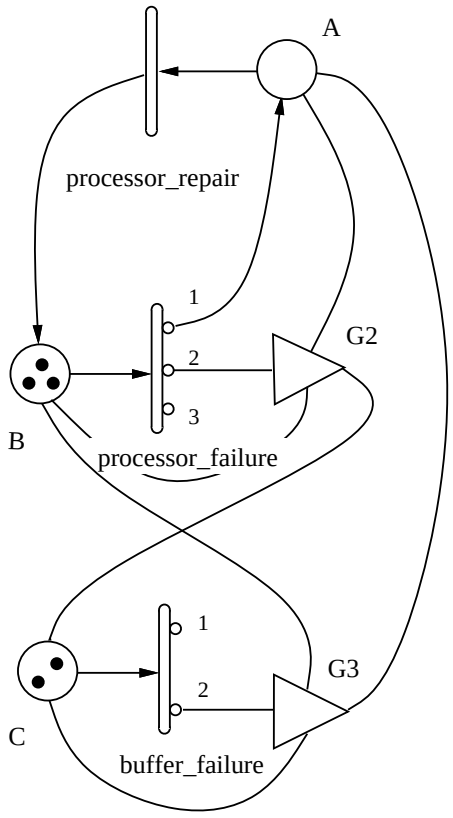
$E[V_t]$ is the reliability (using the above definition of system failure) at time t.

Measures of availability [7, 8] can be represented just as easily using this reward structure type and an associated variable. If we consider the system to be available whenever there is at least one processor functioning, the reward structure

$$\mathcal{C}(a) = 0, \quad \forall a \in A$$

$$\mathcal{R}(\nu) = \begin{cases} 0 & \text{if } \nu = \{(B,0)\} \\ 1 & \text{otherwise,} \end{cases}$$

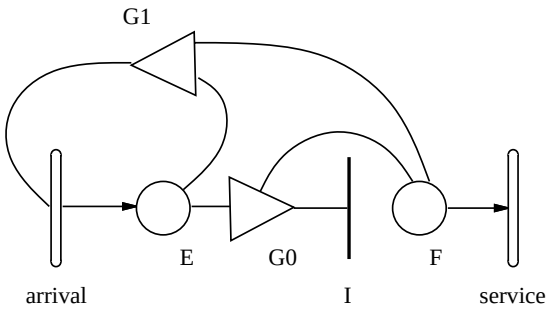
can be used to specify availability. For example, using this reward structure, the steady-state availability of the example multiprocessor system is $E[V_{t\to\infty}]$. The interval availability (i.e., the fraction of time the system is available during some interval of length l starting



Gate	Enabling Predicate	Function
G2	-	MARK(A)=MARK(B)=MARK(C)=0;
G3	-	MARK(A)=MARK(B)=MARK(C)=0;

Activity	Rate	Probability		
		case 1	case 2	case 3
processor_failure	$\gamma * MARK(B)$	cp1	cp2	cp3
buffer_failure	$\lambda * MARK(C)$	cb1	cb2	-
processor_repair	ζ	-	-	-

Figure 2: Multiprocessor Fault Model



Gate	Enabling Predicate	Function
G0	MARK(F) < N and MARK(E) > 0	MARK(E) = MARK(E) - 1; MARK(F) = MARK(F) + 1;
G1	MARK(E)+MARK(F) < M+N	identity

Activity	Rate
arrival	α
service	$\beta * MARK(F)$

Figure 3: Multiprocessor Performance Model

at time t) is $E[W_{[t,l]}]$ using the same reward structure. The distribution of availability, $F(t, l, x)$, is the probability that $W_{[t,l]} \leq x$.

Performance-oriented measures can be specified using a stochastic activity network model that represents task arrivals and completions. A stochastic activity network model of a multiprocessor with N processors and M buffers is given in Figure 3. In this figure, each completion of activity *arrival* represents the arrival of a task to the buffer. The buffer is represented by place E. The marking of place E represents the number of tasks queued for service. Place F represents the status of each of the processors, where the number of tokens in F is the number of processors that are busy. The finiteness of the buffer is represented by gate $G1$. Gate $G1$ specifies (via its predicate) that activity *arrival* is enabled only when the number of tasks in the system is less than system capacity (i.e. the sum of the markings of E and F is less than the sum of the number of working processors and buffers).

The service of tasks is represented by activity *service*. Use of a marking dependent activity completion rate for *service* allows us to represent all processors via a single activity, due to the memoryless property of the exponential distribution. The rate for activity *service*

is therefore the number of busy processors multiplied by the rate of a single processor. The number of busy processors is represented by place F.

If we define the throughput of the system during some interval $[t, t+l]$ as the number of tasks that are processed during the interval divided by the length of the interval, the throughput of the example system can be represented using a reward structure consisting only of impulse rewards. Specifically, consider the reward structure

$$\mathcal{C}(a) = \begin{cases} 1 & \text{if } a = service \\ 0 & \text{otherwise} \end{cases}$$

$$\mathcal{R}(\nu) = 0, \quad \forall \nu \in \mathcal{P}(P, I\!N).$$

Using this reward structure, the throughput is represented by the variable $W_{[t,t+l]}$. Steady-state throughput is given by the limit of this variable when $t = 0$ and $l \to \infty$, i.e. $W_{[0,l]}, l \to \infty$.

An alternate representation of expected steady-state throughput can be formulated based on the arrival rate to the system and the probability that an incoming task is processed. Since tasks arrive as a Poisson process, the probability that an incoming task is processed is one minus the probability the buffer is full. This probability can be captured by the reward structure,

$$\mathcal{C}(a) = 0, \quad \forall a \in A$$

$$\mathcal{R}(\nu) = \begin{cases} 1 & \text{if } \nu = \{(E, M)\} \\ 0 & \text{otherwise,} \end{cases}$$

and variable $E[V_{t\to\infty}]$. The expected steady-state throughput can then be written as

$$(1 - E[V_{t\to\infty}]) \times \alpha,$$

where α is the rate of arrival of tasks to the system.

A representation of expected steady-state response time can be obtained using Little's result [15] and the expected number of tasks in the system in steady-state. The expected number of tasks in the system can be represented using the reward structure

$$\mathcal{C}(a) = 0, \quad \forall a \in A$$

$$\mathcal{R}(\nu) = \begin{cases} i + j & \text{if } \nu = \{(E, i), (F, j)\} \\ 0 & \text{otherwise,} \end{cases}$$

if the variable is taken to be $E[V_{t\to\infty}]$. The expected steady-state response time is then $E[V_{t\to\infty}]$ divided by the rate at which tasks enter the system, i.e.

$$\frac{E[V_{t\to\infty}]}{\alpha}.$$

Processor utilizations can be obtained in a similar manner. Specifically, if the average processor utilization (in steady-state) is defined to be the fraction of the total number of processors that are busy, the reward structure

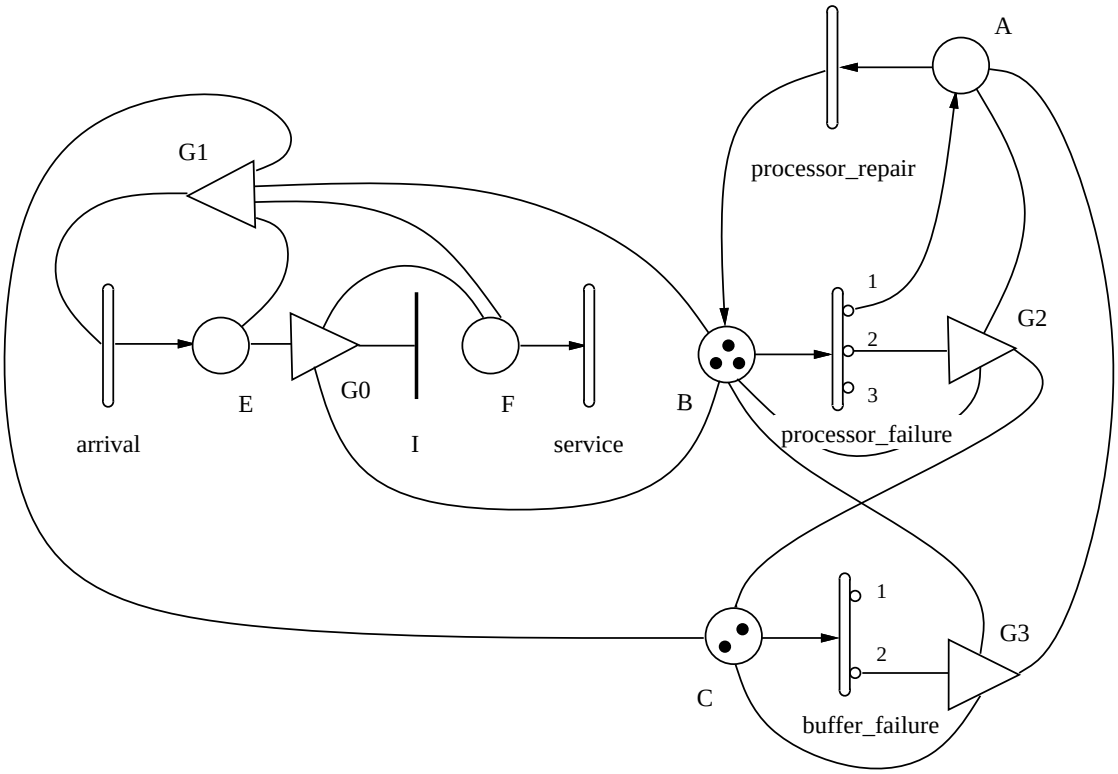
$$\mathcal{C}(a) = 0, \quad \forall a \in A$$

$$\mathcal{R}(\nu) = \begin{cases} i & \text{if } \nu = \{(F, i)\} \\ 0 & \text{otherwise,} \end{cases}$$

can be used. Processor utilization is then $\frac{V_{t\to\infty}}{N}$, where N is the number of processors in the system. As can be seen by the previous examples, traditional performance related and dependability related variables can be easily represented in the reward framework presented. Other performance and dependability related variables can be constructed in a similar manner. In addition, performability measures can be formulated as variables within this type if performance and fault type activities are represented in a single SAN model.

To illustrate the applicability of this method to the specification of performability variables, we consider the performability evaluation of the multiprocessor just used to illustrate traditional performance and dependability variables. The reward structure type and variables developed, together with the decomposition technique of [21, 25], allows us to consider "bottom-line" performability measures that summarize aspects of system performance caused by both fault and workload environments. To see this, we characterize the "total benefit" derived from operating the system for some interval $[t, t + l]$. We assume that "benefit" is derived from the completion of tasks and that costs are associated with the repair of processors. To make the discussion more concrete, we attach a benefit of x dollars to each task completion and a cost of y dollars to each processor repair.

Regarding solution, we construct a performability model which differentiates between "performance" and "structure" related submodels [21, 25]. These submodels are just the two SANs considered in the previous examples linked by two common places. A stochastic activity network representing the multiprocessor is given in Figure 4. Places B and C are the common places. Since task completions are represented in the performance submodel,



Gate	**Enabling Predicate**	**Function**
G0	MARK(F) < MARK(B) and MARK(E) > 0	MARK(E)=MARK(E)-1; MARK(F)=MARK(F)+1;
G1	MARK(E)+MARK(F) < MARK(B)+MARK(C)	identity
G2	-	MARK(A)=MARK(B)=0; MARK(C)=0;
G3	-	MARK(A)=MARK(B); MARK(C)=0;

Activity	Rate	Probability		
		case 1	case 2	case 3
processor_failure	$\gamma * MARK(B)$	cp1	cp2	cp3
buffer_failure	$\lambda * MARK(C)$	cb1	cb2	-
processor_repair	ζ	-	-	-
arrival	α	-	-	-
service	$\beta * MARK(F)$	-	-	-

Figure 4: Degradable Multiprocessor Model

the rate of task completions (i.e. throughput) in each structure state serves as the basis for the determination of the rate component of the reward structure. Specifically, the rate of benefit derived for a structure state is the throughput in that state multiplied by the dollar benefit associated with each task completion. Clearly, the throughput is just the arrival rate of tasks to the system multiplied by the probability that a task which arrives will be processed. In terms of the SAN model of the system, an incoming task will be rejected if the sum of the number of tokens in places E and F is equal to the sum of the number of tokens in places B and C (i.e., the system is full). Since tasks arrive as a Poisson process, the probability that an incoming task is processed is one minus the probability that the system is full. This fact allows us to define a reward structure for the performance submodel that permits the determination of system throughput for each structural configuration of the system. In this case, different structural configurations are distinguished by the number of functioning buffers and processors. Specifically, when the number of functioning buffers is m, the expected throughput can be obtained using a reward structure where

$$\mathcal{C}(a) = 0, \quad \forall a \in A$$

$$\mathcal{R}(\nu) = \begin{cases} 1 & \text{if } \nu = \{(E, m)\} \\ 0 & \text{otherwise,} \end{cases}$$

and taking the variable to be

$$Thru(m, n) = \alpha \times (1 - E_{(m,n)}[V_{t \to \infty}]),$$

where α is the rate of arrival of tasks to the system and $E_{(m,n)}$ is the expected value of the given variable when in there are m functioning buffers and n functioning processors. Costs associated with processor repairs are represented in the reward structure by associating a reward of $-y$ with each completion of activity *processor_repair*. Under these assumptions, the expected total benefit associated with operating the system for some utilization period $[0, t]$ can be found using the reward structure

$$\mathcal{C}(a) = \begin{cases} -y & \text{if } a = processor\_repair \\ 0 & \text{otherwise} \end{cases}$$

$$\mathcal{R}(\nu) = \begin{cases} x \cdot Thru(m, n) & \text{if } \nu = \{(B, n), (C, m)\} \\ 0 & \text{otherwise,} \end{cases}$$

and variable $E[Y_{[0,t]}]$.

Explicit values for this reward structure can now be obtained by solving the stochastic process associated with the performance submodel to obtain the throughput for each structural configuration. The results are not given here, since the intent was to illustrate the measure definition process, but can found in [25].

V Conclusions

As stated in the introduction, the objective of this work was to develop a unified method for specifying measures of performance, dependability, and performability. The framework proposed accomplishes this, we believe, and is particularly useful for critical systems where the measures used must capture what the user perceives as crucial to successful operation. In particular, the flexibility offered by first classifying measures based on their reward structure type and then based on categories within a particular type allows a user to specify precisely what is important in the evaluation process. The class of variables generated by the activity-marking oriented reward structure defined in Section 3 is rich in this regard, and permits specification of both traditional and non-traditional measures, as illustrated by the example variable instantiations given in Section 4.

The framework is also flexible in the sense that it allows room for definition of other reward structure types. In particular, while a large class of variables can be generated from instantiations of the activity-marking oriented reward structure type presented in this paper, these variables do not subsume all the variables that may be of interest to a user. Current work is directed toward defining additional reward structure types that quantify additional aspects of system behavior that may be of interest to a user.

Solution for the defined variables is also important issue, but beyond the scope of this paper. Depending on both the nature of the variable and model, the solution may be achieved by either simulation or analysis. We have investigated both of these solution approaches [26, 27], but have not yet implemented the methods in a software tool that allows one to directly specify variables using this framework. An effort to do this is currently underway, however, and should result in such a tool in the near future. This will allow us test these measure specification methods on larger and more realistic systems.

REFERENCES

[1] B.E. Aupperle, J.F. Meyer and L. Wei, "Evaluation of Fault-Tolerant Systems with Nonhomogeneous Workloads," in *Proc. 19th International Symp. on Fault-Tolerant Computing,* Chicago, IL, 1989.

[2] A. Avizienis and J. C. Laprie, "Dependable computing:From concepts to design diversity," *Proc. of the IEEE*, vol. 74, no. 5, pp. 629–638, May 1986.

[3] B Ciciani and V. Grassi, "Performability evaluation of fault-tolerant satellite systems", *IEEE Trans. on Communications*, vol. COM-35, no. 4, pp. 403–409, April 1987.

[4] L. Donatiello and B. R. Iyer, "Analysis of a composite performance reliability measure for fault-tolerant systems", *JACM*, vol. 34, no. 1, pp. 179–199, January 1987.

[5] D. Ferrari, *Computer Systems Performance Evaluation*, Prentice-Hall, Englewood Cliffs, NJ, 1978.

[6] D. G. Furchtgott and J. F. Meyer, "A performability solution method for degradable, nonrepairable systems", *IEEE Trans. on Computers*, vol. C-33, June 1984.

[7] A. Goyal and S. S. Lavenberg, "Modeling and analysis of computer system availability", *IBM Journal of Research and Development*, vol. 31, no. 6, pp. 651–664, November 1987.

[8] A. Goyal, S. S. Lavenberg, and K. S. Trivedi, "Probabilistic modeling of computer system availability", *Annals of Operations Research*, vol. 8, pp. 285–306, 1987.

[9] A. Goyal and A. N. Tantawi, "Evaluation of performability for degradable computer systems", *IEEE Trans. on Computers*, vol. C-36, no. 6, pp. 738–744, June 1987.

[10] R. A. Howard, *Dynamic Probabilistic Systems, Vol II: Semi-Markov and Decision Processes*, New York: Wiley, 1971.

[11] B. R. Iyer, L. Donatiello, and P. Heidelberger, "Analysis of performability for stochastic models of fault-tolerant systems", *IEEE Trans. on Computers*, vol. C-35, no. 10, pp. 902–907, October 1986.

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

[13] J. Laprie, "Dependable computing and fault tolerance: Concepts and terminology," in *Proc. 15th International Symp. on Fault-Tolerant Computing,* pp. 2–11, Ann Arbor, MI, June 1985.

[14] S. S. Lavenberg, *Computer Performance Modeling Handbook*, Academic Press, New York, NY, 1983.

[15] J. D. C. Little, "A Proof of the Queueing Formula $L = \lambda W$", *Operations Research*, vol. 9, pp. 383–387, 1961.

[16] R. Martinez, W.H. Sanders, Y. Alsafadi, J. Nam, T. Ozeki and K. Komatsu, "Performance evaluation of a picture archiving and communication system using stochastic activity networks", in *Proc. SPIE Medical Imaging IV*, Newport Beach, February 1990.

[17] J. F. Meyer, "On evaluating the performability of degradable computing systems," in *Proc. 1978 Int. Symp. on Fault-Tolerant Computing*, Toulouse, France, June 1978, pp. 44–49.

[18] J. F. Meyer, "On evaluating the performability of degradable computing systems," *IEEE Trans. Comput.*, vol. C-22, pp. 720–731, Aug. 1980.

[19] J. F. Meyer, "Closed-form solutions of performability", *IEEE Trans. on Computers*, vol. C-31, pp. 648–657, July 1982.

[20] J. F. Meyer, "Performability modeling of distributed real-time systems", in *Mathematical Computer Performance and Reliability*, Amsterdam: North-Holland, 1984.

[21] J. F. Meyer, A. Movaghar, and W. H. Sanders, "Stochastic activity networks: structure, behavior, and application," in *Proc. International Workshop on Timed Petri Nets*, Torino, Italy, July 1985, pp. 106–115.

[22] J.F Meyer, K.H. Muraldihar and W.H. Sanders, "Performability of a token bus network under transient fault conditions," in *Proc. 19th Int. Symp. on Fault-tolerant computing*, Chicago, June 1989.

[23] J.F. Meyer and L. Wei, "Influence of workload on error recovery in random access memeories," in *IEEE Transactions on Computers, Vol. 37, No. 4,* April 1988.

[24] A. Movaghar and J. F. Meyer, "Performability modeling with stochastic activity networks," in *Proc. 1984 Real-Time Systems Symp.*, Austin, TX, Dec. 1984.

[25] W. H. Sanders, "Construction and solution of performability models based on stochastic activity networks," Computing Research Laboratory Technical Report CRL-TR-9-88, The University of Michigan, Ann Arbor, MI, August 1988.

[26] W. H. Sanders and J. F. Meyer, "METASAN: a performability evaluation tool based on stochastic activity networks," in *Proc. ACM-IEEE Comp. Soc. 1986 Fall Joint Comp. Conf.*, Dallas, TX, Nov. 1986.

[27] W.H. Sanders and J.F. Meyer, "Reduced Base Model Construction Methods for Stochastic Activity Networks," in *Proc. Third International Workshop on Petri Nets and Performance Models*, Kyoto, Japan, Dec. 11-13, 1989.

[28] C. H. Sauer and K. M. Chandy, *Computer Systems Performance Modeling*, Prentice-Hall, Englewood Cliffs, NJ, 1981.

[29] R. M. Smith, K. S. Trivedi, and A. V. Ramesh, "Performability analysis: Measures, an algorithm, and a case study", *IEEE Trans. on Computers*, vol. C-37, no. 4, pp. 406–417, April 1988.

[30] E. de Souza e Silva and H. R. Gail "Calculating availability and performability measures of repairable computer systems using randomization", *JACM*, vol. 36, no. 1, pp. 171–193, January 1989.

[31] K. Trivedi, A. Reibman, and R. Smith, "Transient analysis of Markov and Markov reward models", in *Computer Performance and Reliability*, ed. G. Iazeolla, P.J. Courtois, and O.J. Boxma, North Holland, 1988.

[32] L.T. Wu, "Operational models for the evaluation of degradable computing systems", in *Proc. ACM/SIGMETRICS Conf. on Measurement and Modeling of Computer Systems*, pp. 179–185, Seattle, WA, August 1982.