The KAT (Knowledge-Action-Transformation) Approach to the Modeling and Evaluation of Reliability and Availability Growth

Jean-Claude Laprie, *Member, IEEE,* Karama Kanoun, Christian Béounes, and Mohamed Kaâniche

Abstract— **Reliability growth phenomena are not usually taken into account when performing dependability evaluations of systems during their operational life. However, such phenomena are significantly important. In this paper, an approach is presented which is aimed at the modeling and evaluation of reliability and availability of systems from the knowledge of the reliability growth of their components. Detailed models of reliability and availability for single-component systems are derived under much weaker assumptions than usually considered. These models, termed *knowledge* models enable phenomena to be precisely characterized, and a number of properties to be deduced. As they are, however, too complex in order to be directly applied in real life for performing predictions, it is necessary to derive simplified models for practical purposes, which can be termed as *action* models. Such an action model, the hyperexponential model, is presented; its properties are stated, and the model is applied to field data of software and hardware failures. An important property of the hyperexponential model is its interpretation as a Markov model, which enables classical Markov models (without relaibility growth) to be transformed into other Markov models which account for reliability growth. This *transformation* is applied to multicomponent systems. The hyperexponential model is comparable to other models as far as reliability of single-component systems is of concern; in addition, it enables estimating and predicting the reliability of multicomponent systems, as well as their availability. The transformation approach enables reliability growth phenomena to be accounted for, and this is performed in reusing the whole body of results available for Markov models. The results presented constitute a significant step toward the evaluation of reliability and availability of systems, with respect to both physical and design faults, as they enable 1) *reliability growth* phenomena to be incorporated in *hardware models*, and 2) *system structure* to be accounted for in *software models*. These results are thus a contribution to a solution to the (unfortunate) current separation between hardware models and software models.**

Index Terms—**Hardware, multicomponent systems, reliability and availability modeling and evaluation, software, stable reliability and reliability growth.**

I. Introduction

WHEN dealing with the assessment of dependability, the users of computing systems are interested in obtaining figures resulting from modeling and evaluation of systems, composed of hardware and software, with respect to both physical and design faults. Faced with these user requirements, hardware-and-software evaluations are far from being current practice, with a few exceptions (e.g., [53], [11], [5], [54], [51]). An explanation of this state of affairs lies in the fact that hardware and software evaluation have followed courses which can hardly be more separate [35].

Manuscript received July 25, 1990; revised November 24, 1990. Recommended by F. B. Bastani. This work was supported in part by ESPRIT Basic Research Actions, in the framework of the PDCS project (Predictably Dependable Computing Systems).

The authors are with LAAS-CNRS, 7 Avenue du Colonel Roche, 31400 Toulouse, France.

IEEE Log Number 9042427.

Hardware evaluation has concentrated on operational life, focusing on the influence on dependability of the system structure [57]; however, in spite of early work [16], it has largely been ignored that the reliability of hardware parts is significantly growing during the whole system's life, as shown, for instance, by the experimental data displayed in [8].

Software evaluation has mainly concentrated on the development-validation phase, focusing on the reliability growth of single-component ("black box") systems. Many models have been proposed (see surveys such as [60], [45]). Less attention has been paid to accounting for the structure of software systems, and has been restricted to the failure process, either for non-fault tolerant [40], [33] or for fault-tolerant software systems (e.g., [22], [33], [4]).

This paper elaborates on previous work [36]. It presents an approach aimed at filling the gaps identified above. The results we have obtained enable both reliability and availability of hardware and/or software systems to be evaluated, from the knowledge of the reliability growth of their components. Section II is devoted to models of reliability and availability for single-component systems which are derived under much weaker assumptions than usually considered. These models are termed *knowledge models*. As they are, however, too complex in order to be directly applied in real life for performing predictions, it is necessary to derive simplified models for practical purposes, which can be termed *action models*. Such an action model, the hyperexponential model, is introduced in Section III. The hyperexponential model can be seen as resulting from a transformation of classical Markov models in order to account for reliability growth phenomena. This transformation is then applied to multicomponent systems in Section IV. Appendix A gives the detailed derivations for the knowledge models, and Appendix B illustrates the relation between knowledge models and the hyperexponential model.

II. Characterization of System Behavior and Knowledge Models

The reliability of a system is conveniently illustrated by the failure intensity, as it is a measure of the frequency of the system failures as noticed by its user(s). Failure intensity is typically first decreasing (reliability growth), due to the removal of residual design faults, either in the software or in the hardware. It may become stable (stable reliability) after a certain period of operation; the failures due to internal faults occurring in this period are due to either physical faults, or to design faults which are admitted not to be removed. Failure intensity generally exhibits an increase (reliability decrease) upon the introduction of new versions incorporating modified functionalities; it then tends toward an asymptote again, and so on. It is noteworthy

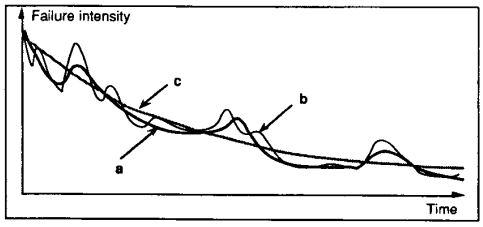


Fig. 1. Typical variations of a system's failure intensity.

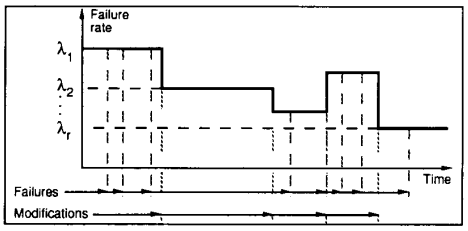


Fig. 2. Failure rate.

that such behavior is not restricted to the operational life of a system, but also applies to situations occurring during the development-validation phases of a system, e.g., 1) during incremental development [15], or 2) during system integration [37], [56].

Typical variations of the failure intensity may be represented as indicated on Fig. 1, curve **a**. Such a curve depends on the granularity of the observations, and may be felt as resulting from the smoothing of more noticeable variations (curve **b**); it may in turn be smoothed into a continuously decreasing curve (**c**). Although such a representation is very general and covers many practical situations, there are situations which exhibit discontinuities important enough that the smoothing process cannot be considered as reasonable (e.g., upon introduction of a new system generation).

We have above identified three classes of behavior: stable reliability, reliability growth, and reliability decrease. They may be defined as follows.

- *Stable Reliability:* The system's ability to deliver proper service is preserved (stochastic identity of the successive times to failure).
- *Reliability Growth:* The system's ability to deliver proper service is improved (stochastic increase of the successive times to failure).
- *Reliability Decrease:* The system's ability to deliver proper service is degraded (stochastic decrease of the successive times to failure).

Practical interpretations of stable reliability and of reliability growth are as follows.

- *Stable Reliability:* At a given restoration, the system is identical to what it was at the previous restoration; this corresponds to the following situations:
  - — in the case of a hardware failure, the failed part is changed for another one, identical and nonfailed;
  - — in the case of a software failure, the system is restarted with an input pattern different from the one having led to failure.
- *Reliability Growth:* The fault whose activation has led to failure is diagnosed as a design fault (in software or in hardware) and is removed.

Reliability decrease is theoretically, and practically, possible. In such a case, it has to be hoped that the decrease is limited in time, and that reliability is globally growing over a long observation period of time. Reliability decrease may originate from:

- introduction of new faults during corrective actions, whose probability of activation is greater than for the removed fault(s);
- introduction of a new version, with modified functionalities;
- change in the operating conditions (see, e.g., [26], where such situations are depicted);
- dependencies between faults: some software faults can be masked by others, i.e., they cannot be activated as long as the latter are not removed [50]; removal of the masking faults will have as a consequence an increase in the failure intensity.

In order to formalize the above mentioned properties, our approach [25], [34], [35], [28] has been to generalize the renewal theory [18] to the nonstationary case, i.e., when the successive times to failures are not stochastically identical. This generalization of the renewal theory leads to the knowledge model, which enables various operation resumption and maintenance policies to be considered:

- operation resumption after correction only;
- restart with off-line correction(s) performed either one-at-a-time or batch (after several failures have occurred);
- system modification without any failure occurrence since the last modification, i.e., preventive maintenance [1].

Detailed derivations for the knowledge model are given in Appendix A, and the main results are summarized below when the interfailure times constitute a piecewise Poisson process converging toward a Poisson process after r modifications have taken place; let $\{\lambda_1, \lambda_2, \ldots, \lambda_r\}$ be the sequence of these failure rates. Zero, one, or several failures can occur between two modifications, which can result from fault removal actions (Fig. 2); let a_j be the number of failures that have occurred since the $(j-1)$th system modification. It is assumed that system modifications are introduced a negligible time after the a_jth failure.

The failure intensity $h(t)$ has the following properties:

- $h(t)$ is a continuous function of time, with $h(0) = \lambda_1$ and $h(\infty) = \lambda_r$;
- when the a_j's are finite,
  - — a condition for $h(t)$ to be a nonincreasing function of time, i.e., a condition for *reliability growth*, is
    $$\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_j \geq \cdots \geq \lambda_r;$$
  - — the smaller the a_j's, the faster the reliability growth;
  - — if a (local) increase in the failure rates occurs, then the failure intensity correspondingly (locally) increases;
- when $a_1 = \infty$, no correction takes place and we are in the case of a classical renewal process; then $h(t) = \lambda_1$, which is the formulation of *stable reliability*.

These results are illustrated by Fig. 3 in the case of a reliability growth phase followed by a stable reliability phase: the failure intensity is plotted for $a_j = a \ \forall j$.

Since we have up to now considered reliability, the times to restoration have been assumed to be equal to zero. Considering

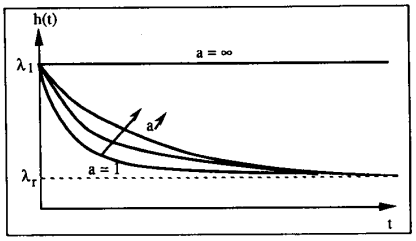


Fig. 3. Failure intensity.

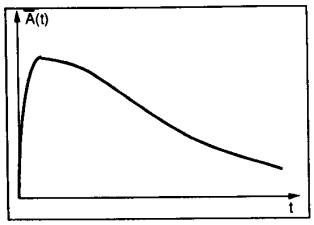


Fig. 4. Typical system unavailability.

nonzero times to restoration leads to availability, a measure of major interest for a wide variety of systems in operational use. Pointwise availability $A(t)$, denoted simply by availability in the following, is generally growing significantly during the whole operational life of a system (see, e.g., the field data displayed in [58], [51]). It was shown in [11] that the unavailability $\overline{A}(t) = 1 - A(t)$ has the shape displayed by Fig. 4: assuming a system initially working, unavailability rises sharply to a maximum, and then slowly decreases as the design faults are progressively removed.

The same approach of generalizing the renewal theory has been applied to availability in [28] in the case of operation resumption after correction only; the corresponding derivations, leading to a knowledge model for availability, are given in Appendix A. The results confirm and generalize what was previously established in [11], [33] through modeling via multistate Markov and semi-Markov chains. A summary is as follows in the case of a piecewise Poisson failure process and when the restoration rates are constant (see, e.g., [33] for a discussion about this assumption) and denoted by μ_j, $1 \leq j \leq r$:

P1) When reliability becomes stable, unavailability becomes constant: $\overline{A}(\infty) \approx \frac{\lambda_r}{\mu_r}$.

P2) If $\frac{\lambda_1}{\mu_1} \geq \frac{\lambda_r}{\mu_r}$ then there is an overshoot of unavailability in comparison to the asymptotic value $\frac{\lambda_r}{\mu_r}$.

P3) There is a single unavailability maximum (availability minimum) if reliability is growing, and if this growth is not nullified by possible increase in the restoration rates, that is if $\frac{\lambda_{n+1}}{\mu_{n+1}} \leq \frac{\lambda_n}{\mu_n}$, $n = 1, \cdots$; in the converse, local maxima (minima) occur.

P4) If the piecewise Poisson process of the interfailure occurrence times is continuously nonincreasing from λ_1 to λ_r, and if the times to failure are large with respect to the times to restoration, then $\overline{A}_{\max} \approx \frac{\lambda_1}{\mu_1}$.

P5) The time to reach the maximum unavailability (minimum availability) is of the order of magnitude of the mean time to restoration $\left(\frac{1}{\mu_1}\right)$.

P6) The changes in availability are significantly more influenced by the stochastic changes in the times to failure than in the times to restoration, which can be assumed as stochastically identical over the system's life.

The results relative to the knowledge models presented in this section are—although still suffering from some limitations such as the assumed independency between the times to failure necessary for performing the renewal theory derivations as discussed in Appendix A—more general than what was previously published in the literature, especially with respect to:

- their ability to consider reliability and availability measures for both hardware and software;
- the maintenance policies considered;
- their ability to cover stable reliability, reliability growth, and reliability decrease.

However, the results derived are too complex (see Appendix A) to be directly applied in real life for performing predictions, hence the need for simplified models for practical purposes, which can be termed *action models.*

III. An Action Model for Reliability and Availability Growth: The Hyperexponential Model

A. The Model and Its Properties

Our willingness to focus on the operational life rather than on the development-validation phase can be expressed through the following requirements:

R1) ability to model stable reliability occurring after reliability growth;

R2) derivation of both reliability and availability;

R3) applicability to hardware and/or software multicomponent systems.

None of the other models appearing previously in the literature, which can be considered as action models with respect to the knowledge model summarized in the previous section, satisfies this set of requirements. In particular, requirement R2 has not been previously paid attention. It is, however, worth mentioning that some models bring solutions for R1 or for R3, e.g.,

- requirement R1 is satisfied by the extension of Duane's model presented in [17]; the S-shaped model [50], [59] represents reliability decrease followed by reliability growth (although on the number of uncovered faults rather than on reliability per se), whereas the other models usually represent reliability growth only with a failure intensity tending asymptotically toward zero;
- with respect to requirement R3, Duane's model has been applied to single-component software in [30], multicomponent software has been examined in [50], [56], and Littlewood's model has been applied to single-component systems, either software or hardware, in [42].

In this section, we examine compliance to requirements R1 and R2. Requirement R3 will be dealt with in the next section.

We define the hyperexponential model[1][32], [28] as a nonhomogeneous Poisson process (NHPP) of failure intensity given by

$$h(t) = \frac{\omega\zeta_{\sup}e^{-\zeta_{\sup}t} + \overline{\omega}\zeta_{\inf}e^{-\zeta_{\inf}t}}{\omega e^{-\zeta_{\sup}t} + \overline{\omega}e^{-\zeta_{\inf}t}} \tag{1}$$

[1]The label "hyperexponential" covers a variety of models which are based on different rationales and/or interpretations (see, e.g., [50], [60]). They, however, share a common property, hence their name: they are issued from the Cox hyperexponential law.

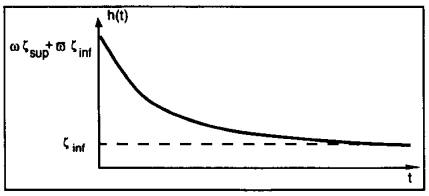


Fig. 5. Typical failure intensity for the hyperexponential model.

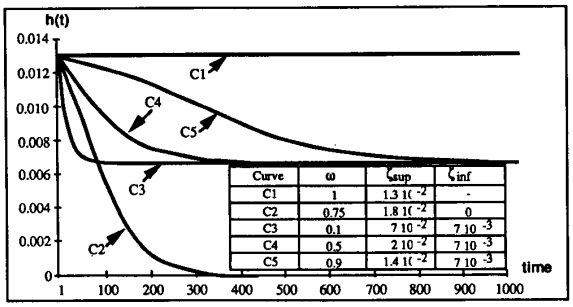


Fig. 6. Example failure intensity for the hyperexponential model.

with

$$0 \leq \omega \leq 1, \ \omega + \overline{\omega} = 1.$$

$h(t)$ is nonincreasing with time[2] when $0 < \omega < 1$, from $h(0) = \omega\zeta_{\text{sup}} + \overline{\omega}\zeta_{\text{inf}}$ to $h(\infty) = \zeta_{\text{inf}}$, as indicated in Fig. 5.

This model admits as special cases:

- the stable reliability situation, with exponential times to failures: 1) $\zeta_{\text{sup}} = \zeta_{\text{inf}}$, or 2) $\omega = 0$ or $\overline{\omega} = 0$;
- a failure intensity tending asymptotically toward zero: $\zeta_{\text{inf}} = 0$.

The rate of decrease of $h(t)$ can be adjusted via the values of the three parameters ζ_{sup}, ζ_{inf}, ω,, as illustrated by Fig. 6.[3]

An important property of NHPP's is as follows: let s_{i-1} denote the instant of occurrence of the $(i-1)$th failure, and τ denote the time elapsed since s_{i-1}. Then, we have the relation (see, e.g., [48]) $\lambda_i(\tau|s_{i-1}) = h(s_{i-1} + \tau)$: the failure rate for the ith

[2]The expressions of the first and second derivatives are as follows:

$$\frac{dh(t)}{dt} = -\frac{\omega\overline{\omega}(\zeta_{\text{sup}} - \zeta_{\text{inf}})^2 e^{-(\zeta_{\text{sup}}+\zeta_{\text{inf}})t}}{\left(\omega e^{-\zeta_{\text{sup}}t} + \overline{\omega}e^{-\zeta_{\text{inf}}t}\right)^2}$$

$$\frac{d^2h(t)}{dt^2} = -\frac{\omega\overline{\omega}(\zeta_{\text{sup}} - \zeta_{\text{inf}})^3 e^{-(\zeta_{\text{sup}}+\zeta_{\text{inf}})t}\left(\omega e^{-\zeta_{\text{sup}}t} - \overline{\omega}e^{-\zeta_{\text{inf}}t}\right)}{\left(\omega e^{-\zeta_{\text{sup}}t} + \overline{\omega}e^{-\zeta_{\text{inf}}t}\right)^3}.$$

These expressions show that the first derivative is always negative, whereas the second derivative may change sign according to the parameter values:

- if $\omega \leq \overline{\omega}$, the second derivative is always positive;
- if $\omega > \overline{\omega}$, the second derivative is first negative, then positive after $t = \frac{1}{\zeta_{\text{sup}} - \zeta_{\text{inf}}} \log \frac{\omega}{\overline{\omega}}$.

[3]A generalization of the model would be as follows:

$$h(t) = \left(\sum_{i=1}^{k} \omega_i \zeta_i e^{-\zeta_i t}\right) \Big/ \left(\sum_{i=1}^{k} \omega_i e^{-\zeta_i t}\right).$$

This generalization may enable better fitting to real situations at the expense of added complexity in parameter estimation.

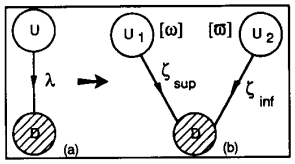


Fig. 7. Markov interpretation of the hyperexponential model.

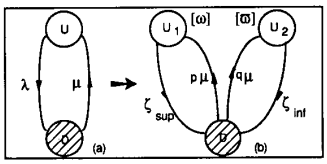


Fig. 8. Availability models.

failure cannot be distinguished from the failure intensity. We can thus, from a prediction viewpoint, consider our model as originating from the two-stage hyperexponential Cox law [14] (hence its name): it can be interpreted as a Markov model, where, instead of a single UP state, there are two such states, with initial probabilities ω and $\overline{\omega}$. The reliability growth model of a system can then be modeled from the transformation of a traditional Markov model (stable reliability, exponential failure process) into another Markov model, as indicated by Fig. 7.

When considering availability, the transformation approach just introduced (Fig. 7) leads when accounting for property P6 of Section II to the model of Fig. 8(b), from the classical Markov model of a single-component system [Fig. 8(a)]. Parameters p and q, $0 \leq p \leq 1, p + q = 1$, are determined hereafter.

The processing model of Fig. 8(b) leads, when accounting for the fact that $\mu \gg \zeta_{\text{sup}}$, to the following expression for the unavailability $\overline{A}(t)$:

$$\overline{A}(t) = \alpha + \beta \exp\left(-\left(q\zeta_{\text{sup}} + p\zeta_{\text{inf}} + o\left(\frac{\zeta_{\text{sup}}}{\mu}, \frac{\zeta_{\text{inf}}}{\mu}\right)\right)t\right) + \gamma \exp\left(-\left(\mu + o\left(\frac{\zeta_{\text{sup}}}{\mu}, \frac{\zeta_{\text{inf}}}{\mu}\right)\right)t\right)$$

with

$$\alpha = \frac{\zeta_{\text{sup}}\zeta_{\text{inf}}}{\mu(q\zeta_{\text{sup}} + p\zeta_{\text{inf}})} + o\left(\frac{\zeta_{\text{sup}}}{\mu}, \frac{\zeta_{\text{inf}}}{\mu}\right)$$

$$\beta = \frac{(\omega\zeta_{\text{sup}} + \overline{\omega}\zeta_{\text{inf}})(q\zeta_{\text{sup}} + p\zeta_{\text{inf}}) - \zeta_{\text{sup}}\zeta_{\text{inf}}}{\mu(q\zeta_{\text{sup}} + p\zeta_{\text{inf}})} + o\left(\frac{\zeta_{\text{sup}}}{\mu}, \frac{\zeta_{\text{inf}}}{\mu}\right)$$

$$\gamma = -\frac{\omega\zeta_{\text{sup}} + \overline{\omega}\zeta_{\text{inf}}}{\mu} + o\left(\frac{\zeta_{\text{sup}}}{\mu}, \frac{\zeta_{\text{inf}}}{\mu}\right).$$

$o(x)$ denotes quantities such that $\lim_{x \to 0} \frac{o(x)}{x} = 0$.

The maximum of unavailability is given by

$$\overline{A}_{\max} = \frac{\omega\zeta_{\text{sup}} + \overline{\omega}\zeta_{\text{inf}}}{\mu} + o\left(\frac{\zeta_{\text{sup}}}{\mu}, \frac{\zeta_{\text{inf}}}{\mu}\right).$$

This relation is in accordance with the results concerning the failure intensity, since $\omega\zeta_{\text{sup}} + \overline{\omega}\zeta_{\text{inf}} = h(0)$ which corresponds to the maximum of the intensity.

The asymptotic value $\overline{A}_\infty$ of $\overline{A}(t)$ is $\overline{A}_\infty = \alpha$. We wish $\overline{A}_\infty = h(\infty)/\mu = \zeta_{\text{inf}}/\mu$, which is obtained for $q = 1$ and $p = 0$ (another solution would be $\zeta_{\text{sup}} = \zeta_{\text{inf}}$, thus contrary to

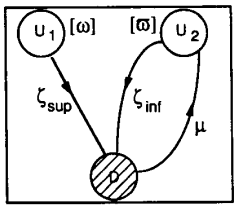


Fig. 9. Availability model for a single-component system.

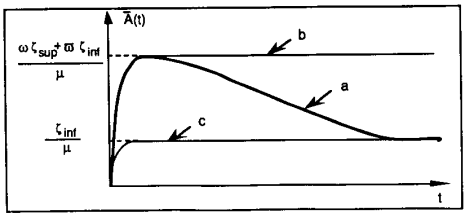


Fig. 10. Unavailability curves.

the assumption $\zeta_{sup} \neq \zeta_{inf}$ which is necessary for representing reliability growth).

The availability model for a single-component system with reliability growth is then given by Fig. 9.

Processing the model of Fig. 9 leads to the following expression for the unavailability $\overline{A}(t)$:

$$\overline{A}(t) = \frac{\zeta_{inf}}{\mu + \zeta_{inf}} + \frac{\omega(\zeta_{sup} - \zeta_{inf})}{\mu - (\zeta_{sup} - \zeta_{inf})} e^{-\zeta_{sup} t} - \frac{\mu(\omega\zeta_{sup} + \overline{\omega}\zeta_{inf}) - \overline{\omega}\zeta_{inf}(\zeta_{sup} - \zeta_{inf})}{(\mu + \zeta_{inf})(\mu - \zeta_{sup} + \zeta_{inf})} e^{-(\mu + \zeta_{inf})t}.$$

This expression becomes, when accounting for $\mu \gg \zeta_{sup}$,

$$\overline{A}(t) \approx \frac{\zeta_{inf}}{\mu} + \frac{\omega(\zeta_{sup} - \zeta_{inf})}{\mu} e^{-\zeta_{sup} t} - \frac{\omega\zeta_{sup} + \overline{\omega}\zeta_{inf}}{\mu} e^{-\mu t}. \quad (2)$$

The corresponding unavailability curve is given by Fig. 10, curve **a**; curves **b** and **c** are relative to stable reliability, either pessimistic before reliability growth took place (**b**) or optimistic after (**c**).

Interpretation of the model of Fig. 9 as representing a behavior such as reaching stationarity after having left the transient state U1 would be misleading, since such an interpretation does not account for the fact that initial probabilities of states U1 and U2 are both different from 0 or 1.

B. Application to Field Data

The utilization of the hyperexponential model to reliability prediction has been published elsewhere, and we shall restrict ourselves to summarizing the corresponding results. We shall then present an application of the model to availability.

1) Reliability: In [25], the hyperexponential model was applied to the data reported in [47] and compared to other models: the Jelinski–Moranda[23], the Duane [16], the Littlewood–Verral [39], and the Keiller–Littlewood models [30]. Comparison showed that it gives results comparable to the other models, in that sense that it is better than other models on some sets of data, whereas other models are better on other sets; this is a usual

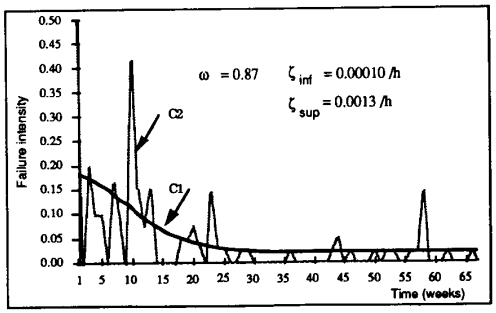


Fig. 11. Failure intensity of the TROPICO-R software.

situation when comparing several reliability growth models (see, e.g., [2]).

The system considered in [26] is a subsystem of the E-10 switching system from ALCATEL installed on an increasing number of sites, with more than 1400 sites at the end of the observation period. Data were collected over three years, including the end of the validation phase and a part of operational life. The software size is about 100 kilobytes. The ability of the hyperexponential model to predict the software behavior has been compared to the Jelinski–Moranda and the Keiller–Littlewood models, and it was revealed to give predictions as good as these two models. In addition, the ability of the hyperexponential model to cover stable reliability enables the asymptotic failure rate to be evaluated, and thus to perform an evaluation of the system (hardware and software).

The work reported in [24] addresses a telecommunication equipment. The size of the software is 1.4 Mlines; 2146 failure reports have been recorded during functional and integration testing. Application of the hyperexponential model allowed us to estimate the number of corrections to be performed during the next period of time.

Application of the hyperexponential model to real data is shown hereafter on data relative to the software of the TROPICO-R 4096 switching system,[4]developed by TELEBRAS, the Brazilian telecommunication company. The software is about 330 kilobytes. The data collection extended over the validation phase and the beginning of operational life. A total of 211 failure reports have been issued, over a period of 32 months; we have conducted the estimation of the failure intensity on the last 15 months of the observation period concerning operational life of the system, during which the number of systems in service went progressively from 4 to 42 and 50 failures have been reported. Fig. 11 gives the observed failure intensity (curve C2) and the estimated one (curve C1) using the hyperexponential model. The parameters $(\omega, \zeta_{sup}, \zeta_{inf})$ of the latter have been determined by a least-squares optimization. Comparison of the results given for this data set by the hyperexponential model to those obtained from the exponential model [19], the Littlewood–Verral model, and the S-shaped model presented in [59], has been carried out in [44]; it is shown that the results are equivalent for the hyperexponential, the exponential, and the Littlewood–Verral models, and are slightly better than the results obtained with the S-shaped model.

The hyperexponential model can be applied to hardware as well. Fig. 12 shows its application to field data displayed in [8],

[4]Application of reliability growth models to the previous version of this software (TROPICO-R 1500) is carried out in [29].

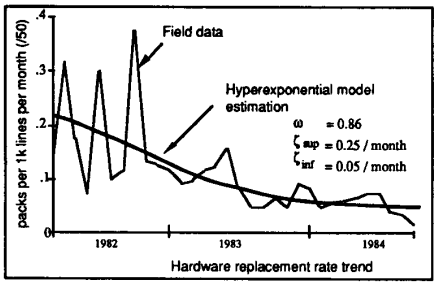


Fig. 12. Reliability growth estimation from [8].

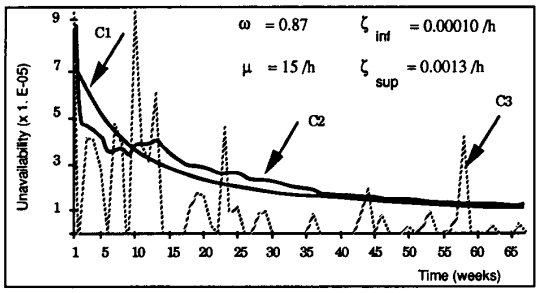


Fig. 13. Unavailability of the TROPICO-R software.

which are relative to a hardware subsystem of AT&T's 5ESS.

2) Availability: Statistical estimation of the availability for a set of systems is given by the ratio of nonfailed systems at time t to the total number of systems in the set. When the field data are relative to times to failure and to times to restoration, considering average availability instead of availability eases the estimation process, as average availability is the expected proportion of time a system is nonfailed [6]. The average availability over $[0,t]$ is defined by $A_{\text{av}}(t) = \frac{1}{t}\int_0^t A(\tau)\,d\tau$. Denoting, respectively, by UT_i and DT_i the observed times to failure and times to restoration, a statistical estimator of $A_{\text{av}}(t)$ is $\hat{A}_{\text{av}}(t) = \frac{1}{t}\sum_{i=1}^{n_t} UT_i$, where n_t is the number of failures which occurred in $[0,t]$.

The expression of the average unavailability for the hyperexponential model (Fig. 9) obtained from (2) is

$$\overline{A}_{\text{av}}(t) \approx \frac{\zeta_{\text{inf}}}{\mu} + \frac{\omega(\zeta_{\text{sup}} - \zeta_{\text{inf}})}{\mu}\,\frac{1 - e^{-\zeta_{\text{sup}}t}}{\zeta_{\text{sup}}t} - \frac{\omega\zeta_{\text{sup}} + \overline{\omega}\zeta_{\text{inf}}}{\mu}\,\frac{1 - e^{-\mu t}}{\mu t}. \tag{3}$$

The failure reports of the TROPICO-R switching system considered in the previous section do not give any information on the time to restoration subsequent to the reported failure. Since the corrections are performed off-line, restoration corresponds to system restart or software reloading. A data collection conducted on the times to restoration showed that they vary between 1 and 5 minutes. In order to complement the information given by the failure reports, we have performed a random sampling according to a uniform distribution with values between 1 and 5.

The results are displayed in Fig. 13, which gives the observed average unavailability (C2) and the average unavailability evaluated via the hyperexponential model (C1); the observed instantaneous unavailability is also given as an indication (C3). These results are evidently satisfactory.[5]

IV. The Transformation Approach to Modeling Reliability and Availability Growth of Multicomponent Systems

Owing to the Markov interpretation of the hyperexponential model, a natural approach to modeling multicomponent systems is to build a Markov model for the system from models of its components which incorporate reliability growth (Fig. 9). We introduce this approach through two deliberately simple examples. We then propose a general approach, based on stochastic Petri nets, and we apply it to a more realistic example. We finally give the interest and limits of the transformation approach.

[5]Criteria for reliability growth model validation (and comparison) are based either on the distribution of the considered random variable (e.g., the Kolmogorov–Smirnov distance or the prequential approach) or on the residue (which corresponds to the sum of the differences between the observed and the evaluated measure). Criteria of the first kind are not suited for availability, and the residue criterion is useful only when comparing predictions achieved by different models.

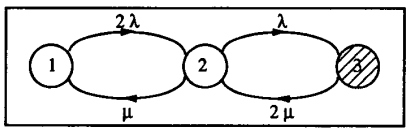


Fig. 14. Markov model of a two-component redundant system in stable reliability.

A. Two Simple Examples

Let us consider first a two-component hardware redundant system, with perfect coverage and unrestricted repair facilities. The classical stable reliability Markov model of such a system is given in Fig. 14.

As the two components are stochastically independent, the model of Fig. 14 can be formally derived from the transition matrices of its components through the Kronecker algebra[6] [3].

Let $\mathbf{\Lambda}_c = \begin{pmatrix} -\lambda & \lambda \\ \mu & -\mu \end{pmatrix}$ denote the transition matrix of each component. The transition matrix of the system is then $\mathbf{\Lambda}_s = \mathbf{\Lambda}_c \oplus \mathbf{\Lambda}_c$, where $\oplus$ denotes the Kronecker sum. The transition matrix corresponding to the model of Fig. 13 is derived from $\mathbf{\Lambda}_s$ by the Kemeny–Snell theorem for Markov chain reduction [31] since the two components are identical.

When reliability growth of the components is accounted for, the transition matrix $\mathbf{\Lambda}_c$ of each component becomes (Fig. 9):

$$\mathbf{\Lambda}_c = \begin{pmatrix} -\zeta_{\text{sup}} & 0 & \zeta_{\text{sup}} \\ 0 & -\zeta_{\text{inf}} & \zeta_{\text{inf}} \\ 0 & \mu & -\mu \end{pmatrix}.$$

[6]The Kronecker product of two matrices $\mathbf{A} = [a_{ij}]$, of dimensions (p,q), and $\mathbf{B} = [b_{ij}]$, of dimensions (m,n), is given by

$$\mathbf{A} \otimes \mathbf{B} = \begin{pmatrix} a_{11}\mathbf{B} & a_{12}\mathbf{B} & \cdots & a_{1q}\mathbf{B} \\ a_{21}\mathbf{B} & a_{22}\mathbf{B} & \cdots & a_{2q}\mathbf{B} \\ \vdots & \vdots & & \vdots \\ a_{p1}\mathbf{B} & a_{p2}\mathbf{B} & \cdots & a_{pq}\mathbf{B} \end{pmatrix}.$$

The Kronecker sum of two matrices $\mathbf{C} = [c_{ij}]$, of dimensions (n,n), and $\mathbf{D} = [d_{ij}]$, of dimensions (m,m), with $\mathbf{I}_x$ denoting the identity matrix of dimensions (x,x), is given by $\mathbf{C} \oplus \mathbf{D} = \mathbf{C} \otimes \mathbf{I}_m + \mathbf{I}_n \otimes \mathbf{D}$.

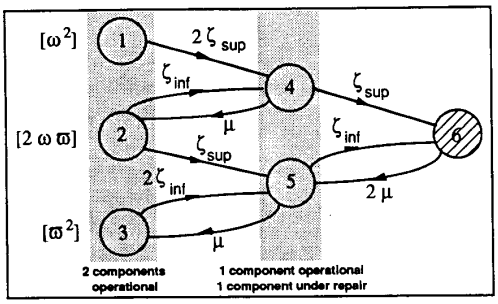


Fig. 15. Markov model of a two-component redundant system in reliability growth.

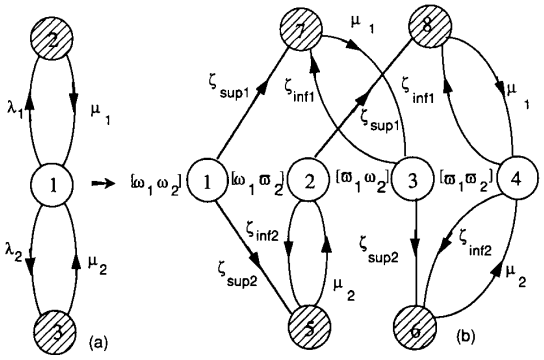


Fig. 16. The transformation approach for a two-component system.

Let $\mathbf{P}_c(0) = (\omega \ \ \overline{\omega} \ \ 0)$ be the initial probability vector of each component. The initial probability vector of the system is then given by: $\mathbf{P}_s(0) = \mathbf{P}_c(0) \otimes \mathbf{P}_c(0)$, where $\otimes$ denotes the Kronecker product.

The Markov chain so generated, reduced in the Kemeny–Snell sense, is given by Fig. 15.[7]

The approach obviously applies to the case when components are not identical. Let us consider for instance a two-component nonredundant system. Let λ_1, μ_1, and λ_2, μ_2, denote the failure and restoration rates of the components C1 and C2, respectively; the components can be either hardware or software. Going through the same steps as in the previous example (except the reduction step à la Kemeny–Snell as the two components are different here), leads to the models of Fig. 16, where (a) is the stable reliability model, and (b) is the reliability growth model.

Processing of the model of Fig. 16(b) leads to Fig. 17. On this figure, for the values selected for the various parameters, the unavailability of the system is conditioned by the unavailability of component C2 at the beginning and by the unavailability of component C1 later. This example shows that a seemingly smooth and orderly reliability growth of a system can mask

[7]This model is simple enough in order to derive analytical expressions. We get for instance for unavailability, accounting for $\mu \gg \zeta_{\text{sup}} > \zeta_{\text{inf}}$:

$$\overline{A}(t) = \frac{\zeta_{\text{inf}}^2}{\mu^2} + 2\frac{\omega\zeta_{\text{inf}}(\zeta_{\text{sup}} - \zeta_{\text{inf}})}{\mu^2} e^{-\zeta_{\text{sup}}t} + \frac{\omega^2(\zeta_{\text{sup}} - \zeta_{\text{inf}})^2}{\mu^2} e^{-2\zeta_{\text{sup}}t} - 2\frac{(\omega\zeta_{\text{sup}} + \overline{\omega}\zeta_{\text{inf}})^2}{\mu^2} e^{-\mu t}.$$

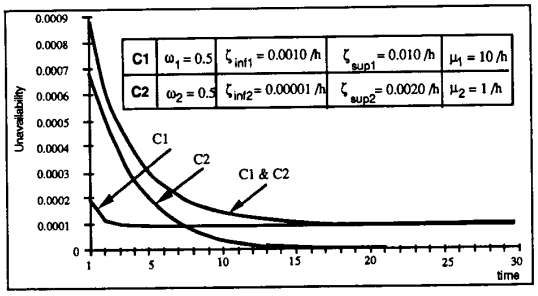


Fig. 17. Unavailability of the two-component system.

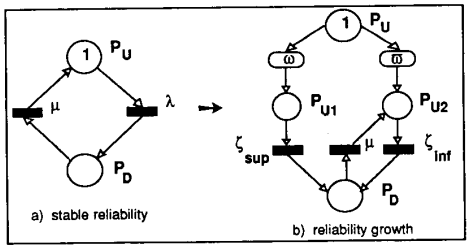


Fig. 18. GSPN models for single-component systems.

different situations for its components (see, e.g., [44] where such a situation from field data is depicted).

B. General Approach

In most realistic systems, there exist stochastic dependencies among components, which are conveniently modeled by stochastic Petri nets (SPN) [10], [46], [9] under the form of synchronization and cooperation among concurrent processes. The introduction of instantaneous transitions led to the generalized stochastic Petri nets (GSPN) [43].

Considering the hyperexponential model of a single-component introduced in Section III, the GSPN modeling approach leads to the models of Fig. 18 where the white ovals represent instantaneous transitions and the black rectangles represent timed transitions. The Markov models corresponding to the GSPN's of Fig. 18(a) and (b) are, respectively, the Markov models of Figs. 8(a) and (9). In the GSPN of Fig. 18(b), the instantaneous transitions enable the assignment of initial probabilities distinct from 1 or 0 to the initial states of the associated Markov chain to be modeled (this is in fact a relaxation of the implicit assumption in classical GSPN's about the equivalence of the initial marking and a probability equal to 1 assigned to the initial state of the associated Markov chain).

The transformation approach when using GSPN's then consists of the following steps:

S1) Construction of the GSPN of the system assuming stable reliability.
S2) Transformation of the GSPN according to Fig. 18.
S3) Derive the reachability graph of the transformed GSPN, which is the Markov chain accounting for reliability growth.

We illustrate this approach on a two-component system with imperfect coverage and single repair facility. Let:

- λ_c and $\lambda_{\overline{c}}$ denote the failure rates corresponding to covered

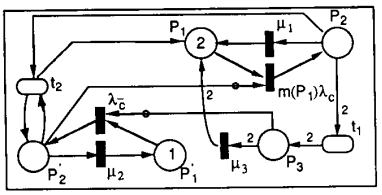


Fig. 19. GSPN of the system in stable reliability.

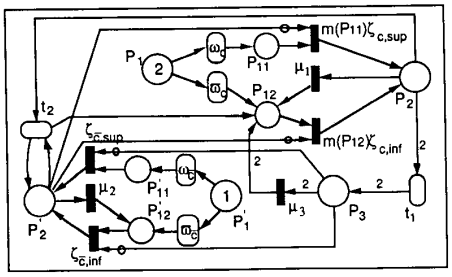


Fig. 20. GSPN of the system in reliability growth.

and noncovered errors, respectively; $\lambda_{\bar{c}}$ may be thought as being the failure rate of the error recovery software;

- μ_1, μ_2, and μ_3 denote the restoration rates after a) the first covered failure, b) a noncovered failure, and c) a second covered failure, respectively.

The stable reliability GSPN is given by Fig. 19. The transformed GSPN in order to account for reliability growth via the hyperexponential model is given by Fig. 20, where $\zeta_{c,\text{sup}}$ and $\zeta_{c,\text{inf}}$ result from the transformation of λ_c, and $\zeta_{\bar{c},\text{sup}}$ and $\zeta_{\bar{c},\text{inf}}$ result from the transformation of $\lambda_{\bar{c}}$.

The Markov chains corresponding to the GSPN's of Figs. 19 and 20 are given by Figs. 21 and 22, respectively.

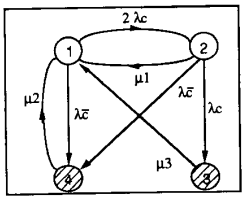
Processing the Markov chain of Fig. 22 by the SURF program [12] leads to the results plotted in Fig. 23, where:

- the covered failure rate and the noncovered failure rate after reliability growth ($\zeta_{c,\text{inf}}$ and $\zeta_{\bar{c},\text{inf}}$) have been taken as equal to 10^{-4}/h and $5 \cdot 10^{-7}$/h, respectively (thus $\zeta_{\bar{c},\text{inf}}/\zeta_{c,\text{inf}} = 5 \cdot 10^{-3}$);
- the extent of reliability growth for both sources of failures is equal to 10; thus, we have $10^{-4} \leq \lambda_c \leq 10^{-3}$ and $5 \cdot 10^{-7} \leq \lambda_{\bar{c}} \leq 5 \cdot 10^{-6}$;
- the restoration rates are such that: $\zeta_{c,\text{inf}}/\mu_1 = 5 \cdot 10^{-3}$; $\zeta_{c,\text{inf}}/\mu_2 = \zeta_{c,\text{inf}}/\mu_3 = 10^{-3}$ (restoration is considered easier when one of the components is still operating than when the system is totally down);
- curve C1 (respectively C5) corresponds to stable reliability conditions, with λ_c and $\lambda_{\bar{c}}$ taking their minimal values (respectively maximal values);
- C4 corresponds to reliability growth, with λ_c and $\lambda_{\bar{c}}$ decreasing from their maximal values to their minimal values, respectively, and $\omega_c = \omega_{\bar{c}} = 0.5$;
- curves C2 and C3 are displayed in order to show the influence of the reliability growth of the covered failure rate and of the noncovered failure rate taken in isolation, respectively.

C. Interest and Limitations of the Transformation Approach

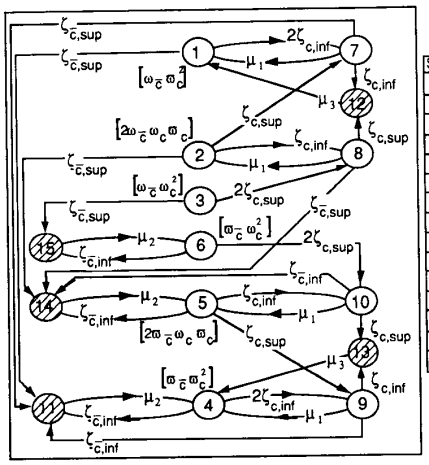
The transformation approach when applied to multicomponent systems satisfies requirement R3 of Section III-A: we can model the dependability growth of a system from the dependability growth of its components. In addition, this is performed in reutilizing all the body of results which have been derived for building, processing, and validating Markov models.

The main limitation of the approach clearly lies in the increase in the state space with respect to the stable reliability models. The cardinality of the state space for an n-component system in stable reliability with two-state components is comprised between $n+1$ (all components identical) and 2^n (all the components different), and it ranges from $(n+1)(n+2)/2$ to 3^n in the case of reliability growth. However, this limitation is not so drastic as it would have been a few years ago, thanks to the powerful current methods for processing Markov models [21], [55], [52], which have been reported to have the capability of processing models of several tens of thousand states [20]. These cardinalities assume stochastic



State	Marking
1	{2p1,p'1}
2	{p1,p2,p'1}
3	{2p3,p'1}
4	{2p1,p'2}

Fig. 21. Markov chain of the system in stable reliability.



State	Marking
1	{2P12,P'11}
2	{P11,P12,P'11}
3	{2P11,P'11}
4	{2P12,P'12}
5	{P11,P12,P'12}
6	{2P11,P'12}
7	{P12,P2,P'11}
8	{P11,P2,P'11}
9	{P12,P2,P'12}
10	{P11,P2,P'12}
11	{2P12,P'2}
12	{2P3,P'1}
13	{2P3,P'12}
14	{P11,P12,P'2}
15	{2P11,P'2}

Fig. 22. Markov chain of the system in reliability growth.

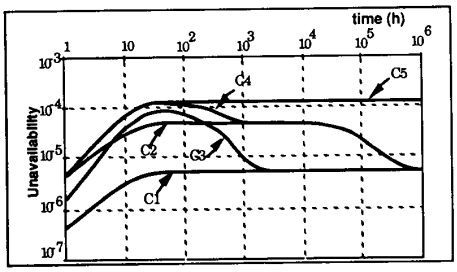


Fig. 23. Unavailability curves for the model of Fig. 22.

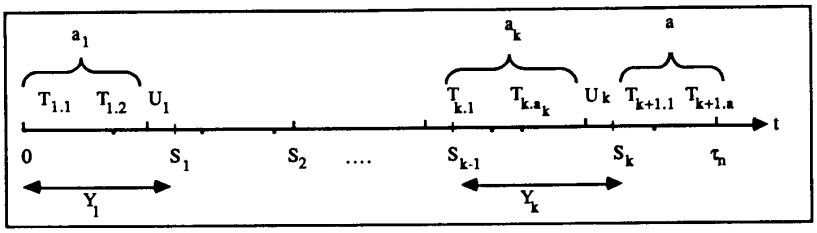


Fig. 24. Relationship between the different time intervals.

independence. However, stochastic dependencies generally have more impact on the interstate transitions than on the state space cardinality. It is noteworthy that the model of Fig. 22 is relative to a system whose components have more than two states, and are stochastically dependent.

V. Conclusion

The results which have been presented in this paper constitute a significant step toward the evaluation of reliability and availability of systems, with respect to both physical and design faults, as they enable 1) *reliability growth* phenomena to be incorporated in *hardware models*, and 2) *system structure* to be accounted for in *software models*. Our results are thus a contribution toward a solution to the (unfortunate) current separation between hardware models and software models.

Central to our results are the hyperexponential model and the transformation approach. The hyperexponential model is comparable to other models as far as reliability of single-component systems is of concern; in addition, it enables estimating and predicting the reliability of multicomponent systems, as well as their availability. The transformation approach enables reliability growth phenomena to be accounted for, and this is performed in reusing the whole body of results available for Markov models.

Incorporating reliability growth into dependability predictions performed for systems in operational use is indeed extremely worthwhile: the field data displayed in [58] show that unavailability is decreased by a factor of 250 during 3 1/2 years of operation. Stable reliability, as usually assumed in the dependability evaluations currently performed for systems in their operational life, is in fact a very special case.

Appendix A
Knowledge Models

A. Reliability Model

Let

- Y_j denote the time intervals between the $(j-1)$th and the jth system modification (a modification may be due to any of the maintenance forms: corrective, perfective, or adaptive);
- S_j denote the instant of introduction of the jth system modification;
- $T_{j,i}$ denote the time intervals between the $(i-1)$th and the ith failures since the $(j-1)$th system modification;
- a_j denote the number of failures that have occurred since the $(j-1)$th modification.

The times to restoration are assumed to be equal to zero.

The relationship between the $T_{j,i}$'s, and the Y_j's is determined by the policies of maintenance and of service restoration. A policy which accepts as special cases the policies generally considered in dependability evaluation is as follows: the jth system modification takes place after a_j failures have occurred since the $(j-1)$th system modification. We then have

$$Y_j = \sum_{i=1}^{a_j} T_{j,i},$$

$$S_j = \sum_{i=1}^{j} Y_i.$$

The number of failures between two modifications (a_j) depends on several factors, such as

- the failure rate of the software;
- the nature of the faults (e.g., time needed to diagnosis the fault, the consequence of the failure due to the activation of this fault);
- the considered phase in the life-cycle and may vary for a given system within the same phase.

Two extreme special cases of this general policy are as follows:

1) $a_j = 1$ and $U_j = 0\ \forall j$, thus $Y_j = T_{j,1}$: service is restored only after a system modification has been performed; this case relates to:
   - a usual hypothesis for software reliability (growth) models;
   - the case of critical systems, after a (potentially) dangerous failure occurrence.
2) $a_1 = \infty\ \forall j$, thus $Y_1 = \sum_{i=1}^{\infty} T_{1,i}$: service is restored without any system modification ever being performed; this case relates to:
   - hardware, when maintenance consists of replacing a failed part by an identical new one;
   - software, when no maintenance is performed, service restoration always corresponding to a restart.

Although this policy is more general than the usually considered policies (the special cases just above), it is noteworthy that it is, however, a simplification of real life, as it does not explicitly model the phenomena of 1) interweaving of failures and corrections [27], and 2) failure rediscoveries [1].

In order to derive the expression of the failure intensity for this general model, some more parameters have to be introduced:

τ_n time of occurrence of the nth failure;
k number of modifications before the nth failure;

a number of failures after the kth modification, before τ_n,

$$n = \sum_{j=1}^{k} a_j + a$$

$$\tau_n = S_k + \sum_{i=1}^{a} T_{k+1,i}.$$

The relationship between all the defined intervals is summarized in Fig. 24.

Let:

- $f_j(t)$ be the probability density function of $T_{j,i}, i = 1 \cdots a_j, j = 1 \cdots k+1$;
- $\phi_n(t)$ be the probability density function of τ_n;
- $h(t)$ be the renewal density (derivative of the expected number of failures in $[0,t]$), i.e., the failure intensity.

Performing derivations adapted from the renewal theory (see, e.g., [18]) are relatively straightforward, provided that the $T_{j,i}$'s and the U_j's are assumed stochastically independent. This assumption, although usual in both hardware and software models, is again a simplification of real life. The Y_j's can reasonably be considered as stochastically independent if resuming execution after introduction of a modification involves a so-called "cold restart"; it has, however, to be mentioned that imperfect maintenance, whose consequences were noticed a long time ago [38], is also a source of stochastic dependency. The stochastic independence of the $T_{j,i}$'s for a given j depends on 1) the extent to which the internal state of the system has been affected, and on 2) the nature of operations undertaken for execution resumption, i.e., whether they involve state cleaning or not. We get, under the stochastic independence assumption:

$$\phi_n(t) = f_1(t)^{*[a_1]} * f_2(t)^{*[a_2]} * \cdots * f_{k+1}(t)^{*\left[n-\sum_{j=1}^{k} a_j\right]}, \quad \text{(A1)}$$

and

$$h(t) = \sum_{n=1}^{\infty} \phi_n(t) \quad \text{(A2)}$$

where the symbol * denotes the convolution operation, and $f_k(t)^{*[a_k]}$ the a_k-fold convolution of $f_k(t)$ by itself.

The above derivations constitute a (simple) generalization of the renewal theory and of the notion of renewal process: in the classical theory, the $T_{j,i}$'s are identically distributed, i.e., $f_j(t) = f(t) \forall j$ (the case where $f_1(t)$ is different from the other $f_j(t), j > 2$, is termed the "modified renewal" process in [7], [13]).

Let us consider the case where the $T_{j,i}$'s are exponentially distributed (see [45], [35] for a discussion about this assumption): $f_j(t) = \lambda_j \exp(-\lambda_j t)$. The interfailure occurrence times constitute in such a case a piecewise Poisson process. No assumption is made here as to the sequence of magnitude of the λ_j's. It will however be assumed that the failure process is converging toward a Poisson process after r modifications have taken place. This assumption means that either 1) there are no more modifications performed or 2) if some modifications are still performed, they do not significantly affect the failure rate of the system. Such an assumption is usual for hardware; it is also an experimentally established fact for software in operational conditions [49].

Using (A1) and (A2) leads to the following expression $\tilde{h}(s)$ of the Laplace transform of the intensity function

$$\tilde{h}(s) = \sum_{i=1}^{r-1} \prod_{m=1}^{i-1} \left(\frac{\lambda_m}{\lambda_m + s}\right)^{a_m} \sum_{j=1}^{a_i} \left(\frac{\lambda_i}{\lambda_i + s}\right)^{j} + \frac{\lambda_r}{s} \prod_{m=1}^{r-1} \left(\frac{\lambda_m}{\lambda_m + s}\right)^{a_m} \quad \text{(A3)}$$

where

$$\prod_{j=a}^{b} x_j = 1 \quad \text{for} \quad b < a, \ \forall x_j.$$

Derivation of $h(t)$ is very tedious and the corresponding expression is very complex; however, assuming operation restart after correction only ($a_j = 1 \forall j$) leads to the following expression of the intensity function:

$$h(t) = \lambda_r + \sum_{i=1}^{r-1} \left(\left(\sum_{k=i}^{r-2} \lambda_i \prod_{j=1,\ j \neq i}^{k} \frac{\lambda_j}{\lambda_j - \lambda_i} \right) + (\lambda_i - \lambda_r) \prod_{j=1\ j \neq i}^{r-1} \frac{\lambda_j}{\lambda_j - \lambda_i} \right) \exp(-\lambda_i t) \quad \text{(A4)}$$

where

$$\prod_{j=a}^{b} x_j = 1 \quad \text{for} \quad b < a, \ \forall \ x_j.$$

Equation (A4) corresponds to the intensity function associated with the most simple knowledge model issued from the more general model given by (A3). It can be seen that even in this very simplified case the expression of intensity versus time is too complex to be used in real situations.

B. Availability Model

The same approach of generalizing the renewal theory can be applied to availability as in [28]. For sake of simplicity the availability model will be derived assuming operation restart after correction only. Let

- Z_j denote the restoration duration and $q_j(t)$ its pdf;
- T_j the time to failure (pdf: $f_j(t)$),
- Y_j the time between the $(j-1)$th and the jth modifications (pdf: $\phi_j(t)$).

We have: $Y_j = T_j + Z_j$.

In terms of probability, the availability $A(t)$ is defined as: $A(t) = P\{\text{the system delivers a proper service at time } t\}$.

Considering the events E_n such that at time S_n, n failures took place and at time t ($t > S_n$) the system delivers a proper service, leads to the expression of availability:

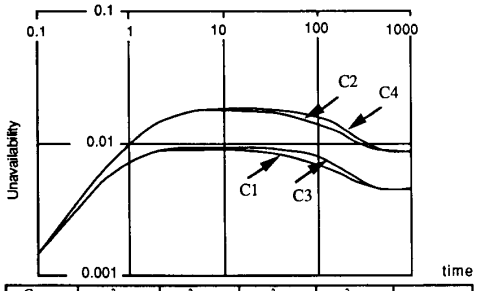
$$A(t) = R_1(t) + \sum_{n=1}^{\infty} \phi_n(t) * R_{n+1}(t) \quad \text{(A5)}$$

where $R_{n+1}(t)$ is the reliability function after n failures:

$$R_{n+1}(t) = P\{T_{n+1} > t\}$$

and the $\phi_n(t)$ are derived in the same manner as in the previous section:

$$\phi_n(t) = f_1(t) * q_1(t) * f_2(t) * q_2(t) * \cdots * f_r(t)^{*[n-r+1]} * q_r(t)^{*[n-r+1]}.$$



Curve	λ_1	λ_2	λ_3	λ_4	μ
C1	$1.5\ 10^{-2}$	$1.1\ 10^{-2}$	$9\ 10^{-3}$	$7\ 10^{-3}$	1.6
C2	$1.5\ 10^{-2}$	$1.1\ 10^{-2}$	$9\ 10^{-3}$	$7\ 10^{-3}$	0.8
C3	$1.5\ 10^{-2}$	$1.4\ 10^{-2}$	$1.2\ 10^{-2}$	$7\ 10^{-3}$	1.6
C4	$1.5\ 10^{-2}$	$1.4\ 10^{-2}$	$1.2\ 10^{-2}$	$7\ 10^{-3}$	0.8

Fig. 25. Unavailability curves: influence of μ, λ_2, and λ_3, (log–log scale).

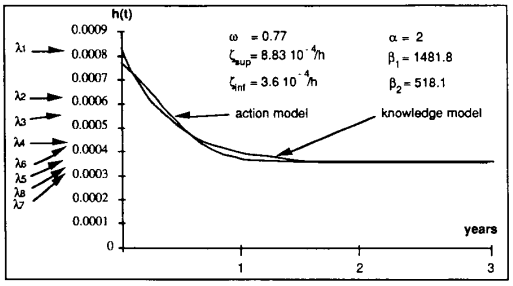


Fig. 26. Compared failure intensities of the knowledge model and of the hyperexponential model.

We assume that the restoration rates are constant; let $\mu_j 1 \leq j \leq r$ denote them. As $\frac{\lambda_j}{\mu_j} \ll 1$, we can perform an asymptotic development with respect to $\frac{\lambda_j}{\mu_j}$ for the unavailability $\overline{A}(t) = 1 - A(t)$; this leads to

$$\overline{A}(t) = \frac{\lambda_r}{\mu_r} + \sum_{n=1}^{r-1} \alpha_n \exp(-\lambda_n t) - \frac{\lambda_1}{\mu_1} \exp(-\mu_1 t) \qquad \text{(A6)}$$

where

$$\alpha_n = \sum_{j=n}^{r-1} \frac{\lambda_j}{\mu_j} \frac{\prod_{i=1}^{j-1} \lambda_i}{\prod_{i=1,\, i\neq n}^{j} (\lambda_i - \lambda_n)} - \frac{\lambda_r}{\mu_r} \prod_{j=1,\, j\neq n}^{r-1} \frac{\lambda_j}{\lambda_j - \lambda_n}.$$

The α_n are related by the equation $\sum_{n=1}^{r-1} \alpha_n = \frac{\lambda_1}{\mu_1} - \frac{\lambda_r}{\mu_r}$.

Equation (A6) gives the unavailability of the knowledge model whose intensity function is (A4).

In Fig. 25, the unavailability curves are plotted for decreasing λ_n and assuming $\mu_n = \mu, \forall n$. The set of the λ_n's corresponds to hypothetical values and, even though the curves are plotted for $r = 4$, they correspond to the general case of reliability growth. The choice of a logarithmic scale allows us to show the evolution of the unavailability at the beginning: it is worth noting that the maximum is reached very quickly (some $\frac{1}{\mu}$), and stable behavior is reached when the terms $\exp(-\lambda_n t)$ become negligible (let us say for $t \simeq 10\frac{1}{\lambda_1}$).

Appendix B
The Hyperexponential Model as an Action Model

The aim of this Appendix is to show how well the hyperexponential model approximates the knowledge models introduced in Section II.

A sequence of random variables $\{\lambda_1, \lambda_2, \cdots, \lambda_r\}$ (which correspond to the successive failure rates) is generated by random sampling such that these variables are stochastically decreasing; the gamma distribution of [41] has been used:

$$\text{pdf}(\lambda_i, \alpha, \psi(i)) = \frac{[\psi(i)]^{\alpha} \lambda_i^{\alpha-1} e^{-\psi(i)\lambda_i}}{\Gamma(\alpha)} \qquad \text{(A7)}$$

where

$$\psi(i) = \beta_1 + \beta_2 * i.$$

The expected values of the successive λ_i are thus

$$E(\lambda_i) = \frac{\alpha}{\beta_1 + \beta_2 * i} \qquad i = 1, 2, \cdots, r \qquad \text{(A8)}$$

from which it can be seen that the sequence $\{\lambda_1, \lambda_2, \ldots, \lambda_r\}$ is such that $\lambda_{i+1} \underset{st}{\leq} \lambda_i$.

The parameters $(\omega, \zeta_{\text{sup}}, \zeta_{\text{inf}})$ of the hyperexponential model are obtained by least-squares estimation conducted on the failure intensities issued from the hyperexponential model and the knowledge model.

Eight failure rates have been generated according to the distribution given by (A7) and computation has been carried out on the knowledge model derived in Appendix A, for restart after correction only (the corresponding intensity function is given by (A4)).

The results are plotted in Fig. 26: it can be seen that the failure intensity curve obtained from the knowledge model is well reproduced by the hyperexponential model. If we look at the different values of the consecutive failure rates, it can be noticed that $\lambda_6 < \lambda_5$ and $\lambda_8 < \lambda_7$ indicating local reliability decrease behavior. However, the failure intensity function is globally decreasing, indicating global reliability growth.

Acknowledgment

The data of the Tropico-R system was provided by J. Moreira de Souza and M. Bastos Martini from Telebras-CPqD, in the framework of a cooperation program between Telebras-CPqD and LAAS-CNRS. Thanks are also due to P. Spiesser from LAAS for his help in numerical processing of the models.

References

[1] N. Adams, "Optimizing preventive service of software products," *IBM J. Res. Develop.*, vol. 28, no. 1, pp. 2–14, Jan. 1984.
[2] A. A. Abdel-Ghaly, P. Y. Chan, and B. Littlewood, "Evaluation of competing software reliability predictions," *IEEE Trans. Software Eng.*, vol. SE-12, no. 9, pp. 950–967, Sept. 1986.
[3] V. Amoia, G. De Micheli, and M. Santomauro, "Computer-oriented formulation of transition-rate matrices via Kronecker algebra," *IEEE Trans. Rel.*, vol. R-30, no. 2, June 1981.
[4] J. Arlat, K. Kanoun, and J. C. Laprie, "Dependability evaluation of software fault-tolerance," in *Proc. 18th IEEE Int. Symp. Fault Tolerant Computing (FTCS-18)*, Tokyo, Japan, June 1988, pp. 142–147.
[5] R. L. Aveyard and F. T. Man, "A study on the reliability of the circuit maintenance system 1-B," *Bell Syst. Tech. J.*, vol. 59, pp. 1317–1332, Oct. 1980.
[6] R. E. Barlow and F. Proschan, *Statistical Theory of Reliability and Life Testing*. New York: Holt, 1975.

[7] A. Birolini, "Some applications of regenerative stochastic processes to reliability theory, Part one: Tutorial introduction," *IEEE Trans. Rel.*, vol. R-23, no. 3, pp. 186–194, Aug. 1974.
[8] H. A. Bauer, L. M. Croxall, and E. A. Davis, "The 5ESS switching system: System test, first-office application, and early field experience," *AT&T Tech. J.*, vol. 64, no. 6, pp. 1503–1522, 1985.
[9] C. Beounes and J. C. Laprie, "Dependability evaluation of complex computer systems: Stochastic Petri net modeling," in *Proc. 15th IEEE Int. Symp. Fault-Tolerant Computing*, Ann Arbor, MI, June 1985, pp. 364–369.
[10] B. Beyaert, G. Florin, P. Lonc, and S. Natkin, "Evaluation of computer systems dependability using stochastic Petri nets," in *Proc. 11th IEEE Int. Symp. Fault-Tolerant Computing*, Portland, ME, June 1981, pp. 79–81.
[11] 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.
[12] A. Costes, J. E. Doucet, C. Landrault, and J. C. Laprie, "SURF: A program for dependability evaluation of complex fault-tolerant computing systems," in *Proc. 11th IEEE Int. Symp. Fault-Tolerant Computing*, Portland, ME, June 1981, pp. 72–78.
[13] D. R. Cox, *Renewal Theory*. London: Methuen, 1962.
[14] D. R. Cox and H. D. Miller, *The Theory of Stochastic Processes*. London: Methuen, 1968.
[15] P. A. Currit, M. Dyer, and H. D. Mills, "Certifying the reliability of software," *IEEE Trans. Software Eng.*, vol. SE-12, no. 1, pp. 3–11, Jan. 1986.
[16] J. T. Duane, "Learning curve approach to reliability monitoring," *IEEE Trans. Aerospace*, vol. 2, pp. 563–566, 1964.
[17] J. M. Finkelstein, "Starting and limiting values for reliability growth," *IEEE Trans. Rel.*, vol. R-28, no. 2, pp. 111–113, June 1979.
[18] B. V. Gnedenko, Y. K. Belyayev, and A. D. Solovyev, *Mathematical Methods of Reliability Theory*. New York: Academic, 1969.
[19] A. L. Goel and K. Okumoto,"Time dependent error detection rate model for software and other performance measures," *IEEE Trans. Rel.*, vol. R-28, pp. 206–211, Aug. 1979.
[20] A. Goyal, W. C. Carter, E. de Souza e Silva, and S. S. Lavenberg, "The system availability estimator," in *Proc. 16th IEEE Int. Symp. Fault-Tolerant Computing*, Vienna, Austria, June 1986, pp. 84–89.
[21] D. Gross and D. R. Miller, "The randomization technique as a modeling tool and solution procedure for transient Markov processes", *Oper. Res.*, vol. 32, no. 2, pp.343–361, 1984.
[22] H. Hecht, "Fault-tolerant software," *IEEE Trans. Rel.*, vol. R-28, pp. 227–232, Aug. 1979.
[23] Z. Jelinski and B. P. Moranda, "Software reliability reseach," in *Statistical Methods fot the Evaluation of Computer System Performance*. New York: Academic, 1972, pp. 465–484.
[24] M. Kaaniche, K. Kanoun, and S. Metge, "Role of the hyperexponential model in the software validation process of a telecommunication equipment," in *Proc. 7th Int. Conf. Reliability and Maintainability*, Brest, France, June 1990, pp. 332–339 (in French).
[25] K. Kanoun and J. C. Laprie, "Modeling software reliability and availability from development validation up to operation," LAAS Res. Rep. 85.042, Aug. 1985.
[26] K. Kanoun and T. Sabourin, "Software dependability of a telephone switching system," in*Proc. 17th IEEE Int. Symp. Fault Tolerant Computing (FTCS-17)*, Pittsburgh, PA, June 1987, pp. 236–241.
[27] K. Kanoun, J. C. Laprie, and T. Sabourin, "A method for software reliability growth analysis and assessment," in *Proc. 1st Int. Workshop Software Engineering and Its Applications*, Toulouse, France, Dec. 1988, pp. 859–878.
[28] K. Kanoun, "Software dependability growth Characterization, modeling, evaluation," Docteur ès-Sciences thesis, Toulouse Polytechnic National Institute, LAAS Rep. 89.320, Sept. 1989 (in French).
[29] K. Kanoun, M. Bastos Martini, and J. Moreira De Souza, "A method for software reliability analysis and prediction application to the TROPICO-R switching system," this issue, pp. 334–344.
[30] P. A. Keiller, B. Littlewood, D. R. Miller, and A. Sofer, "Comparison of software reliability predictions," in *Proc. 13th IEEE Int. Symp. Fault-Tolerant Computing*, Milano, Italy, June 1983, pp. 128–134.
[31] J. G Kemeny and J. L. Snell, *Finite Markov Chains*. Princeton, NJ: Van Nostrand, 1959.
[32] J. C. Laprie, "Dependability modeling and evaluation of hardware-and-software systems," in *Proc. 2nd GI/NTG/GMR Conf. Fault Tolerant Computing*, Bonn, Germany, Sept. 1984, pp. 202–215.
[33] ——, "Dependability evaluation of software systems in operation," *IEEE Trans. Software Eng.*, vol. SE-10, no. 6, pp. 701–714, Nov. 1984.
[34] ——, "Towards an X-ware reliability theory," *Technique et Science Informatiques*, vol. 7, no. 3, pp. 315–330, 1988 (in French); Available in English as LAAS Rep. 86.376, Dec. 1986.
[35] ——, "Hardware-and-software dependability evaluation," in *Proc. IFIP 11th World Computer Congress*, San Francisco, CA, Aug. 1989, pp. 109–114.
[36] J. C. Laprie, C. Beounes, M. Kaaniche, and K. Kanoun, "The transformation approach to modeling and evaluation of the reliability and availability growth of systems," in *Proc. 20th IEEE Int. Symp. Fault Tolerant Computing (FTCS-20)*, Newcastle, England, June 1990, pp. 364–371.
[37] Y. Levendel, "Defects and reliability analysis of large software systems: Field experience," in *Proc. 19th IEEE Int. Symp. Fault Tolerant Computing (FTCS-19)*, Chicago, IL, June 1989, pp. 238–244.
[38] P. A. Lewis, "A branching Poisson process model for the analysis of computer failure patterns," *J. Roy. Statist. Soc. B*, vol. 26, no. 3, pp. 398–456, 1964.
[39] B. Littlewood and J. L. Verral, "A Bayesian Reliability Growth Model for Computer Software," *J. Roy. Statist. Soc. C (Appl. Statist.)*, vol. 22, pp. 332–336, 1973.
[40] B. Littlewood, "Software reliability model for modular program structure," *IEEE Trans. Rel.*, vol. R-28, no. 3, pp. 241–246, Aug. 1979.
[41] ——, "Theories of software reliability: How good are they and how can they be improved?" *IEEE Trans. Software Eng.*, vol. SE-6, no. 5, pp. 489–500, Sept. 1980.
[42] ——, "Stochastic reliability growth: A model for fault-removal in computer programs and hardware designs," *IEEE Trans. Rel.*, vol. R-30, no. 4, pp. 313–320, Oct. 1981.
[43] A. Marsan, G. Balbo, and G. Conte, "A class of generalized stochastic Petri nets for the performance analysis of multiprocessor systems," *ACM Trans. Comput.*, vol. 2, no. 2, pp.93–122, May 1984.
[44] S. Metge, "Analysis and evaluation of the reliability of two ESS software systems," Eng. thesis, CNAM, Rep. LAAS 90.112 May 1990 (in French).
[45] D. R. Miller, "Exponential order statistic models of software reliability growth," *IEEE Trans. Software Eng.*, vol. SE-12, no. 1, pp. 12–24, Jan. 1986.
[46] M. Molloy, "Performance analysis using stochastic Petri nets," *IEEE Trans. Comput.*, vol 39, no. 9, pp.913–917, Sept. 1982.
[47] J. D. Musa, "Software reliability data," Data and Analysis Centre for Software, Rome Air Development Center (RADC), Rome, NY, Tech. Rep., 1979.
[48] J. D. Musa and K. Okumoto, "A logarithmic Poisson execution time model for software reliability measurement," in *Proc. Compsac'84*, Chicago, IL, 1984, pp. 230–238.
[49] P. M. Nagel and J. A. Skrivan, "Software reliability: repetitive run experimentation and modeling," NASA Rep. CR-165 836, Feb. 1982.
[50] M. Ohba, "Software reliability analysis models," *IBM J. Res. Develop.*, vol. 21, no. 4, pp. 428–443, July 1984.
[51] P. I. Pignal, "An analysis of hardware and software availability exemplified on the IBM 3725 communication controller," *IBM J. Res. Develop.*, vol. 32, no. 2, pp. 268–278, Mar. 1988.
[52] A. Reibman, R. Smith, and K. Trivedi, "Markov and Markov reward model transient analysis: An overview of numerical approaches," *European J. Oper. Res.*, vol. 40,, pp. 257–267, 1989.
[53] W. B. Rohn and T. F. Arnold, "Design for low expected downtime control systems," in *Proc. 4th Int. Conf. Computer Communications*, Philadelphia, PA, June 1972, pp. 16–25.
[54] G. E. Stark, "Dependability evaluation of integrated hardware/software systems," *IEEE Trans. Rel.*, vol. R-36, no. 4, pp. 440–444, Oct. 1987.
[55] W. J. Stewart and A. Goyal, "Matrix methods in large dependability models," IBM Res. Rep. RC-11485, Yorktown Heights, NY, Nov. 1985.
[56] Y. Tohma, K. Tokunaga, S. Nagase, and Y. Murata, "Structural approach to the estimation of the number of residual faults based on the hyper-geometric distribution," *IEEE Trans. Software Eng.*, vol. 15, no. 3, pp. 345–355, Mar. 1989.
[57] K. S. Trivedi, "Reliability evaluation for fault-tolerant systems," in *Mathematical Computer Performance and Reliability*, G. Iazeolla, P. J. Courtois, and A. Hordijk, Eds. Amsterdam, The Netherlands: North-Holland, 1984, pp. 403–414.

[58] J. J. Wallace and W. W. Barnes, "Designing for ultrahigh availability: The Unix RTR operating system," *Computer*, pp. 31–39, Aug. 1984.
[59] S. Yamada and S. Osaki, "Reliability growth modeling for software error detection," *IEEE Trans. Rel.*, vol. R-32, no. 5, pp. 475–478, 1983.
[60] S. Yamada and S. Osaki, "Software reliability growth modeling: Models and assumptions," *IEEE Trans. Software Eng.*, vol. SE-11, no. 12, pp. 1431–1437, Dec. 1985.

Jean-Claude Laprie (M'83) received the Certified Engineer degree from the Higher National School for Aeronautical Constructions, Toulouse, France, in 1968, and the Doctor of Engineering degree in automatic control and the Doctor ès-Sciences degree in computer science from the University of Toulouse in 1971 and 1975, respectively.

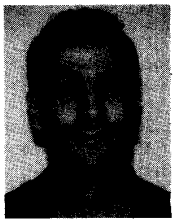
He is currently Directeur de Recherche of CNRS, the French National Organization of Scientific Research. He joined LAAS-CNRS in 1968, where he has directed the research group on fault tolerance and dependable computing since 1975. His research has focused on dependable computing since 1973, and especially on fault tolerance and on dependability evaluation, subjects on which he has authored and coauthored more than 100 papers, as well as several books; he is the Principal Investigator of several contracts in these areas of interest. From January to August 1985, he was an Invited Visiting Professor at the University of California, Los Angeles, Department of Computer Science. He has also acted as a consultant and expert in the area of dependable computing in France and abroad for government agencies as well as industrial organizations.

Dr. Laprie served as the General Chairman of the 8th International Symposium on Fault Tolerant Computing in 1978, as well as on program committees for numerous conferences and workshops. He was the Chairman of the IEEE Computer Society's Technical Committee on Fault Tolerant Computing in 1984 and 1985 and is now the Chairman of the IFIP Working Group 10.4 on Reliable Computing and Fault Tolerance. He is the founding Chairman of the AFCET (French Association for Economics and Techniques of Cybernetics) Group on Computing System Dependability. He is co-editor of the Springer-Verlag series, *Dependable Computing and Fault Tolerant Systems*. He is a member of the Association for Computing Machinery and AFCET.

Karama Kanoun, for a photograph and biography, see this issue, p. 344.

Christian Béounes received the Certified Engineer degree from the National Institute of Applied Sciences, Toulouse, France, in 1973 and the Doctor in Engineering degree in automatic control from the University of Toulouse in 1977.

He is currently Chargé de Recherche of INRIA, the French National Institute for Computing and Automatic Control Research. He joined LAAS in 1974 as a member of the Dependable Computing and Fault Tolerance Group. His current research interests include stochastic Petri net modeling and dependability evaluation of computer systems.



Mohamed Kaâniche was born in Sfax, Tunisia, in 1963. He received the Engineer degree from the National School of Civil Aviation of Toulouse, France, in 1987, and the Diploma for Further Studies in computer science and automation from the University of Toulouse in 1988.

He is now a member of the Dependable Computing and Fault Tolerance Group at LAAS-CNRS. He is preparing a Ph.D. dissertation on software reliability growth modeling and evaluation of single and multicomponent systems.