Fault Injection and Dependability Evaluation of Fault-Tolerant Systems

Jean Arlat, *Member, IEEE,* Alain Costes, Yves Crouzet, Jean-Claude Laprie, and David Powell, *Member, IEEE*

Abstract— This paper describes a dependability evaluation method based on fault injection that establishes the link between the experimental evaluation of the fault tolerance process and the fault occurrence process. The main characteristics of a fault injection test sequence aimed at evaluating the coverage of the fault tolerance process are presented. Emphasis is given to the derivation of experimental measures. The various steps by which the fault occurrence and fault tolerance processes are combined to evaluate dependability measures are identified and their interactions are analyzed. The method is illustrated by an application to the dependability evaluation of the distributed fault-tolerant architecture of the ESPRIT Delta-4 Project.

Index Terms— Coverage, dependability modeling and evaluation, experimental evaluation, fault injection, fault tolerance, Markov chains.

I. INTRODUCTION

THE EVALUATION of a fault-tolerant system is a complex task that requires the use of different levels of modeling (axiomatic, empirical, and physical models) and related tools [1]. A large number of studies (e.g., see [2]–[4]), have shown the prominence of the efficiency of the fault tolerance algorithms and mechanisms (FTAM's) on the dependability of a wide range of systems and architectures. Determination of the appropriate model for the fault tolerance process and proper estimation of the associated coverage parameters are therefore essential in any dependability evaluation study.

Compared with other possible approaches such as proving or analytical modeling, *fault injection* is particularly attractive [5]–[13]. By speeding up the occurrence of errors and failures, fault injection is a method for *testing* the FTAM's with respect to their own specific inputs: *the faults* that they are intended to tolerate.

As pointed out in [14], fault injection addresses both dimensions of FTAM validation: *fault removal and fault forecasting* [15], [16]. With respect to the *fault removal* objective, fault injection is explicitly aimed at reducing, by *verification,* the presence of FTAM design and implementation faults. Since such faults can cause incorrect behavior of the FTAM's when they are faced with the faults they are intended to handle, we call them *fault-tolerance deficiency* faults (in short, *ftd-faults*).

From the verification viewpoint, fault injection therefore aims to reveal such ftd-faults and to determine appropriate actions to correct the design or implementation of the FTAM's. In the case of *fault forecasting,* the main issue is to rate, by *evaluation,* the *efficiency* of the operational behavior of the FTAM's. This type of test thus constitutes primarily a test of the FTAM's with respect to their overall behavioral specification. In practice, this means estimating the parameters that characterize the operational behavior of the FTAM's: coverage factors, dormancy, latency, etc.

Both dimensions are of interest for validating the FTAM's. The relationships and complementarity between these two objectives, as well as the main characteristics of the ftd-fault removal objective, are addressed in [14], [17], [18]. This paper focuses on the fault forecasting objective.

The fault tolerance coverage estimations obtained through fault injection experiments are estimates of *conditional* probabilistic measures characterizing dependability. They need to be related to the fault occurrence and activation rates to derive overall measures of system dependability. Such a necessary relationship is—at least conceptually—well established. However, few studies consider its actual incorporation into the dependability evaluation of real fault-tolerant systems. Among the most significant related studies, see the work reported in [19], the ESS, SIFT, and FTMP validation processes depicted in [20, ch. 12, 16, 17] and, more recently, the evaluation of the MAFT architecture presented in [21].

This paper describes a *dependability evaluation method* based on fault injection that establishes the link between the experimental evaluation of the coverage of the fault tolerance process and the fault occurrence process. The paper also illustrates the application of the method to the *evaluation of a real system.* Such an experiment-based evaluation method combining fault injection experiments and analytical evaluation has been—along with formal protocol verification activities—the central point in the validation of the distributed fault-tolerant architecture of the ESPRIT Delta-4 Project (see [22, ch. 15] for a global description of the validation tasks). Markov-based modeling and evaluation, and especially sensitivity analysis of the impact of the coverage parameters (both coverage factors and latencies), helped to identify the most significant parameters to be estimated from the fault injection experiments. Conversely, the experiments not only made it possible to obtain the range of values for the coverage parameters used in the analytical models, but also helped in the validation and refinement of these models. In particular,

the models were refined to capture specific behaviors revealed by the experiments.

More recently, the study presented in [23] described an example of cross-fertilization between experimental evaluation and analytical modeling. However, that study relied more on the analysis of recorded field data than on fault injection experiments. The physical fault injection experiments carried out on the Delta-4 prototype testbed made it possible to iterate the evaluation process for validating the design assumptions (e.g., the *fail-silence* assumption) and thus had an impact—albeit during the final phases—on the development of the Delta-4 architecture.

The paper defines and analyzes the relationships between experimental and analytical dependability evaluation. The results obtained in the case of the evaluation of a real system provide practical examples of such relationships. The remainder of this paper consists of four sections. Section II depicts the main characteristics of a fault injection test sequence aimed at evaluating the fault tolerance process. This section—adapted and extended from [24]—summarizes some definitions and results that are necessary for the understanding of the developments presented in the next section. Section III describes the main steps of the integration of the fault occurrence and fault tolerance processes that were defined and fully detailed in [25]. Section IV applies the method to the dependability evaluation of the Delta-4 distributed fault-tolerant architecture. Section V concludes the paper.

II. EXPERIMENTAL EVALUATION OF FAULT TOLERANCE

The proposed experimental evaluation method embodies the concept of a fault injection *test sequence*, characterized by an *input* domain and an *output* domain. The input domain corresponds to a set of injected *faults* F and a set A that specifies the data used for the *activation* of the target system and thus, of the injected faults. The output domain corresponds to a set of *readouts* R that are collected to characterize the target system behavior in the presence of faults and a set of *measures* M that are derived from the analysis and processing of the *FAR* sets.

Together, the *FARM* sets constitute the major attributes that can be used to fully characterize a fault injection test sequence. In practice, the fault injection test sequence consists of a series of *experiments;* each experiment specifying a particular point in the $\{F \times A \times R\}$ space.

A. Characterization of a Fault Injection Test Sequence

During each *experiment* in a fault injection test sequence, a fault from the F set is injected that, in conjunction with the activity of the target system (A set), determines an error pattern that constitutes a test input for the FTAM's to be validated. For increased confidence in the estimates obtained, it is necessary to carry out a large number of experiments. For minimum bias in the estimation, it is further recommended to select both F and A sets by *statistical sampling* among the expected operational fault and activation domains of the target fault tolerant system. Further issues concerning the combination of the F and A sets to produce error patterns

Predicates — F: Fault, E: Error, D: Detection, T: Tolerance
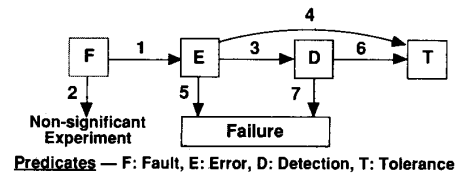
Fig. 1. Example of a predicate graph.

are discussed in detail in [14]; we focus here on the R and M sets characterizing the experimental evaluation process.

The readouts collected in R during an experiment contribute to a characterization of the state of the target system. This is achieved by way of the assertion or not of a set of predicates that are meant to abstract the specification of the behavior of the target system and thus of the FTAM's under test. Typical examples of predicates are: {fault_activated}, {fault_activated error_signaled}, {error_signaled proper service delivered}. Such predicates or their combinations define the set of vertices of a graph that models the behavior of the target system (or of the FTAM's) in the presence of faults. This graph can be either established *a priori* to describe anticipated behaviors or obtained *a posteriori* from the analysis of the R set, which is a form of model extraction from the experimental results (e.g., see [12]).

Fig. 1 gives an example of such a graph. Transition 1 corresponds to the activation of an injected fault as an error; the associated time defines the *fault dormancy*. Transition 2 represents the situation where an injected fault is not activated; such an experiment is not significant when FTAM coverage is evaluated with respect to error patterns (resulting from activated faults) rather than with respect to the faults injected. Transition 3 depicts the case of a detected error; the associated time characterizes the *latency of error detection*. Transition 4 represents the case where an error is apparently tolerated although it was not detected whereas transition 6 depicts the (normal) situation where the error is tolerated after having been detected. Transitions 5 and 7 distinguish the cases of failure of the detection and tolerance mechanisms. This graph depicts the faulty behavior observed during the experiments carried out on the Delta-4 architecture. In particular, transition 4 characterizes a singular behavior, that is not always easy to diagnose in practice since it may result from either 1) an activated fault that remains hidden (latent) or 2) a propagated error that is tolerated or that is eliminated by some other—unobserved—mechanism.

Fig. 2 further illustrates the types of predicates and system state transitions that can be deduced from the readout set R, in the case of a single binary predicate p; three principal cases are accounted for, depending on whether the predicate is expected 1) to maintain its value for the whole interval $T = [t_F, t_M]$ that defines the observation domain for an experiment (Fig. 2(a)) or 2) to change value once (Fig. 2(b)) or several times (Fig. 2(c)) during the experiment.

A typical example of Fig. 2(a) is the case of a reliability or availability predicate characterizing the *continuity of service*
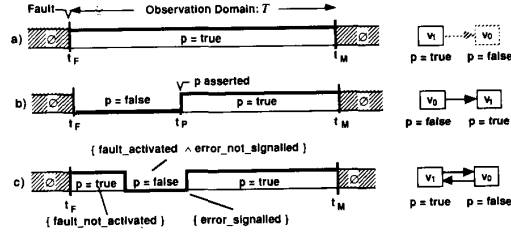
Fig. 2. Examples of tests of predicates.

delivery in the presence of faults (e.g., fault masking):

$$p = \text{true} \Leftrightarrow \{\text{acceptable\_results\_delivered}\}$$

and

$$p = \text{false} \Leftrightarrow \{\text{erroneous\_result\_delivered}\}.$$

The testability property, for which an error must be signaled whenever a fault is present, is a possible example for Fig. 2(b):

$$p = \text{true} \Leftrightarrow \{\text{error\_signaled}\} \tag{1}$$

and

$$p = \text{false} \Leftrightarrow \{\text{error\_not\_signaled}\}.$$

Fig. 2(c) provides an example for the test of a *fail-safe* property defined as

$$p = \text{true} \Leftrightarrow \{\text{fault\_not\_activated} \vee \text{error\_signaled}\}$$

and

$$p = \text{false} \Leftrightarrow \{\text{fault\_activated} \wedge \text{error\_not\_signaled}\}$$

where $\vee$ and $\wedge$ denote, respectively, OR and AND connectives.

This corresponds to an alternating behavior between graph vertices v_0 and v_1 that may be described by the decomposition of the predicate p into two elementary predicates of the types shown in Fig. 2(b):

$$p = (\neg p_1) \vee p_2$$

with $p_1 = \{\text{fault\_activated}\}$ and $p_2 = \{\text{error\_signaled}\}$ where $\neg$ is the NOT operator.

The observation of the instant of assertion of a predicate characterizes the temporal performance of the FTAM under test; in particular for the predicate of Fig. 2(b), (1) can be modified to

$$p = \text{true} \Leftrightarrow \exists t_p \in T.$$

Since relevant timing measurements are related to the instant of fault occurrence, it is simpler to consider hereafter that the observation domain T is defined by the interval $[0, T]$.

B. Definition of Experimental Measures

We only summarize here the major experimental measures that can be derived from a fault injection test sequence.

Let T_p denote the random variable characterizing the *instant of assertion of a predicate p*; then, the *cumulative distribution function of the coverage* (with respect to predicate p) can be defined as

$$C(t) = \text{prob.}\,\{T_p \leq t\}. \tag{2}$$

Other related studies (e.g., see [26]) focus on the probability density function of the coverage. Both approaches are equivalent in principle; however, we advocate the use of the cumulative function as this facilitates the relationship with analytical models: the asymptotic value simply tends towards the constant coverage parameters usually used in these models.

Two principal constraints have to be considered in the derivation of experimental measures. First, it is worth noting that $C(t)$ is usually defective (e.g., see [3]) since all the faults cannot be properly covered, thus its asymptotic value is less than or equal to one, i.e.,

$$C(\infty) = \lim_{t \to \infty} C(t) \leq 1. \tag{3}$$

Also, the observation domain T is bounded and the readouts obtained from the experiments form a set of so-called Type I (or time) censored data (e.g., see [27, p. 248]); the unobserved times are known only to be above the upper bound T (*censoring time*) of the observation domain. The characteristics of the considered target system and especially the temporal parameters of the FTAM's to be evaluated have a direct impact on the determination of T. The choice of T relies on a careful analysis of the *a priori* (partial) information available concerning the temporal parameters of the FTAM's and may necessitate a set of preliminary experiments for its proper adjustment.

The combination of these two constraints results in a total uncertainty for the experiments for which no outcome (predicate assertion) is observed. Indeed, either the assertion would occur in a finite time beyond T or the assertion is not true for that experiment (which denotes a coverage deficiency). These implications are further analyzed in the following subsections.

1) Basic Estimation of the Coverage Function: Consider a test sequence of n independent fault injection experiments; in each experiment, a point in the $\{F \times A\}$ space is randomly selected according to the distribution of occurrences in $\{F \times A\}$ and the corresponding readouts collected. If t_{pi} denotes the instant of assertion of p for experiment $i, i = 1, \cdots, n$, let $\varepsilon_i(t)$ denote the random variable defined by

$$\varepsilon_i(t) = I(t_{pi} \leq t) = \begin{cases} 1, & \text{if assertion of } p \text{ is observed in } [0, t] \\ 0, & \text{otherwise.} \end{cases}$$

The number of assertions of p cumulated within the time interval $[0, t]$ can thus be expressed as

$$N(t) = \sum_{i=1}^{n} \varepsilon_i(t) \tag{4}$$

and the *coverage function* $C(t)$ can be simply estimated by

$$\hat{C}(t) = \frac{N(t)}{n}. \qquad (5)$$

The *asymptotic coverage* is estimated by

$$\hat{C}(T) = \frac{N(T)}{n}. \qquad (6)$$

Due to the monotonically increasing behavior of $C(t)$ and to the finite restriction of the observation domain, this estimation is always pessimistic. Furthermore, as $C(t)$ is defective, another interesting measure corresponds to the *conditional coverage* expressed as

$$\text{prob.}\,\{T_p \leq t | T_p \leq \infty\} = \frac{\text{prob.}\,\{T_p \leq t\}}{\text{prob.}\,\{T_p \leq \infty\}} = \frac{C(t)}{C(\infty)}. \qquad (7)$$

This experimental conditional coverage also refers to the conditional distributions defined for the coverage model presented in [3].

If T_p' designates the random variable characterizing the noninfinite coverage times (noninfinite instants of assertion of p), then T_p' can be described by the following distribution:

$$C'(t) = \text{prob.}\,\{T_p' \leq t\} = \frac{C(t)}{C(\infty)}, \qquad (8)$$

that is estimated by

$$\hat{C}'(t) = \frac{C(t)}{\hat{C}(T)}. \qquad (9)$$

2) Estimation of the Mean Coverage Time: The mean coverage time is defined by: $\tau = E[T_p] = \int_0^\infty t \cdot dC(t)$. The two constraints identified previously also complicate the estimation of τ; three types of estimators can be considered:

$$\hat{\tau}' = \frac{1}{N(T)} \sum_{i=1}^{n} \varepsilon_i(T) \cdot t_{pi} = \frac{n}{N(T)} \frac{1}{n} \sum_{i=1}^{n} \varepsilon_i(T) \cdot t_{pi}$$

$$= \frac{1}{\hat{C}(T)} \frac{1}{n} \sum_{i=1}^{n} \varepsilon_i(T) \cdot t_{pi} \qquad (10)$$

$$\hat{\tau}'' = \frac{1}{n} \sum_{i=1}^{n} \{\varepsilon_i(T) \cdot t_{pi} + [1 - \varepsilon_i(T)] \cdot T\}$$

$$= \hat{C}(T) \cdot \hat{\tau}' + [1 - \hat{C}(T)] \cdot T \qquad (11)$$

$$\hat{\tau}''' = \frac{1}{N(T)} \sum_{i=1}^{n} \{\varepsilon_i(T) \cdot t_{pi} + [1 - \varepsilon_i(T)] \cdot T\}$$

$$= \hat{\tau}' + \frac{1 - \hat{C}(T)}{\hat{C}(T)} \cdot T = \frac{1}{\hat{C}(T)} \cdot \hat{\tau}''. \qquad (12)$$

The first estimator given by (10) corresponds to the estimation of the mean of the coverage times actually observed. It is thus an estimator of $E[T_p']$, i.e., of the mean of the
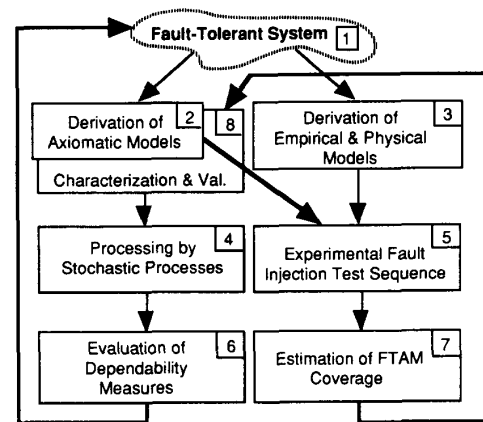

Fig. 3.   Interactions between analytical and experimental dependability evaluation.

conditional coverage time. The second estimator defined by (11) estimates the random variable $\min(T_p, T)$. It has been modified to assign a time T (i.e., the upper bound of the observation domain) to each of the $[n - N(T)]$ experiments for which the assertion of p was not observed. The third estimator (12) corresponds to the estimator typically used when dealing with time-censored exponentially distributed test data (e.g., see [28, pp. 105–106]) or with the estimation of the mean time to first failure (MTFF) [29].

It is worth noting that $\hat{\tau}' \leq \hat{\tau}'' \leq \hat{\tau}'''$; it follows that $\hat{\tau}'$ constitutes an "optimisitic" estimation of the mean coverage time. However, the fact that $C(t)$ is defective prevents conclusions being drawn about the bias induced by the other estimators. We therefore selected the first estimator.

III. INTEGRATION OF EXPERIMENTAL MEASURES OF FAULT TOLERANT WITH THE FAULT OCCURENCE PROCESS

In this section we first identify the main *interactions* between analytical dependability modeling and experimental evaluation. We then present a framework for characterizing the *relationship* between the experimental estimates obtained in a fault injection test sequence and the coverage parameters usually considered to account for FTAM behavior in Markov chain models. An example is given to illustrate the respective impact on dependability evaluation of *asymptotic* coverage and coverage *distribution*.

A. Bridging the Gap between Analytical Modeling and Fault Injection

Fig. 3 depicts the principal phases of analytical dependability evaluation and experimental dependability evaluation based on fault injection that rely respectively on the construction and the processing of either axiomatic models (*sequence 1-2-4-6*), or empirical and physical models (*sequence 1-3-5-7*).

Of course, both sequences may be used separately to impact the target system (e.g., parameter sensitivity analysis for early architectural design decisions in the case of model-based

evaluation or as a design aid for fault removal in the case of fault injection-based experimental testing). However, we would like to stress here the benefits that can be obtained from the interactions between these two sequences. For sake of conciseness, we will emphasize only the most significant interactions (identified by bold arrows in Fig. 3).

The transition from 2 to 5 depicts the necessary impact of modeling on the definition of the readouts in the R set and the determination of the measures in the M set. In particular, one impact represented by this transition may be that of considering the relative ratios of the occurrence rates of different fault classes in order to refine the general estimators of the coverage function given in Section II (e.g., see [30] and [31]).

The transition from 7 to 8 identifies two types of interactions:

- *Impact of Models on Experiments:* The reference to the fault occurrence process, usually described in axiomatic models, is necessary to derive dependability measures.
- *Impact of Experiments on Models:* This includes estimation of the coverage parameters of the original models, validation of the assumptions made in the elaboration of these models, and refinement of the structure of the models.

Relevant measures of system dependability can be obtained by processing models thus supported by experiments. This provides an objective foundation for proposing modifications to the design and implementation of the target fault-tolerant system.

The interactions induced by transition 7–8 are analyzed further in the next subsection.

B. Dependability Evaluation

If we assume that the major risk of system failure is that induced by the failure of the FTAM's in properly processing the first fault occurrence, the reliability expression for a nonmaintained fault-tolerant system can be written as

$$R(t) = 1 - \Phi_F(t) + \int_0^t \phi_F(t_F) \cdot C(t - t_F) \cdot dt_F \quad (13)$$

where $\Phi_F(t)$ and $\phi_F(t)$ are, respectively, the cumulative distribution and density functions characterizing the fault occurrence process of the whole target fault-tolerant system and $C(t)$ designates the cumulative distribution of the FTAM coverage function (see Section II). In particular, the first part of (13), $1 - \Phi_F(t)$ expresses the probability that *no fault* occurred before t and the last term expresses the probability of survival to the *first component failure*.

The derivation of (13) is based on the fact that, in a fault-tolerant system, the risks of failure resulting from exhaustion of redundancy correspond generally to much lower orders of magnitude than those induced by a coverage deficiency in the FTAM's. This is especially true when the mission time is small compared to the mean time to fault occurrence. It should also be pointed out that the reference to the fault occurrence process
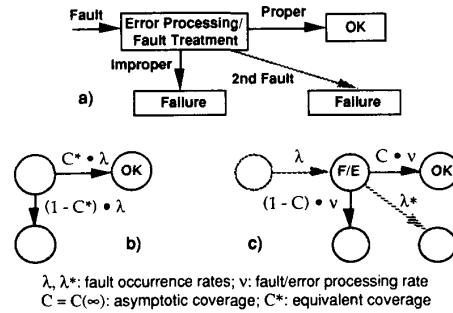
λ, λ*: fault occurrence rates; v: fault/error processing rate
C = C(∞): asymptotic coverage; C*: equivalent coverage

Fig. 4. Markov model of the coverage distribution.

is by no means a limiting factor: the extension to the error occurrence process (fault activation) can be simply achieved by substituting E for F in the indexes.

Three major techniques (namely, Monte Carlo simulation, closed-form expressions, and Markov chains) can be considered for implementing the relationship between the fault occurrence process and the experimental coverage parameters formally expressed by (13) (see [14]). Of these three, Markov chains are especially attractive since they provide a tractable means to account for the main temporal characteristics of the coverage distribution, as exemplified in the following subsections.

1) Estimation of the Coverage Parameters of a Markov Chain: Let us consider the model of Fig. 4(a) that describes the behavior of a fault-tolerant system. This model accounts for the coverage of the FTAM's with respect to the occurrence of a fault and the possible occurrence of a second (near-coincident [32]) fault while processing the first one.

As shown in [3], an equivalent Markov representation (Fig. 4(b)) can be derived for such a behavior where the equivalent coverage C^* is defined as

$$C^* \approx C \left(1 - \lambda^* \cdot E[T_d] + \frac{(\lambda^*)^2}{2!} \cdot E[T_D^2] - \cdots \right) \quad (14)$$

where the constant parameter C can be identified as the asymptotic value of the coverage cumulative distribution $C(t)$ (see Section II), λ^* is the rate of occurrence of a near coincident fault, and the $E[T_d^i]$ designates the successive moments of the random variable characterizing the processing time of the FTAM's.

By limiting (14) to the first order and letting $\nu = 1/E[T_d]$, the decision rate of the FTAM's, we obtain the model of Fig. 4(c). This model provides an essential "building block" to describe the coverage process, in particular for studying the impact of the temporal distribution.

Although the truncation of the observation domain leads to a conservative estimation of the asymptotic coverage [see (6)], the estimation of the distribution of T_d is in practice more complex. Basically, the distribution of T_d can be related to the distribution of the random variable T_p' characterizing the noninfinite coverage times [see (10)] which is in turn related to the random variable T_p characterizing the coverage process
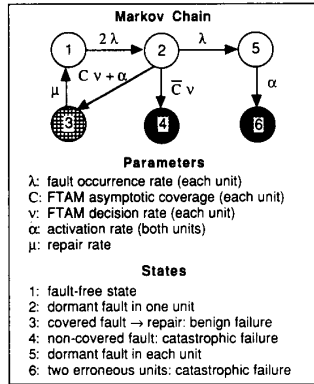
Fig. 5. Duplex system model.

(i.e., the assertion of predicate p) by

$$
\begin{aligned}
C(t) &= \text{prob.} \{T_p \le t\} = \text{prob.} \{\text{assertion of } p \text{ in } [0, t]\} \\
&= \text{prob.} \{\text{assertion of } p \text{ in } [0, t] | \text{assertion of } p \text{ in } [0, \infty[\} \\
&\quad \cdot \text{prob.} \{\text{assertion of } p \text{ in } [0, \infty[\} \\
&= \text{prob.} \{T_p \le t | T_p < \infty\} \cdot \text{Prob.} \{T_p < \infty\} \\
&= \text{prob.} \{T_p' \le t\} \cdot C(\infty)
\end{aligned}
\tag{15}
$$

Accordingly, $E[T_d] \approx E[T_p]/C$, and thus (14) can be simply expressed as

$$
C^* \approx C - \lambda^* \cdot E[T_p],
\tag{16}
$$

and $E[T_p]$ can then be (under-)estimated by the estimator given in (10).

2) Impact of the Time Distribution of the Fault Tolerance Process: In order to study this impact, we consider as an example the case of a duplex architecture featuring 1) self-checking units whose coverage is characterized by an asymptotic coverage C and a mean decision time denoted by $1/\nu$ and 2) a procedure for cross-checking both units—with perfect coverage—and whose timing is characterized by the activation process that is *common* to both units; let α denote the associated rate. Fig. 5 describes the considered model and defines the model parameters as well as the meaning of the states. This model corresponds to the basic model used in the safety and availability evaluation of the potential architectures for the computerized interlocking of the French National Railways [33].

The analysis of the failure states explicitly distinguishes whether or not an error was detected. Accordingly, state 3, although unreliable since the service delivery has been interrupted during the repair action that follows the detection of an error, can be considered as a *safe* state (benign failure); therefore, only states 4 and 6 are catastrophic failure states. State 5 represents the system after a second failure but before (re-)activation of the system. Its positive effect on system dependability is usually very slight (since $\alpha \gg \lambda$) and can be neglected (by merging it into state 6).

For the evaluation, we use the equivalent catastrophic failure rate (denoted Γ) associated with the absorbing states 4 and 6. The *strong connectivity* property of the graph consisting of

the nonabsorbing states as well as the very small values that are usual for the model parameter ratios λ/α and λ/ν ensure that the absorption process is asymptotically a homogeneous Poisson process and that the associated equivalent failure rate Γ is given by (e.g., see pp. 178–182 of [34]):

$$
\Gamma = \sum_{\substack{\text{paths from} \\ \text{initial state } (I) \text{ to} \\ \text{failed state(s)}}}
\left\{ \frac{\prod \text{transition rates of the considered path}}{\displaystyle\prod_{\substack{\text{states in path} \\ (I \text{ expected})}} [\sum \text{output rates of the considered state}]} \right\}.
\tag{17}
$$

Application of (17) to the model of Fig. 5 and some algebraic manipulations lead to the following normalized failure rate:

$$
\frac{\Gamma}{\lambda} = \frac{2 \cdot \left(\overline{C} + \dfrac{\lambda}{\nu}\right)}{1 + \dfrac{\lambda}{\nu} + \dfrac{\alpha}{\nu}} = \frac{2 \cdot \dfrac{\lambda}{\alpha}\left(\overline{C} + \dfrac{\lambda}{\nu}\right)}{\dfrac{\lambda}{\nu} + \dfrac{\lambda}{\alpha} + \dfrac{\lambda^2}{\nu \cdot \alpha}}.
\tag{18}
$$

Expression (18) reveals the prominent role of the asymptotic noncoverage of the self-checking mechanisms $(\overline{C})$, or by the activation rate (α), according to the value of the ratio λ/ν (with respect to $\overline{C}$). It is worth noting that this ratio corresponds to the normalized mean decision time $(1/\nu)$—with respect to the MTFF of one unit (i.e., $1/\lambda$). These results extend and refine the results usually found in the existing literature, which are mainly restricted to the influence of the asymptotic coverage. It is also worth noting that the ratio obtained when inverting (18) corresponds to the ratio of the MTFF procured by the redundant duplex architecture (i.e., MTFFduplex $\approx 1/\Gamma$) over the MTFF of one unit (MTFFunit $= 1/\lambda$). This is illustrated by the curves shown in Fig. 6 that plot the gain in MTFF procured by the redundant duplex architecture as a function of the ratio λ/ν. Fig. 6(a) illustrates the impact of the lack of coverage $(\overline{C})$, while Fig. 6(b) illustrates the influence of the activation rate through the normalized mean activation time (λ/α).

The curves provide useful insight about the domains where the impact of the FTAM coverage time distribution is significant. The variations observed explicitly show that, for the usual order of magnitudes of the ratio λ/ν, i.e., $\lambda/\nu \ll 1$, the impact of the asymptotic coverage is the most prominent. This clearly indicates that, in the experimental evaluation, specific attention should be paid to the estimation of asymptotic coverages.

IV. EXAMPLE OF FAULT INJECTION-BASED DEPENDABILITY EVALUATION

This section illustrates the concepts set forth in the previous sections by applying them to the Delta-4 distributed fault-tolerant architecture. The reader interested by the Delta-
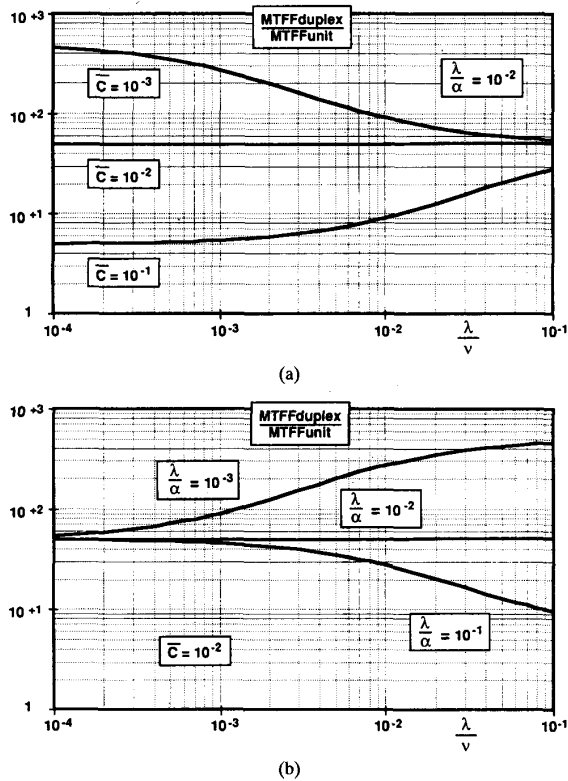
(a)

(b)

Fig. 6. Impact of the temporal and asymptotic parameters of the FTAM's.

4 architecture can refer to [22] and [35]. Two issues are considered here:

- model construction, exemplified by the description of a typical *experimental graph,*
- calibration of coverage parameters for the *evaluation of dependability measures.*

A. Experimental Graph

The target system considered for the experimental validation of the Delta-4 architecture is made up of a local area network of four nodes. Each node is composed of a host computer and of a *network attachment controller* (NAC). The NAC features *hardware self-checking mechanisms* specifically designed to ensure a *fail-silent* behavior (by provoking the extraction of a faulty node from the network). Tolerance of faults at the host computer level is achieved through data and code replication and a variety of alternate mechanisms of which the basic building block is an *atomic multicast protocol* (AMp) also implemented in the NAC.

The fault injection test sequence was aimed at testing the hardware self-checking mechanisms implemented in the NAC's as well as the behavior of the AMp software in the presence of hardware faults. Faults were injected in the NAC of a single node (the *faulty node*) that was monitored to assess the efficiency of its hardware self-checking mechanisms. Successful hardware error-detection (resulting in node extraction)
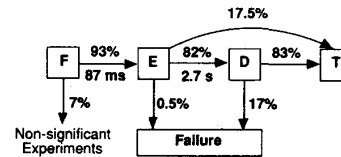

Fig. 7. Example of experimental graph.

is characterized by a predicate D. The resulting behavior of the noninjected nodes (the *nonfaulty nodes*) was also observed to assess the efficiency of the AMp mechanisms in tolerating the faults at the communication level. Correct operation of AMp is specified in terms of atomicity, order, and group membership properties that are globally characterized by a predicate T.

To carry out the test sequence, a *general distributed testbed,* featuring automatic control and sequencing of the experiments, as well as reset and recovery of the crashed nodes, was built around the fault injection tool MESSALINE [24]. This enabled us to carry out *extensive* fault injection experiments (almost *20000 experiments* of about *5-min duration* each) on a prototype of the Delta-4 architecture.

Faults in the F set were injected by *forcing* "zero" or "one" levels on the pins (up to 3 pins simultaneously)—and thus on the connected equipotential lines—of 86 IC's on the NAC board. To account for the most likely faults, the injected faults were mainly *intermittent* faults, but transient as well as permanent faults were also injected. Activation of the target system (the A set) consisted of two types of traffic exchanged among the nodes with various traffic profiles that ensured different activation modes for the injected node. Further details on the testing environment can be found in [36].

The experimental results obtained proved very useful in building a relevant model of fault tolerance behavior. Fig. 7 gives an example of values obtained for a typical experimental graph, which in fact corresponds to the predicate graph discussed earlier (Fig. 1).

The percentages indicate the values of asymptotic coverage for the predicates E (error), D (hardware error detection), and T (tolerance by the communication protocol). The time measures indicate the mean values of the fault dormancy and error detection latency distributions; only asymptotic coverage is considered with respect to the T predicate since such a predicate is of the type described in Fig. 2(a).

The main feature of this graph concerns the inclusion of transitions that might have been omitted from an *a priori* model of system behavior and thus also from the evaluation of the associated probabilities; two such transitions exist, which are related to 1) the identification of the injected faults that were not activated as errors, and 2) the inclusion of a transition between states E and T accounting for injected faults that were actually activated as errors but were apparently tolerated without being detected.

The *first transition* represents the experiments that are nonsignificant (i.e., experiments that cannot activate the tested FTAM's); relevant error-based coverage estimates can be obtained by processing only the readouts of the significant experiments. The results show that, thanks to the large proportion of intermittent faults injected and to the variety of activation
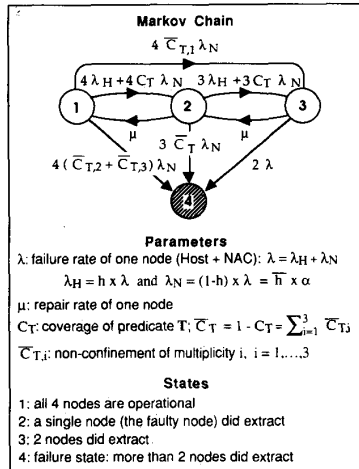
Fig. 8. Model of the target system.

modes applied, a very large proportion of experiments (i.e., 93%) were significant ones since the injected faults were actually activated as errors; this information was obtained by means of current sensors attached to each fault-injection device.

The T predicate coverage can be estimated more conservatively when the percentage associated with the *second transition* (E to T) is taken (pessimistically) to represent experiments that terminate with errors that have remained latent but could eventually lead to failure.

An experimental graph such as this, together with the experimental values obtained, serve as the basis for the system-level dependability evaluation sketched out in the next subsection.

B. Evaluation of Dependability Measures

To relate the dependability evaluation to the experiments that were carried out, we consider a system made up of four nodes, as in the case of the target system used for the fault injection test sequence. Such an architecture may for example correspond to the case of a system requirement for triplex task execution with a fourth node as a back-up in order to tolerate two consecutive faults.

Fig. 8 shows the Markov model that describes the behavior of this architecture. A proportion h of the total node failure rate is considered to be that of the host computer, the remaining $(1 - h)$ that of the NAC.

In the model, parameter C_T accounts only for the asymptotic coverage associated with the tolerance predicate T of the NAC's (see Fig. 7); as a high coverage majority voting decision is applied to the results of task replicas running on the host computers, the coverage of the faults in the host computers is considered here as perfect. The rate at which task replicas exchange results for voting is considered to be much greater than the mean time to node failure $(1/\lambda)$. Consequently, the host and the NAC fault-tolerance processes (activated respectively by the exchange of results for majority voting and execution of the underlying AMp protocol) are considered as instantaneous in comparison to the other model

parameters. Therefore, this model contains no parameters analogous to the α and ν parameters of Fig. 5.

The experiments that were carried out clearly revealed cases of nonconfinement of errors (i.e., some injected faults not only resulted in the fact that the faulty node did extract from the rest of the network, but also provoked the extraction of several nonfaulty nodes). The multiplicity of such multiple node extractions impacts the dependability behavior of the system; therefore, the model includes parameters $\overline{C}_{T,i}$ to account for the undesired extraction of i nonfaulty nodes.

The model assumes that it is possible to tolerate up to two simultaneous extractions. Although this assumption is valid for the redundant configuration considered here and it has been possible to obtain these figures in the case of our four-node testbed, this might not be true in practice for more complex configurations. This model can thus be considered as leading to an (optimistic) upper bound for dependability evaluation. It is also interesting to account for the (pessimistic) case when any multiple extraction results in total network failure. In practice, this can be achieved by simply transferring the rate associated with transition 1–3 to transition 1–4 on the model of Fig. 8.

The equivalent failure rate of the system described by the model of Fig. 8, normalized with respect to the failure rate of a single node, is

$$\frac{\Gamma}{\lambda} \approx \left[\frac{\mathrm{MTFFnetwork}}{\mathrm{MTFFnode}}\right]^{-1} \approx 4 \cdot (\overline{C}_{T,2} + \overline{C}_{T,3}) \cdot \overline{h}$$

$$+ \frac{8 \cdot \overline{C}_{T,1} \cdot \overline{h} \cdot \dfrac{\lambda}{\mu}}{2 + \dfrac{\lambda}{\mu}}$$

$$+ \frac{12 \cdot (1 - \overline{h} \cdot \overline{C}_T) \cdot \overline{h} \cdot \overline{C}_T \cdot \dfrac{\lambda}{\mu}}{3 + \dfrac{\lambda}{\mu}}$$

$$+ \frac{24 \cdot (1 - \overline{h} \cdot \overline{C}_T)^2 \cdot \left(\dfrac{\lambda}{\mu}\right)^2}{\left(3 + \dfrac{\lambda}{\mu}\right) \cdot \left(2 + \dfrac{\lambda}{\mu}\right)}. \tag{19}$$

When considering the more restrictive assumption, then the equivalent failure rate becomes

$$\frac{\Gamma}{\lambda} \approx \left[\frac{\mathrm{MTFFnetwork}}{\mathrm{MTFFnode}}\right]^{-1}$$

$$\approx 4 \cdot \overline{C}_T \cdot \overline{h} + \frac{12 \cdot (1 - \overline{h} \cdot \overline{C}_T) \cdot \overline{h} \cdot \overline{C}_T \cdot \dfrac{\lambda}{\mu}}{3 + \dfrac{\lambda}{\mu}}$$

$$+ \frac{24 \cdot (1 - \overline{h} \cdot \overline{C}_T)^2 \cdot \left(\dfrac{\lambda}{\mu}\right)^2}{\left(3 + \dfrac{\lambda}{\mu}\right) \cdot \left(2 + \dfrac{\lambda}{\mu}\right)} \tag{20}$$

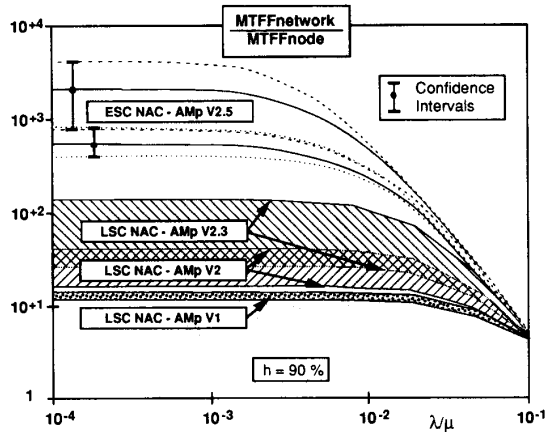| Systems	C_T	$\overline{C}_{T,1}$	$\overline{C}_{T,2}$	$\overline{C}_{T,3}$
LSC NAC — AMp V1	79.08%	2.32%	11.77%	6.83%
LSC NAC — AMp V2	85.02%	8.73%	2.80%	3.45%
LSC NAC — AMp V2.3	90.32%	7.79%	1.05%	0.84%
Upper Confidence Limit	*99.70%*	*0.51%*	*0.07%*	*0.26%*
ESC NAC — AMp V2.5	99.55%	0.32%	0.00%	0.12%
Lower Confidence Limit	*99.34%*	*0.21%*	*0.00%*	*0.06%*

Fig. 9. Experimental coverages.

Fig. 10. Dependability gain variations.

For the analysis, we consider the results obtained during the experiments concerning two distinct versions of the NAC hardware architecture: a NAC with only limited self-checking capabilities (LSC NAC) and a NAC featuring enhanced self-checking capabilities provided by a duplex architecture (ESC NAC). The fault injection experiments that were carried out—in particular on the LSC NAC (featuring a lower coverage)—had a significant impact in the debugging of the AMp software and several releases of the AMp software (denoted AMp $V_{x \cdot y}$) were therefore tested. The table in Fig. 9 summarizes the experimental measures considered for the analysis. More details on the experimental results can be found in [37] and [38].

The results obtained for the ESC NAC—AMp V2.5 configuration show a very appreciable improvement in the coverage. Out of the 4019 significant experiments that were carried out, only 18 faults were not tolerated; no nonconfinement of multiplicity 2 was observed. To provide a more objective estimation for this configuration, we have therefore considered confidence intervals for the coverage estimations. The percentages in bold characters give the nominal estimates; the figures in italics correspond to the upper and lower confidence limits for a 95% confidence level. Confidence intervals are not considered for the other configurations as their impact would be negligible due to the relatively low coverage values.

Fig. 10 compares the ranges of variations observed on the dependability gain measure for the configurations considered and for both optimistic and pessimistic assumptions.

The *upper* and *lower bounds* that define the areas shown for each configuration are obtained, respectively, from (19) and (20) when considering the *nominal coverage percentages* of Fig. 9; they can thus be considered as *nominal bounds*. Note that the areas associated with configurations LSC NAC–AMp V2 and LSC NAC—AMp V2.3 overlap.

The confidence limits for the coverage of configuration ESC NAC—AMp V 2.5 enable confidence limits to be obtained for these bounds; these limits appear as dotted lines. Note that the *lower limit* of the *upper bound* and the *upper limit* of the *lower bound* almost coincide.

The best nominal upper and lower bounds obtained for configuration ESC NAC–AMp V 2.5 indicate an MTFF improvement factor of 2000 and 500, respectively. However, the limits shown for each bound indicate how the uncertainty in the estimation of the coverages may affect these dependability predictions. As could be expected, the influence is stronger for the upper bound; the lower/upper limits are respectively 800/4000 for the upper bound and 400/800 for the lower bound. This shows that, even in the most conservative case, the Delta-4 architecture still provides a substantial dependability improvement.

Fig. 10 shows that the ESC NAC–AMp V2.5 combination provides almost one order of magnitude gain over the best results obtained for the LSC NAC architecture. This improvement can be attributed mainly to the improved self-checking mechanisms of the ESC NAC architecture rather than the change in AMp version since some partial tests using version 2.5 of AMp software on the LSC NAC were carried out and it was observed that there was no significant modification with respect to those obtained for version 2.3.

It should be pointed out that the curves shown here have been plotted for $h = 90\%$, that is, the proportion of node failure rate associated to the host computer. Although it is clear from (19) and (20) that parameter h impacts the absolute value of the gain, the sensitivity analysis with respect to h carried out in [36] has shown than for $h \leq 95\%$ (which covers the most realistic values of the ratio of host and NAC failure rates), the relative impact of the software and hardware modifications of the architecture shown in Fig. 10 is not significantly changed.

V. CONCLUSION

The dependability evaluation of complex fault-tolerant systems requires a combination of both experimental and analytical methods. This issue has been addressed by proposing a framework that establishes the link between the experimental evaluation of the coverage of the fault tolerance process and the fault occurrence process.

By investigating the relationship between the time distributions of the fault occurrence and coverage processes, we were able to show how it is possible to identify the relative domains where the time distribution has an impact on dependability measures.

The examples given clearly illustrate how the main interactions between model-based evaluation and experimental evaluation—namely, model characterization and coverage pa-

rameter calibration—fit within this framework and can be applied in practice.

The insights gained from the combined fault injection and dependability analysis carried out were regarded by the industrial partners of the Delta-4 project as providing very valuable aids in improving the designs and in making architectural decisions concerning the fault tolerance algorithms and mechanisms.

However, much work remains to be carried out towards the incorporation of fault injection at the various stages of the development and validation process of fault-tolerant systems. The results reported in this paper constitute only one facet of the work we are carrying out towards this goal. Other investigations include:

- the use of fault simulation as a complement to physical fault injection on a fault-tolerant system prototype,
- the identification of specific input patterns aimed at distinguishing the various error/fault processing actions of the fault-tolerance algorithms and mechanisms so that they can be adequately and efficiently verified,
- the clustering of the experimental results in order to refine the computation of the coverage estimators by accounting for significant differences in the operational fault occurrence rates associated with these clusters.

ACKNOWLEDGMENT

REFERENCES

[1] W. C. Carter and J. A. Abraham, "Design and evaluation tools for fault-lerant systems," in *Proc. AIAA Computers in Aerospace Conf.*, 1987, . 70–77.

[2] v. G. Bouricius, W. C. Carter, and P. R. Schneider, "Reliability modeling techniques for self-repairing computer systems," in *Proc. 24th Nat. Conf.*, ACM, 1969, pp. 295–309.

[3] J. B. Dugan and K. S. Trivedi, "Coverage modeling for dependability analysis of fault-tolerant systems," *IEEE Trans. Comput.*, vol. 38, no. 6, pp. 775–787, June 1989.

[4] D. Powell, "Failure mode assumptions and assumption coverage," in *Proc. 22nd Int. Symp. Fault-Tolerant Computing (FTCS-22)*, IEEE, Boston, MA, July 1992, pp. 386–395.

[5] Y. Crouzet and B. Decouty, "Measurements of fault detection mechanisms efficiency: Results," in *Proc. 12th Int. Symp. Fault-Tolerant Computing (FTCS-12)*, IEEE, Santa Monica, CA, June 1982, pp. 373–376.

[6] J. H. Lala, "Fault detection, isolation and reconfiguration in FTMP: Methods and experimental results," in *Proc. Digital Avionics Systems Conf.*, AIAA/IEEE, Nov. 1983, pp. 21.3.1–21.3.9.

[7] Z. Segall, D. Vrsalovic, D. Siewiorek, D. Yaskin, J. Kownacki, J. Barton, D. Rancey, A. Robinson, and T. Lin, "FIAT—Fault Injection based Automated Testing Environment," in *Proc. 18th Int. Symp. Fault-Tolerant Computing (FTCS-18)*, IEEE, Tokyo, Japan, June 1988, pp. 102–107.

[8] A. Damm, "Experimental evaluation of error-detection and self-checking coverage of components of a distributed real-time system," Doctoral dissertation, Technical Univ., Vienna, Austria, Oct. 1988.

[9] R. Chillarege and N. S. Bowen, "Understanding large system failures—A fault injection experiment," in *Proc. 19th Int. Symp. Fault-Tolerant Computing (FTCS-19)*, IEEE, Chicago, IL, June 1989, pp. 356–363.

[10] U. Gunneflo, J. Karlsson, and J. Torin, "Evaluation of error detection schemes using fault injection by heavy-ion radiation," in *Proc. 19th Int. Symp. Fault-Tolerant Computing (FTCS-19)*, IEEE, Chicago, IL, June 1989, pp. 340–347.

[11] E. W. Czeck and D. P. Siewiorek, "Effect of transient gate-level faults on program behavior," in *Proc. 20th Int. Symp. Fault Tolerant Computing (FTCS-20)*, IEEE, Newcastle upon Tyne, UK, June 1990, pp. 236–243.

[12] G. S. Choi, R. K. Iyer, R. Saleh, and V. Carreno, "A fault behavior model for an avionic microprocessor: A case study," in *Proc. 1st Int. Working Conf. Dependable Computing for Critical Applications*, Santa Barbara, CA, Aug. 1989, Avizienis and J.-C. Laprie, Eds. Vienna, Austria: Springer-Verlag, 1991, pp. 177–195.

[13] G. A. Kanawati, N. A. Kanawati, and J. A. Abraham, "FERRARI: A tool for the validation of system dependability properties," in *Proc. 22nd Int. Symp. Fault-Tolerant Computing (FTCS-22)*, IEEE, Boston, MA, July 1922, pp. 336–344.

[14] J. Arlat, Y. Crouzet, and J. -C. Laprie, "Fault injection for the experimental validation of fault tolerance," in *Proc. Esprit Conf. '91 (CEC-CGXIII)*, Brussels, Belgium, Nov. 1991, pp. 791–805.

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

[16] J. -C. Laprie, Ed., "Dependability: Basic concepts and terminology," in *Dependable Computing and Fault-Tolerance*, Vienna, Austria: Springer-Verlag, 1992, vol. 5.

[17] K. Echtle and Y. Chen, "Evaluation of deterministic fault injection for fault-tolerant protocol testing," in *Proc. 21st Int. Symp. Fault-Tolerant Computing (FTCS-21)*, IEEE, Montréal, Québec, Canada, June 1991, pp. 418–425.

[18] D. Avresky, J. Arlat, J.-C. Laprie, and Y. Crouzet, "Fault injection for the formal testing of fault tolerance," in *Proc. 22nd Int. Symp. Fault-Tolerant Computing (FTCS-22)*, IEEE, Boston, MA, July 1992, pp. 345–354.

[19] J. McGough, F. Swern, and S. J. Bavuso, "Methodology for measurement of fault latency in a digital avionic miniprocessor," in *Proc. AGARD Conf. 303, Tactical Airborne Distributed Computing and Networks*, Røros, Norway, June 1981, pp. 21.1–21.18.

[20] D. P. Siewiorek and R. S. Swarz, *The Theory and Practice of Reliable System Design*. Digital Press, 1982.

[21] C. J. Walter, "Evaluation and design of an ultra-reliable distributed architecture for fault tolerance," *IEEE Trans. Rel.*, vol. 39, no. 4, pp. 492–499, Oct. 1990.

[22] D. Powell, Ed., *Delta-4: A Generic Architecture for Dependable Distributed Computing*. Berlin, Germany: Springer-Verlag, 1991.

[23] D. Tang and R. K. Iyer, "Impact of correlated failures on dependability in a VAXcluster system," in *Proc. 2nd Int. Working Conf. Dependable Computing for Critical Applications*, Tucson, AZ, Feb. 1991, J. F. Meyer and R. D. Schlichting, Eds. Vienna, Austria: Springer-Verlag, 1992, pp. 175–194.

[24] J. Arlat, M. Aguera, L. Amat, Y. Crouzet, J.-C. Fabre, J.-C. Laprie, E. Martins, and D. Powell, "Fault injection for dependability validation—A methodology and some applications," *IEEE Trans. Software Eng.*, vol. 16, no. 2, pp. 166–182, Feb. 1990.

[25] J. Arlat, "Dependability validation by fault injection: Method, implementation, application," State Doctoral dissertation, National Polytechnic Inst. Toulouse, France, Dec. 1990 (in French).

[26] R. Geist, M. Smotherman, and R. Talley, "Modeling recovery time distributions in ultrareliable fault-tolerant systems," in *Proc. 20th Int. Symp. Fault-Tolerant Computing (FTCS-20)*, IEEE, Newcastle upon Tyne, UK, June 1990, pp. 499–504.

[27] W. Nelson, *Applied Life Data Analysis*. New York: Wiley, 1982.

[28] J. F. Lawless, *Statistical Models and Methods for Lifetime Data*. New York: Wiley, 1982.

[29] V. F. Nicola, M. K. Nakayama, P. Heidelberger, and A. Goyal, "Fast simulation of dependability models with general failure, repair and maintenance processes," in *Proc. 20th Int. Symp. Fault-Tolerant Computing (FTCS-20)*, IEEE, Newcastle upon Tyne, UK, June 1990, pp. 491–498.

[30] D. A. Rennels, "Some past experiments and future plans in experimental evaluations of fault tolerance," in *Proc. Int. Symp. Mini and Micro-Computers in Control and Measurement*, San Francisco, CA, 1981, pp. 91–98.

[31] E. Martins, "Validation of distributed systems by fault injection,"

Doctoral dissertation, ENSAE, Toulouse, France, June 1992 (in French).

[32] J. McGough, "Effects of near-coincident faults in multiprocessor systems," in *Proc. 5th Digital Avionics Systems Conf.,* AIAA/IEEE, Nov. 1983, pp. 16.6.1–16.6.7.

[33] J. Arlat and J. -C. Laprie, "On the dependability evaluation of high safety systems," in *Proc. 15th Int. Symp. Fault-Tolerant Computing (FTCS-15),* IEEE, Ann Arbor, MI, 1985, pp. 318–323.

[34] A. Pagès and M. Gondran, *System Reliability.* Paris: Eyrolles, 1980 (in French).

[35] D. Powell, G. Bonn, D. Seaton, P. Veríssimo, and F. Waeselynck, "The Delta-4 approach to dependability in open distributed computing systems," in *Proc. 18th Int. Symp. Fault-Tolerant Computing Systems (FTCS-18),* IEEE, Tokyo, Japan, June 1988, pp. 246–251.

[36] J. Arlat, M. Aguera, Y. Crouzet, J. Fabre, E. Martins, and D. Powell, "Experimental evaluation of the fault tolerance of an atomic multicast protocol," *IEEE Trans. Rel.,* vol. 39, no. 4, pp. 455–467, Oct. 1990.

[37] J. Arlat, Y. Crouzet, E. Martins, and D. Powell, "Dependability testing report LA2—Fault-injection on the fail-silent NAC: Preliminary results," LAAS-CNRS, Rep. 91043, Mar. 1991.

[38] _____, "Dependability testing report LA3—Fault-injection on the extended self-checking NAC," LAAS-CNRS, Rep. 91396, Dec. 1991.

Jean Arlat (M'80) was born in Toulouse, France, in 1953. He received the Engineer degree from the National Institute of Applied Sciences of Toulouse in 1976, and the Doctor in Engineering and Doctor ès-Sciences degrees from the National Polytechnic Institute of Toulouse in 1979 and 1990, respectively.

He is currently "Chargé de Recherche" at CNRS. He joined the research group on Dependabale Computing and Fault Tolerance at LAAS-CNRS in 1976. His research interests focus mainly on the evaluation of hardware-and-software fault-tolerant systems including both analytical modeling and experimental fault injection approaches, subjects on which he has authored or coauthored more than 30 papers. From April to September 1992, he was an Invited Visiting Professor at the Department of Computer Science of the Tokyo Institute of Technology holding a Toshiba Endowed Chair. He has also contributed to various National and European research contracts and acted as a consultant to several organizations in France.

Dr. Arlat has served as the Vice Chairman of the IEEE Computer Society's Technical Committee on Fault-Tolerant Computing since January 1992. In France, he is a member of the AFCET working group on Computing Systems Dependability and of the OFTA working group on Fault-Tolerant Computing.

Alain Costes received the Certified Engineer degree from the Higher National Engineering School in "E.N.S.E.E.I.H.T.," Toulouse, France, in 1963, and the Doctor in Engineering and Doctor es-Sciences degrees from the University of Toulouse in 1966 and 1972, respectively.

He is currently a Professor at the Polytechnic National Institute of Toulouse. He joined LAAS in 1963, where he is currently the Director. He has been involved with research in fault-tolerant computing in the group fault-tolerance and dependable computing, managed by Dr. Laprie since it was created.

From 1977 to 1983, Dr. Costes participated in the Directorate team of the French National Action for Systems Dependability as a scientific advisor of the agency for Informatics. From 1980 to 1985, he was Vice Chairman of Working Group 10.4 on Reliable Computing and Fault Tolerance of the International Federation for Information Processing. From 1989 to 1991, he was chairman of the section on Computer Science, Automatic Control, and Signal Processing of the French National Committee of Scientific Research. Since 1992 he has been a member of the section on Computer Science and Automatic Control of the same Committee, Chairman of the Council of Engineering Sciences at CNRS, and a member of the Scientific Council of CNRS.

Yves Crouzet was born in Toulouse, France, in 1952. He received the Engineer degree from the Higher National School of Electronics, Electrical Engineering, Computer Science, and Hydraulics, Toulouse, in 1975 and the Doctor in Engineering degree from the National Polytechnic Institute, Toulouse, in 1978.

He is currently "Chargé de Recherche" at the National Center for Scientific Research (CNRS). Since 1975 he has been a member of the Dependable Computing and Fault-Tolerance group at LAAS-CNRS. During 1975–1982 he worked on the design and realization of self-checking VLSI circuits. Since 1982 his research interests have concerned the experimental validation of dependable systems by fault-injection and the experimental validation of software testing methods by mutation analysis.
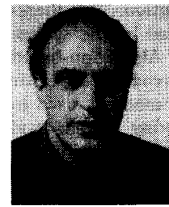
Jean-Claude Laprie received the Certified Engineer degree from the Higher National School for Aeronautical Constructions, Toulouse, France, in 1968, the Doctor in Engineering degree in automatic control, and the Doctor ès-Sciences degree in computer science from the University of Toulouse, in 1971 and 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 or coauthored more than 100 papers, as well as coauthored or edited 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 UCLA Department of Computer Science, Los Angeles. He has also acted as a consultant and as an expert in the area of dependable computing in France and abroad for government agencies as well as industrial organizations.

Dr. Laprie served in 1978 and 1993 as the General Chairman of the 8th and 23rd International Symposia on Fault Tolerant Computing, and on program committees for numerous conferences and workshops. He was Chairman of the IEEE Computer Society's Technical Committee on Fault Tolerant Computing in 1984 and 1985. He has been Chairman of the IFIP Working Group 10.4 on Reliable Computing and Fault Tolerance since 1986. In 1987 he founded the AFCET (French Association for Science and Technology of Information) Group on Computing Systems Dependability, which he chaired until 1991; he was recently elected to the Board of Directors of AFCET. He is coeditor of the Springer Verlag series on Dependable Computing and Fault Tolerant Systems. He is a member of ACM, AFCET, and the IEEE Computer Society.

David Powell (M'92) was born in Greenwich, England, in 1951. He received the Bachelor of Science degree in electronic engineering from the University of Southampton, England, in 1972.

He has been at LAAS-CNRS since 1972 and is a member of the Dependable Computing and Fault Tolerance Research Group. He obtained his Speciality and State Doctorates in 1975 and 1981, respectively. He is currently a "Directeur de Recherche" at CNRS. His current research work concerns the design and validation of fault-tolerant distributed computing systems. He has written over 50 papers for international and national journals and conferences, is coauthor of a book on dependable computing, and holds a patent for a fault and damage-tolerant network for data transmission.

Dr. Powell has managed several national and European research contracts and carried out consultancy work with several aerospace, telecommunication, and data processing companies in France. He was the Scientific Director of the Delta-4 project (1986–1991) of the CEC ESPRIT programme and is a member of AFCET.