Dynamic Aspects and Behaviors in System Reliability Evaluation

SALVATORE DISTEFANO\(^1\), ANTONIO PULIAFITO\(^1\) and KISHOR S. TRIVEDI\(^2\)

\(^1\)Dipartimento di Matematica, Università di Messina, Messina, Italy
\(^2\)Electrical and Computer Engineering Department, Duke University, Durham, U.S.A.

(Received on December 01, 2010 and revised on May 15, 2011)

Abstract: Reliability is one of the key attributes of dependability and quality of service. Techniques and tools for reliability assessment are therefore required in order to evaluate and to predict system behavior. In many contexts, merely taking into account of static system structure is not adequate and it is necessary to take into account dynamic aspects and behaviors. The focus of the paper is to first identify and characterize such dynamic aspects in reliability, and therefore to provide an overview of the technique that can adequately evaluate them.

Keywords: Dynamic reliability, dependencies, changing environment, resource sharing

1. Introduction

A system is a set of entities, an integrated composite of people, products, and processes that provide a capability to satisfy a stated need, pursuing a common objective. In performing its task, a system usually changes its internal state and evolves, thus identifying its own dynamics. The system dynamics can be characterized, qualified and quantified through specific quantities that are therefore, in general, variable in time. Among these performance and reliability indices are of great importance.

In order to investigate the behavior of a system and to provide adequate information on its dynamics, assessment of specific performance and reliability attributes is required. The models and the techniques used in such evaluations have therefore to take into account the real nature of the system and of its dynamics. In such context, we can consider the dynamics of a system, the variability of such attributes, as an effect of some specific triggering causes affecting it.

Considering such a cause-effect characterization of system dynamics, in this paper we wish to identify and classify aspects and behaviors that can affect the system dynamics, with specific reference to performance and reliability. Our goal is to provide some guidelines for representing and evaluating the performance and the reliability of a system specifically focusing on the dynamic aspects and behavior.

For this purpose, we use the concepts from system engineering of component and component-based systems, identifying a component as the elementary unit, the building block of a system, that therefore can be considered as an aggregate of components. In this way, we make a distinction among those aspects related to the structure of the system, the direct interconnections among its components due to specific and predetermined information exchange (datapaths, known I/O interactions, workflows, etc.), and the aspects related to the interactions among components as well as among the system and its
external environment, side effects such as dependencies, interferences, changing environments, and so on. We therefore categorize as static aspects and behaviors exclusively depending on the system structure, belonging to the first class, and as dynamic aspects and behaviors that belong to the second group (interference, dependence, changing environment, etc.).

In particular, in this paper we focus on analytical models mainly used off-line in order to design the system and to evaluate/predict the effects of possible modifications on its reliability. More specifically, starting from the above classification, we try to understand how this affects the choice of the solution technique exploited for evaluating the system performance and reliability.

Thus, in Section 2 we identify and classify static and dynamic reliability aspects, and then, in Section 3, we provide an overview of the technique dealing with such dynamic behaviors. Section 4 closes the paper with a brief review of its contents.

2. System Reliability Aspects Classification

Sometimes, system disasters are caused by neglecting the principles of redundancy and failure dependence (fault coverage models, dependent, on-demand, and common cause failures, etc.) that are obvious in retrospect [12]. More often, they are caused by considering over-simplistic models, coarsely approximating features that, instead, have to be adequately represented, for example dynamic and dependent behaviors.

In this way, from the component-based system perspective, we can classify aspects and behaviors affecting the system dynamics into two broad categories:

- **static**: related to structural direct/explicit relationships among components (datapath, workflow, feedback, etc.)
- **dynamic**: corresponding to indirect relationships among components, side effects such as interference, sharing, dependencies, changing environments, etc.

There are many real examples of dynamic-dependent behaviors in practice: load-sharing, standby redundancy, interferences, dependent, on-demand, cascade, and common cause failures, human factor, fault-coverage, growth, phased-mission systems, and so on. Thus, it is necessary to adequately represent such aspects through detailed models, exploiting specific modeling techniques. The crucial question such models must answer is whether catastrophic situations could have been avoided if the system was designed in an appropriate, reliable manner.

In case the model only contains static aspects, non-state-space models/notations such as reliability block diagrams (RBD) [11], fault/event trees (FT/ET) [1, 10, 13] and reliability graphs (RG) [10, 11] can be used for evaluating its reliability. If the stochastic independence assumption among components is not satisfied, all such techniques cannot be used, lower level techniques and formalisms are needed, each with specific characteristics that can better fit the modeling of some aspects of importance.

In the following we try to identify and to characterize the dynamic reliability aspects.

2.1 Dependent Events

Dependent events arise from interactions among the units composing a system. A classification of dependent failures, starting from the definition taken from [16], is summarized in Table 1.

One of the aims of this paper is to extend the concept of dependency to all the events characterizing a component or a system, not only failures. So, differently from Table 1 and [16], we discuss and deal with dependent, cascade, common cause and common mode
events, considering and extending it to any possible event affecting the component/system dynamics.

Table 1: Dependent, Cascade, Common Cause and Common Mode Failures Hierarchy

<table>
<thead>
<tr>
<th>Failure Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dependent Failure (DF)</td>
<td>Failure whose probability cannot be expressed as the simple product of the failure probabilities of the individual events that caused it.</td>
</tr>
<tr>
<td>Common cause failure (CCF)</td>
<td>Failure, which is the result of one or more events, causing coincident failures of two or more separate channels in a multiple channel system, leading to system failure (shared cause).</td>
</tr>
<tr>
<td>Common mode failure (CMF)</td>
<td>Common-cause failure in which multiple items fail in the same mode (shared cause and effect/mode).</td>
</tr>
<tr>
<td>Causal or cascade failure (CF)</td>
<td>All the dependent failures that are not common-cause, i.e., they do not affect redundant units.</td>
</tr>
</tbody>
</table>

2.2 Redundancy Management

Several possible ways to manage redundancy are possible: simple parallel, N-modular with voting gates, standby, and combinations of these. The simplest one is to replicate the units connecting them in a simple parallel configuration. Sometimes, specific QoS requirements impose that more than one redundant unit \( k \) out of \( n \) must be at the same time operating in order to have the system/subsystem in operational conditions \([1, 12]\).

In some cases, the same inputs are provided to each redundant unit, and the same outputs are expected from these. Therefore, the outputs of the redundant units are compared using a voting circuit or voter, that, anyway, could be a single point of failure (unless the voter is also replicated).

A more fruitful way to achieve higher reliability/availability levels is to implement standby redundancy policies \([1, 12]\). According to such policies, one or more redundant units are active in performing the system tasks, the others are kept in standby status as spare units, ready to replace the active units in case of failures.

2.3 Sharing Aspects

Sharing aspects usually refer to conditions in which there is a competition among two or more components for specific resources. Example of such conditions are load sharing and limited repair facilities.

The load sharing condition characterizes units that, in performing their tasks, share their workload. Since a less stressed unit is usually more reliable, it is clear that the reliability of a unit depends on its workload. So, in case of load sharing among two or more units, since the amount of workload managed by each unit depends on the number of units sharing the whole workload, such aspect can be represented as a mutually dependent behavior among the involved units \([2]\).

Another important aspect of system reliability and availability is maintenance. A maintenance strategy can be characterized according to the repair resources available. In literature this is known as the repairman problem \([1, 9]\). In the case of infinite resource the maintenance/repair policy can be represented by a cdf describing the time-to-repair random variable. Otherwise, it is necessary to evaluate the interactions due to simultaneous failures in the repairs of the corresponding units. A technique usually exploited is to group units identifying repair group sets associated to the corresponding repairmen set. In this way, a repair depends on the repairmen resources available and also on the number of simultaneous failures on a repair group. Such behaviors can be identified as complex interaction events among dependent units.
2.4 Changing Environment

In terms of reliability, a changing environment is a condition in which the external environment may change consequently affecting the system reliability. In such a class of problems two parameters can be considered in order to characterize the process: the number of different environment conditions that can be identified (two or more, but finite or denumerable) and the process that regulates the environmental changes. Due to its simplicity randomly changing environment is commonly dealt with in the literature [18].

A wide range of possible applications and examples of system reliability aspects that can be considered or are equivalent to a changing environment can be identified, such as dependencies, interferences, load sharing and so on.

3. System Reliability/Availability Evaluation

The main goal of system reliability study is the construction of a model (of system lifetime distribution) that represents the time-to-failure of the entire system, based on the lifetime distributions of the components, subassemblies and/or assemblies “black boxes” (units) from which it is composed [6]. Thus, in terms of (system) reliability, a system is identified by its units and corresponding relationships.

Reliability modeling aims to use abstract representation of systems as a mean to assess their reliability. Two basic techniques have been used: empirical or analytical techniques [11, 12]. According to empirical models, a set of $N$ systems is operated over a long period of time and the number of failed systems during that time is recorded. The percentage of failed systems out of the total number ($N$) of operated systems is the reliability parameter.

The analytical approach is based on the use of the failure probability of the individual units of a given system in order to provide a measure of the failure probability of the overall system. The system failure probability will not only depend on the failure probability of its individual units, but also on the way these units are interconnected in order to form the overall system. Analytical techniques are grouped into two classes: state-based and non-state-based models. The former (state-based models) are based on the concept of states and state transitions. System state is a combination of faulty and fault-free states of its units. State transitions show how the system changes its states as time progresses. So, in state-based models (Markov models, stochastic Petri nets, and variants) the states a system can assume, based on those assumed by its units, must be uniquely identified and numbered [10]. Such kinds of models can be used to evaluate different dependability metrics for a system. Unfortunately, it is very complex to analyze state-based models accurately, so this approach is not viable for many systems.

Applications of both Markov models to the reliability/availability modeling and analysis can be found for example in distributed, clustered and grid computing systems, also including several dynamic behaviors (standby redundancy, common cause failure, fault coverage models, load sharing configurations, multiple mode operations, etc.) [15]. Other examples of redundancy policies ($k/n$, TMR, N-modular, standby, etc.), software and network models, optimization techniques, and evaluation of the dependability of fault tolerant computing systems (RAID configurations, Tandem, Stratus, etc.) through Markov models can be found in [12].

Combinatorial models enumerate the number of ways a system can continue to operate, given the failure probability of its individual units. The system reliability is expressed as a combination of its units’ reliability, by exploiting the structure’s relationships equations [11]. While state-based models are more powerful and general
than combinatorial models, these latter are instead more user friendly, characteristic that motivates their success.

The most widely used combinatorial models are fault trees and reliability block diagrams. Static fault trees (FT) [13] use Boolean gates to represent how units’ failures combine to produce system failure. Dynamic trees [8] add a temporal notion, since system failures can depend on the order of unit failures. They can model dynamic replacement of failed units from pools of spares (CSP, WSP and HSP gates); failures that occur only if others occur in certain orders (PAND gates); dependencies that propagate the failure of one unit to others (FDEP gates); and specifications of constraints on the order of failures that simplify the analysis (SEQ gates).

In a reliability block diagram (RBD) [10, 11, 12], the logic diagram is arranged to indicate the combinations of properly working units keeping the system operational. RBD ensure interesting features in reliability modeling such as simplicity, versatility and expressive power. Such interesting features are inherited in a DRBD model, which moreover allows us to take into account the system dynamics [7].

Other interesting methodologies, techniques and tools to evaluate the reliability/availability of a dynamic system are: Boolean logic driven Markov processes (BDMP) [3], SHARPE [10], SMART [5], and so on. They combine high-level modeling formalisms such as RBD, FT, RG, failure propagation graph, etc. and low-level notation as Markov models, stochastic Petri nets, semi-Markov processes, (dynamic) Bayesian networks and so forth. On the other hand, there are many methodologies and tools with dynamics modeling capabilities based on single reliability modeling formalism, such as extended fault trees [17] and CPN with aging tokens [14] (Petri nets). They face the problem of modeling dynamic reliability behaviors in different ways. High-level techniques (DFT and similar) have the benefit to be closer to the modeler, sacrificing the generality and the expressive power to the ease in modeling. On the other hand, the techniques exploiting lower level/general purpose notation such as Markov models, Petri nets, stochastic process algebra and so on (SHARPE, SMART, BDMP, etc.), have greater expressiveness and power but lower user-friendliness than high level techniques [19].

4. Conclusions

This paper deals with dynamic reliability aspects and behaviors in component-based systems. Such aspects and behaviors are firstly identified and characterized and therefore some possible modeling and solution techniques are discussed. The main goal of this paper is to highlight the importance of adequately evaluating the system reliability while taking into account dynamic-dependent aspects and behaviors.

References


For biographies of the authors, please see page 403 of this issue.