TU Clausthal

Clausthal University of Technology

Towards More Comprehensive Measurable Dependability

Günter Kemnitz, Hossam A. Ramadan, and Carsten Giesemann

IfI Technical Report Series IfI-08-11

Department of Informatics
Clausthal University of Technology

Impressum

Publisher: Institut für Informatik, Technische Universität Clausthal
Julius-Albert Str. 4, 38678 Clausthal-Zellerfeld, Germany
Editor of the series: Jürgen Dix
Technical editor: Wojciech Jamroga
Contact: wjamroga@in.tu-clausthal.de
URL: http://www.in.tu-clausthal.de/forschung/technical-reports/
ISSN: 1860-8477

The IfI Review Board

Towards More Comprehensive Measurable Dependability

Günter Kemnitz, Hossam A. Ramadan, and Carsten Giesemann

Department of Informatics, Clausthal University of Technology
38678 Clausthal-Zellerfeld, Germany
http://techwww.in.tu-clausthal.de

Abstract

Dependability is an integrative concept that encompasses the following attributes: availability (readiness for correct service), reliability (continuity of correct service) and safety (absence of catastrophic consequences for the user(s) and the environment). In this paper we redefine these attributes. We are looking at them not only as concepts but as quantities. That makes it possible to measure or estimate them by experiments. The measurability makes the quantities more comprehensive and allows defining experiments to get values and to compare different solutions with each other.
Keywords: Dependability, Reliability, Safety, definition, redefinition, malfunction, decomposition, failure, classification, partial.

1 Introduction

Computer systems are characterized by five fundamental properties: functionality, usability, performance, cost and dependability [1]. The field of dependability grew out of previous related fields such as fault tolerance and system reliability in the 1960s. As interest in these fields increased during the 1970s and early part of the 1980s the term reliability began to be become overloaded and was being used outside of its originally intended definition, as a measurement of failures in a system, to encompass more diverse measures which would now come under other classifications such as safety, integrity, etc. [11]. Jean-Claude Laprie thus coined the term dependability to encompass these related disciplines in the early 1980. [8] The field of dependability has evolved from these beginnings to be an internationally active field of research. This research is fostered by a number of prominent international conferences, notably the International Conference on Dependable Systems and Networks, the International Symposium on Reliable Distributed Systems and the International Symposium on Software Reliability Engineering. The original definition of dependability [8] for a computing system gathers the following attributes or non-functional requirements:

- Availability: readiness for correct service.

- Reliability: continuity of correct service.
- Maintainability: to undergo modifications and repairs and combines them with the concepts of Threats and Failures to create Dependability.

This definition was further enhanced [1] to incorporate Safety and Security.

- Safety is the state of being "safe", it is absence of catastrophic consequences for the user(s) and the environment [1, 2].
- Security is the condition of being protected against danger. In the general sense, security is a concept similar to safety. The nuance between the two is an added emphasis on being protected from dangers that originate from outside.

A widely accepted characterization of dependability is: »the ability to deliver correct service that can justifiably be trusted«. The service delivered is its behavior as it is perceived by its user(s). The user is another system (physical, human) that interacts with the other at the service interface. A malfunction is an event that occurs when the delivered service deviates from correct service. Dependability encompasses the following attributes: availability, reliability and safety. Availability is readiness for correct service. Reliability is the continuity of correct service. Safety is the absence of catastrophic consequences for the user(s) and the environment [1, 2].

The question that can not be answered by these definitions is: How good are the dependability, the availability, the reliability and the safety? A yes/no-decision is not enough. Each complex computer system has unknown faults. Each fault may cause malfunctions or crashes, sometimes even with disastrous consequences [10]. Hardware may fail. One can never trust on a computer entirely, but only to a certain amount.

The paper will present redefinitions, recently published in a textbook by our group [7]. Some of the starting ideas have already been presented at the conference [6]. In the case of the reliability it has been redefined as the main time between malfunctions. The other quantities related to dependability are redefined in a similar way. It will be shown, that the new definitions will make the quantities more comprehensive and allow defining experiments to get values and to compare different solutions.

2 Reliability

What is reliability? We hear the term used a lot in research contexts, but what does it really mean? If you think about how we use the word "reliable" in everyday language, you might get a hint. For instance, we often speak about a machine as reliable: "I have a reliable car". Or, news people talk about a "usually reliable source". In both cases, the word reliable usually means "dependable" or "trustworthy". The term "Reliability" is defined in [8] as the continuity of correct service, but that's not a precise enough definition. The reason "continuity" is not a good enough description; we have to be a little more precise when we try to define reliability. It will be redefined as the main run

time between two malfunctions. A malfunction can be either a single wrong output or a sequence of wrong outputs, caused by a state error.

The reliability Z can be estimated by the ratio of the useful life time t_B and the number of malfunctions $\varphi_\triangleright$ observed during it:

$$Z \approx \frac{t_B}{\varphi_\triangleright}$$

The unit of measurement is hours or years.

The reliability of a computer system changes during its life time. As we can see in figure1, we can divide the life time of computing system into three main phases, a new untested system has often a low reliability. At the first run a large ratio of outputs is usually wrong. Before the system is usable, it needs time consuming iterations of tests and repairs. During these iterations, the number of faults decreases. The most common failures at this phase are Design failures, Infant Mortality failures, and the design failures take place due to inherent design flaws in the system. In a well designed system this class of failures should make a very small contribution to the total number of failures. On the other hand, Infant Mortality failures cause newly manufactured hardware to fail. This type of failures can be attributed to manufacturing problems like poor soldering, leaking capacitor etc. These failures should not be present in systems leaving the factory as these faults will show up in factory system burn in tests. The reciprocal process between Tests and repairs reduces the number of malfunctions and at the same rate it increases the reliability. When the acceptable level of reliability

$$Z \geq Z_{\min}$$

is reached, at this moment the first phase is completed (iteration of test and repair). The second phase is just starts (useful life), in which the system is handed over to the users. During usage the users will also experience malfunctions. Naturally they will look for workarounds, either asking the supplier for support or by looking for an input workaround. Input workaround means, that the users try to avoid operational conditions that cause difficulties in the future. Each removed fault and each input workaround reduces the frequency of malfunctions. The reliability increases, and the system matures. Hardware is subject to attrition [3]. Wires, semiconductor structures etc. are aging. It is always possible that a new fault arises, even such an event is very unlikely. If a new fault arises the reliability may drop dramatically (third phase). A drop below the acceptable level of reliability is called a failure. Such mistakes can be called Random Failures. After a failure, the system must be repaired or replaced before it can be used again, the time between the dropping and the re-rise moment is called repair time phase. Once a computing system has reached the end of its useful life (Wear Out), degradation of component characteristics will cause computing system to fail. This type of faults can be weeded out by preventive maintenance and routing of hardware. One can notice that the system in the second and third phase is a non-repairable system.

(ITR – iteration of test and repair; N – useful life; R – repair time; $\uparrow$ – fault removal; $\downarrow$ – failure)

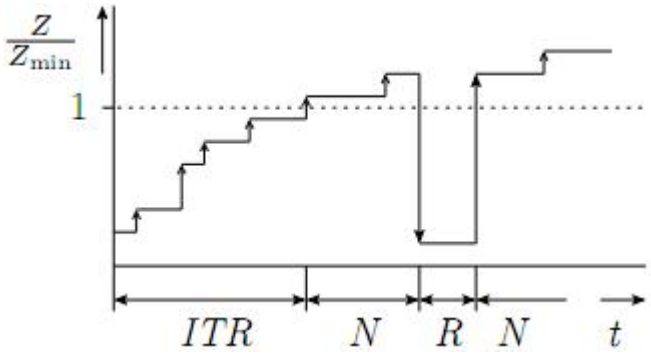

Figure 1: Change of reliability during life time

The reliability of a system can be split into parts. For this purpose the malfunctions are classified e.g. depending on:

- The cause (undetected fault, failure, operating error etc.).
- The duration (single wrong output, burst).
- The size of damage (negligible to critical).
- The affected location or system part.

Note that a single wrong output, a burst of wrong outputs and a system crash, of which the system can only recover by a reinitialization, are counted as single malfunctions.

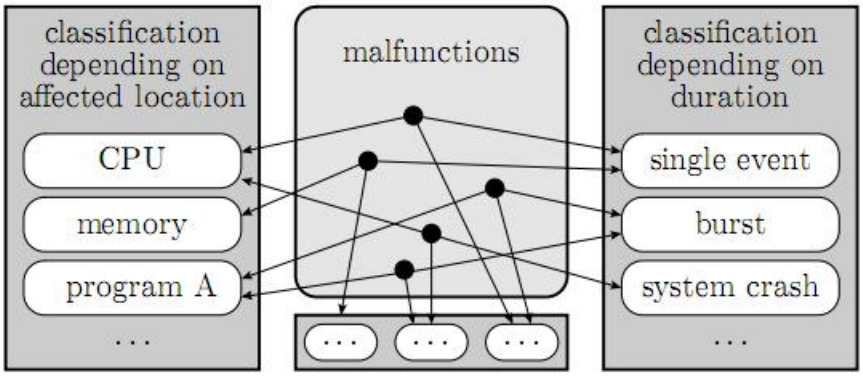

Figure 2: Different classifications of malfunctions

Return to our new redefinition, and with a consideration of non-overlapping mapping of malfunctions to malfunction classes, the total number of malfunctions is equal to the sum of the number of malfunctions of the individual classes:

$$\varphi_{\triangleright} = \sum_{i=1}^{N_{\mathrm{FK}}} \varphi_{\triangleright i} \tag{1}$$

(N_{FK} – number of malfunction classes; $\varphi_{\triangleright}$– total number of malfunctions; $\varphi_{\triangleright i}$ – number of malfunctions of class i).

The reliability is inversely proportional to the number of malfunctions. In the summary, the reciprocal value of the total reliability is the sum of the reciprocal values of the partial reliabilities due to the single malfunction classes:

$$Z^{-1} = \sum_{i=1}^{N_{\mathrm{FK}}} Z_i^{-1}$$

(Z_i – partial reliability due to malfunction class i).

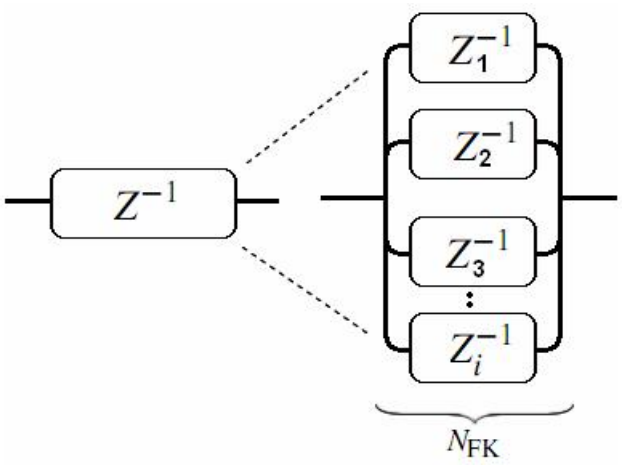

Figure 3: Total reliability due to partial reliabilities

The main advantage of this decomposition is that different aspects of reliability can be treated separately from each other. A decline of total reliability is described by a positive partial reliability because it increases the number of malfunctions. An improvement can be described by a negative partial reliability because it reduces the number of malfunctions. Fault tolerance, a subject to extensive research [4, 12, 8], could be described e.g. by a negative partial reliability.

At least one of the partial reliabilities has been used for a long time. It is the MTBF (main time between failures [5]). It is the partial reliability due to failures. The new definition is a generalization of an existing one, which takes into account that most malfunctions of current computer systems are not caused by failures but by other reasons.

3 Availability

Availability has been defined in [8] as the readiness for correct service. Another definition is the probability that the system is ready for correct service [5]. The slight difference is again, that a probability can be estimated by an experiment.

There are at least two reasons for unavailability that has to be treated differently:

- State errors: The system has crashed and can only recover by a reinitialization.
- Failures: An indispensable system part has failed and must be repaired or replaced.

One could think that the undetected faults are another reason for unavailability. However we consider only extensively tested systems with an acceptable level of reliability. System crashes caused by those faults are already considered and other fault related malfunctions affect only reliability.

Again the possible state errors and failures should be divided into classes, e.g. according to the necessary error handling (e.g. the part that has to be reinitialized, repaired or replaced). Every aspect of potential unavailability is described by a partial availability.

- $V_{\blacktriangledown.i}$ partial availability due to state error i.
- $V_{\blacklozenge.i}$ partial availability due to failure i.

Each cause of unavailability should be assigned only to one class and the components should fail independently of each other. The system is available, if it is affected by none of the causes. Though the total availability is product of all partial availabilities:

$$V = \prod_{i=1}^{N_{\blacktriangledown}} V_{\blacktriangledown.i} \cdot \prod_{i=1}^{N_{\blacklozenge}} V_{\blacklozenge.i} \tag{2}$$

($N_{\blacktriangledown}$ – number of state error classes; $N_{\blacklozenge}$– number of failure classes).

Availability is typically specified in nines notation. For example 3-nines availability corresponds to 99.9% availability. A 5-nines availability corresponds to 99.999% availability.

Downtime per year is a more intuitive way of understanding the availability. The table below compares the availability and the corresponding downtime.

The following example illustrates the usage of the model. Let us assume the following for a fictive computer system:

- The system consists of $N_{\blacklozenge} = 10$ components.

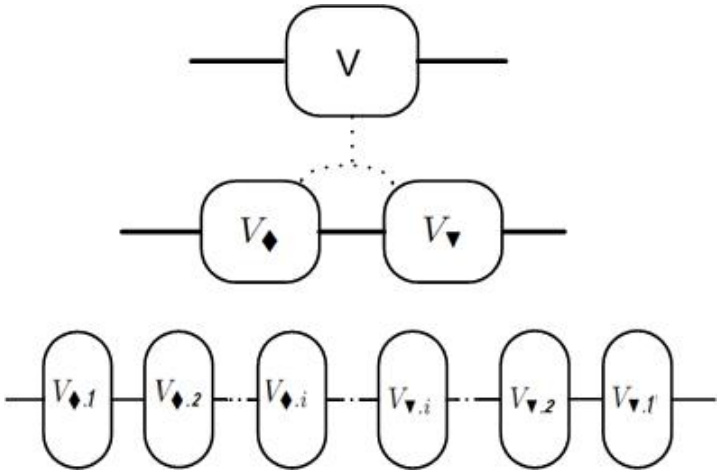

Figure 4: Total availability due to partial availabilities

Availability	Downtime
90% (1-nine)	36.5 days/year
99% (2-nines)	3.65 days/year
99.9% (3-nines)	8.76 hours/year
99.99% (4-nines)	52 minutes/year
99.999% (5-nines)	5 minutes/year
99.9999% (6-nines)	31 seconds/year !

Table 1: Compare between the availability and the corresponding downtime

- The probability that a component has failed and is still not repaired or replaced is 10^{-5}.
- The probability that the system has crashed and is not yet restarted is 10^{-3}.
- The probability that the system is not ready for use, because it eliminates an inconsistency in the data base is also 10^{-3}.

How large are the partial availabilities and the total availability?

In the example all possible failures in one component are merged to a component related failure class. The partial availability of each failure class is $V_{\blacklozenge.i} = 1-10^{-5}$. The number of different state error classes is two (crash and data base inconsistency). The partial availabilities are both $V_{\blacktriangledown.i} = 1 - 10^{-3}$. Using equation 2 the total availability is:

$$V = \left(1 - 10^{-3}\right)^2 \cdot \left(1 - 10^{-5}\right)^{10} = 99,79\%$$

This is also the order of magnitude of the availability of real computer systems.

Again at least one of the partial availabilities is in common usage. It is the partial availability due to failures, estimated by:

$$V_{\blacklozenge} = \frac{MTBF}{MTBF + MTR}$$

($MTBF$ - main time between failures; MTR – main time to repair [5]). Again the new definition is a generalization.

4 Safety

For some applications safety is more important than reliability [12]. Safety is the absence of catastrophic consequences on the users and the environment [1]. The areas characterized by the two terms safety and reliability are somewhat overlapping, as illustrated by fig.5.

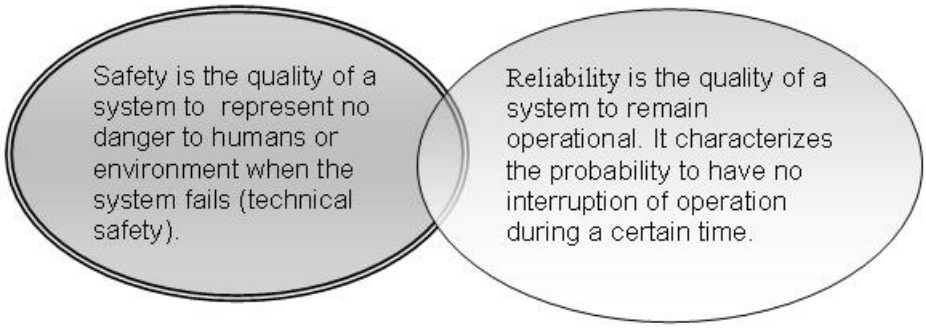

Figure 5: The overlapping between Safety and Reliability

what is interested in this overlapping that a safety and reliability of a system belong someway to each other. So the Safety will be redefined as the partial reliability due to the malfunctions causing disasters. It is the main useful time between two disasters caused by the system. The order of magnitude should be many years. In order to avoid disasters, it is mandatory that the safety is much higher than the useful life time.

Again the safety should be divided into partial safeties due to disaster classes, e.g. according to the system part or function causing the potential disaster or the handling in case, the disaster would happen. Because all partial safeties are also partial reliabilities, the reciprocal value of the total safety is the sum of the reciprocal values of the partial safeties due to the single disaster classes:

$$Z_{\dagger}^{-1} = \sum_{i=1}^{N_{\dagger}} Z_{\dagger i}^{-1}$$

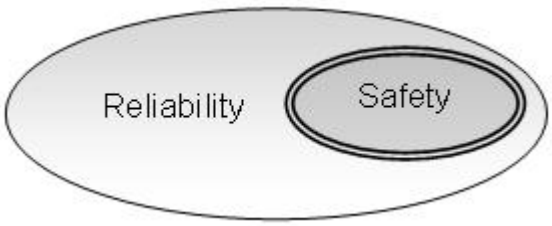

Figure 6: The Inclusion relationship between Safety and Reliability

($N_\dagger$ – number of disaster classes; $Z_{\dagger i}$ – safety of disaster class i).

Again the usefulness of the redefinition should be illustrated using fictive numbers. The exercise should be to estimate the minimum acceptable safety of a technical system. First a reference system will be needed. In case of a system, that may cause damage to the life and health of people, e.g. transport systems or medical devices, humans are the reference system. The partial safety of a person due to death cases is not larger than:

$$Z_{\dagger \mathrm{D}} < 10^2 \frac{\text{years}}{\text{death case}}$$

The technical system should improve the safety. An air bag e.g. should reduce the number of fatal injuries in car accidents. The safety increment $Z_{\dagger\uparrow}$ is a negative partial safety because it reduces the number of death cases that would happen otherwise. On the other hand, each technical system has a limited safety:

$$Z_{\dagger \mathrm{T}} < \infty$$

The total safety is:

$$Z_\dagger^{-1} = Z_{\dagger \mathrm{D}}^{-1} + Z_{\dagger\uparrow}^{-1} + Z_{\dagger \mathrm{T}}^{-1}$$

It should be improved by applying the technical system:

$$Z_\dagger > Z_{\dagger \mathrm{D}}$$

So, the safety of the technical system must be greater than the absolute value of the safety increment:

$$Z_{\dagger \mathrm{T}} > -Z_{\dagger\uparrow}$$

If the technical system may only cause but not avoid disasters, it is difficult to build it with an acceptable level of safety [9]. In this case the safety of the technical system must be much higher than the safety of the reference system:

$$Z_{\dagger \mathrm{T}} \gg Z_{\dagger \mathrm{D}}$$

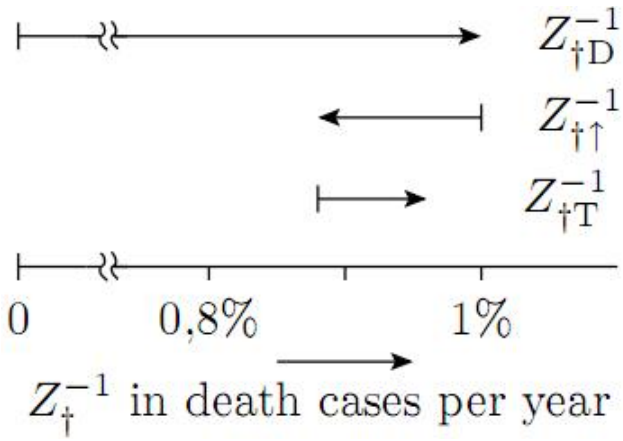

Figure 7: The estimation of the minimum acceptable safety of a technical system ($Z_{\dagger \mathrm{D}}$ – safety of the reference system; $Z_{\dagger\uparrow}$ – safety increment by the technical system; $Z_{\dagger \mathrm{T}}$ – safety of the technical system; $Z_{\dagger}$ – total safety)

A technical system, dangerous to the life and health of people must have a safety of thousands of years. To guarantee such a high amount of safety is very difficult. Philosophical reasoning on the logic of science shows that safety can only be improved, step after step. Again a model is presented, that allows quantifying all aspects or single factors of influence.

5 Conclusions

The attributes of dependability – reliability, availability and safety – have been redefined and generalized respectively in a way that they can be estimated by counting and classifying events and by measuring time. The events are malfunctions, observed by the user, and the time is the useful life time, the time to repair, the time to reinitialize etc.. The redefinitions allow describing the dependability of a system by a tuple of quantities instead of attributes. Though, the efficiency of the different means to attain dependability (fault prevention, test, fault tolerance etc.) can be quantified. Up to a certain amount they can be quantified even independently of each other.

In the text book [7] the redefinitions of the dependability attributes are used to describe the effect of design and manufacturing technology, the effect of test and repair etc. up to the effect of fault tolerance to the overall dependability of a system.

References

[1] A. Avizienis, J. Laprie, and B. Randell. Fundamental concepts of dependability. In *Research Report N01145, LAAS-CNRS*, 2001.

[2] G. Dewsbury, I. Sommerville, K. Clarke, and M. Rouncefield. A dependability model for domestic systems. In *SafeComp Conference*, 2003.

[3] P. B. J. A. R. Jayanth Srinivasan, Sarita V. Adve. The case for lifetime reliability-aware microprocessors. In *International Symposium on Computer Architecture (ISCA-04)*, 2004.

[4] J. C. C. L. Jean Arlat, Karama Kanoun. Dependability modeling and evaluation of software fault-tolerant systems. In *IEEE Transactions on Computers*, volume 39, pages 504 – 513, 1990.

[5] R. Kärger. *Diagnose von Computern*. Teubner, 1996.

[6] G. Kemnitz. Guardbands in random testing. In *Baltic Electronic Conference*, pages 85 – 88, 1996.

[7] G. Kemnitz. *Test und Verlässlichkeit von Rechnern*. Springer, 2007.

[8] J. Laprie. Dependable computing and fault tolerance: concepts and terminology. In *Digest of FTCS-15*, pages 2 – 11, 1985.

[9] N. G. Leveson. *Safeware: System Safety and Computers*. Addison-Wesley, 1995.

[10] B. Parhami. From defects to failures: a view of dependable computing. In *ACM SIGARCH Computer Architecture News*, volume 16, pages 157 – 168, 1988.

[11] B. Randell. Software dependability: A personal view. In *the Proc of the 25th International Symposium on Fault-Tolerant Computing (FTCS-25)*.

[12] W. Torres-Pomales. Software fault tolerance: A tutorial. In *NASA/TM-2000-210616*, 2000.