DOI: 10.5604/01.3001.0012.0704

MODIFIED, COMPLEMENTED TAXONOMY OF FAULTS IN FAULT-TOLERANT REAL-TIME SYSTEMS

Volodymyr Mosorov, Taras Panskyi, Sebastian Biedron
Lodz University of Technology, Institute of Applied Computer Science

Abstract*. This paper presents the main definitions relating to dependability. Basic definitions including reliability, security, maintainability, etc. are described first. They are then supplemented by additional definitions, which address to the threats of dependability (faults, errors, failures). Overlapping dependability standards, renumbering and integration can cause uncertainty when using of a certain definition. For this purpose, authors present complemented fault taxonomy for fault-tolerant real-time systems to eliminate inconsistencies and to unify existing fault taxonomies.*

Keywords: fault, taxonomy, classification, dependability

ZMODYFIKOWANA, UZUPEŁNIONA TAKSONOMIA USTEREK W TOLERUJĄCYCH AWARIE SYSTEMACH CZASU RZECZYWISTEGO

Streszczenie. *W artykule przedstawiono najważniejsze definicje dotyczące słowności. Podstawowe definicje w tym niezawodność, bezpieczeństwo, obsługiwalność, itp. opisane są w pierwszej kolejności. Następnie są one uzupełniane dodatkowymi definicjami, które odnoszą się do zagrożeń słowności (usterki, błędy, awarie). Nakładające się standardy słowności, renumeracja i integracja mogą spowodować niepewność przy korzystaniu z pewnych definicji. W tym celu autorzy przedstawiają uzupełnioną taksonomię usterek w tolerujących błędy systemach czasu rzeczywistego. Celem jest wyeliminowanie niespójności oraz unifikacji istniejących taksonomii usterek.*

Słowa kluczowe: usterki, taksonomia, klasyfikacja, słowność

Introduction

The problem of reliable computing is as old as the first computers appear which used electric switches, mechanical relays, vacuum tubes, etc. The era of modern computing began with a flurry of technical development before and during World War II. Computer systems while the early 40's were slowed by various problems, including relatively unreliable components, complex equipments, cumbersome operations, and component synchronization imperfections. The invention of the transistor could be considered as an important milestone of computer system reliability. However, much more strict and demanding reliability requirements were caused by the space program in the early 60's, as well as by other real-time safety-critical practical applications where human lives could be threatened by a computer system failure.

The concept of fault tolerance unifies different approaches to system reliability by means of testing, diagnosis, prediction, redundancy in hardware and software, etc. It emerged in the late 60's when more emphasis was given to reliability testing on component and system level. Moreover, the first reliability standards were created at that time, namely military standard 781, military handbook 217. The concept of fault tolerance in 80's became more formalized due to International Organization for Standardization and its stand-alone International Electrotechnical Commission and reached maturity with the formation of the IEEE Computer Society Technical Committee on Fault-Tolerant Computing in 1969 [1].

Nowadays, there are a various combinations of national and international standards, government organizations, professional societies which have promulgated a dizzying number of system dependability standards, guidelines, recommended practices, rapports and other frameworks [4]. However, the majority of standard define only the basic term of fault, errors or failure without indentations in their properties, types, and relationships.

This paper aims to give precise definitions characterizing the various types of faults that come into play when addressing the dependability and security of computing and communication fault-tolerant systems. Furthermore, article aims to complement and unify existing fault taxonomies to eliminate inconsistencies and overlapping terms.

1. Dependability in fault-tolerant real-time system

Many terms can be used informally to describe the desired result that a system performs without going wrong. Besides the reliability, dependability is one of the key and expected system requirements. The term dependability seems not to be clearly defined. Therefore different meanings are cited:

- The original definition of dependability is the ability to deliver service that can justifiably be trusted [6]. In a broad sense, dependability includes its related attributes namely, reliability, availability, safety as well as maintainability. Fig. 1 summarizes the relationship between dependability and its principal attributes.
- According to [4] dependability is a form of availability that has the property of always being available when required. It is the degree to which a system is operable and capable of performing its required operation at any randomly chosen time during its specific operating time, on condition that the system is available at the start of the period.

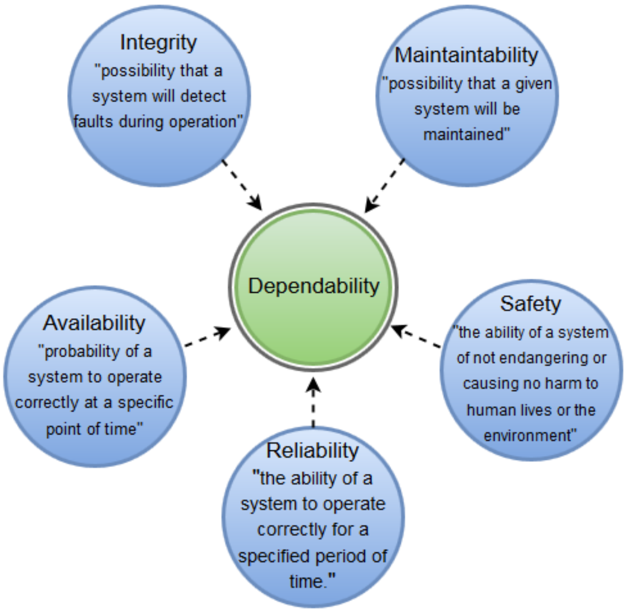
Integrity
"possibility that a system will detect faults during operation"
Maintainability
"possibility that a given system will be maintained"
Dependability
Availability
"probability of a system to operate correctly at a specific point of time"
Safety
"the ability of a system of not endangering or causing no harm to human lives or the environment"
Reliability
"the ability of a system to operate correctly for a specified period of time."

Fig. 1. Dependability and its attributes

Being fault tolerant is strongly related to what is called dependable system. A fault-tolerant real-time system is capable of performing the operations with satisfactory performance even if one or several faults, or more critically, one or several failures occur in this system [12]. Moreover, system is not only required to deliver correct results but also timely results. Thus, a system is dependable if it exhibits a *high* probability of behaving according to its specification. How high is it? This naturally depends on the purpose of the target system: the requirements of a life-supporting system and of a game console are quite different as well as nuclear power plant control system and light control system in a residential stairwell have completely not comparable requirement levels. The consequences of a failure are much more dramatic in life-supporting systems and in nuclear power plant control systems than in a gaming machine or in living illumination systems. The knowledge of the required degree of dependability entails the awareness of the impairments to dependability, i.e., the potential causes for incorrect behavior (faults, errors, failures) and the possible ways of their elimination.

2. Threats to dependability: faults, errors, failures

A fault-tolerant real-time system provides continuous, safe operation in the presence of faults. This system must detect errors caused by faults, assess the damage caused by the fault, recover from the error, and isolate the fault. The faults the system is to be designed to tolerate must be defined based on analysis of high requirements including the probability of each fault occurring, and the impact on the system performance in general [11].

In everyday language, the terms fault, failure, and error are used interchangeably. In fault-tolerant computing, however, they have distinctive meanings. A fault is an unpermitted deviation of at least one characteristic property (feature) of the system from the acceptable, usual, standard condition. A fault corresponds to an abnormal behavior of the system, which may not affect the overall functioning of the system but may eventually lead to a failure.

A failure is a permanent interruption of a system's ability to perform a required function under specified operating conditions. Resulting from one or more faults, a failure is therefore an event that terminates the functioning of a unit in the system or a system as a whole (critical failure).

An error is a discrepancy between a computed, observed or measured value or condition, and the true, specified or theoretically correct value or condition [9]. An error within a system may be caused by fault of one or more of its components, or by the activation of a systematic fault.

According to [5] the relationship between terms fault, failure and error is illustrated in Fig. 2. That is, an error leads to a failure event (unless the error is not removed), and the last leads to the fault state.

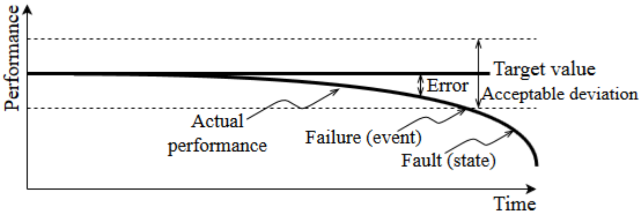

Fig. 2. The difference between failure, fault, and error

However, according to [8] an error may lead to a failure–a failure occurs when the error causes the delivered service to deviate from correct service. A fault is the cause of an error, and an error is the cause of a failure. The relationship between fault, failure and error is shown in Fig. 3.

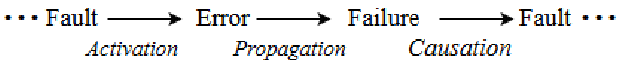

Fig. 3. The relationship between failure, fault, and error

Dependability as well as its threats differs in its meaning in some standards, e.g. MIL HDBK, IEC, DIS, etc. Organizations have different overlapping standards difficult to know which are applicable for a given situation or system. Therefore, many authors referring to the same standards confuse the reader. In this article, the authors tend to [8] explanation, where a fault is the cause of an error, and an error is the cause of a failure.

In any fault-tolerant real-time system, the range of potential fault that is quite large; enumerating all such possibilities is a vital yet formidable task in validating the system's readiness for deployment.

3. A taxonomy of faults

"A fault is an unpermitted deviation of at least one characteristic property (feature) of the system from the acceptable, usual, standard condition." [7]. Based on this definition, a fault corresponds to an abnormal behavior of the system, leading to the inability to perform as required, due to an internal state, which may not affect the overall functioning of the system but may eventually lead to a failure of one or several components. As suggested in [3] faults could be categorized in several ways according to eight basic viewpoints: phase of creation or occurrence, system boundaries, phenomenological cause, dimension, objective, intent, capability, persistence (duration). Additional viewpoints, namely extent and nature have been appeared in [4]. Classifications of faults in a tree form have been depicted in [10] with mode, domain and value viewpoints. Fig. 4 shows a complemented taxonomy of faults all aforementioned and suggested few extra viewpoints.

The new classification of faults includes several new viewpoints (classes). The classification of viewpoints is as follows:

1. The fault ***detection*** class indicates a capability of fault to be detected. Thus, faults could be distinguished as *detected* and *undetected*. Detected faults are subdivided into *targeted* and *accidentally detected* faults. Fault detection techniques are used to diagnose the presence of faults so that adequate countermeasures can be taken to prevent failures.
2. The ***simplicity*** of the faults:
   A *simple fault* is a fault that can be fixed by making a single change to a source statement. A complex fault is a fault that cannot be fixed by making a single change to a source statement. Terms simple and *complex faults* have never been formally defined, we introduce the working definitions only.
3. The ***sensitivity*** of the faults:
   According IEC 192-04-13 and IEC 192-04-14, *data sensitive fault* is a fault that is only activated when particular data are encountered. *Program sensitive fault* is a fault that is only activated when a particular sequence of program steps is executed. Generally these types of faults are for software only, but also could appear in hardware as well.
4. Two types of faults are considered relating to the ***correlation*** class: *independent* and *related* faults. Related faults result from a fault in a common specification or from dependencies in a separate design and implementation
5. The ***plurality*** class of the faults:
   Single fault is a fault caused by one adverse physical or one harmful human action. *Multiple faults* are two or more concurrent, overlapping or sequential single faults whose consequence, e.g. failures, errors, etc.
6. The ***style*** of the faults:
   An *omission fault* occurs of not doing something system should has done (the absence of actions when it should be) A *commission fault* occurs when a component generates incorrect results or when wrong actions are performed.
7. The ability to identify the activation pattern of a fault that had caused one or more errors is the ***fault activation reproducibility***. Faults can be categorized according to their activation reproducibility [2]:
   Faults whose activation is reproducible are called *solid* faults, whereas faults whose activation is not systematically reproducible are *elusive* faults.

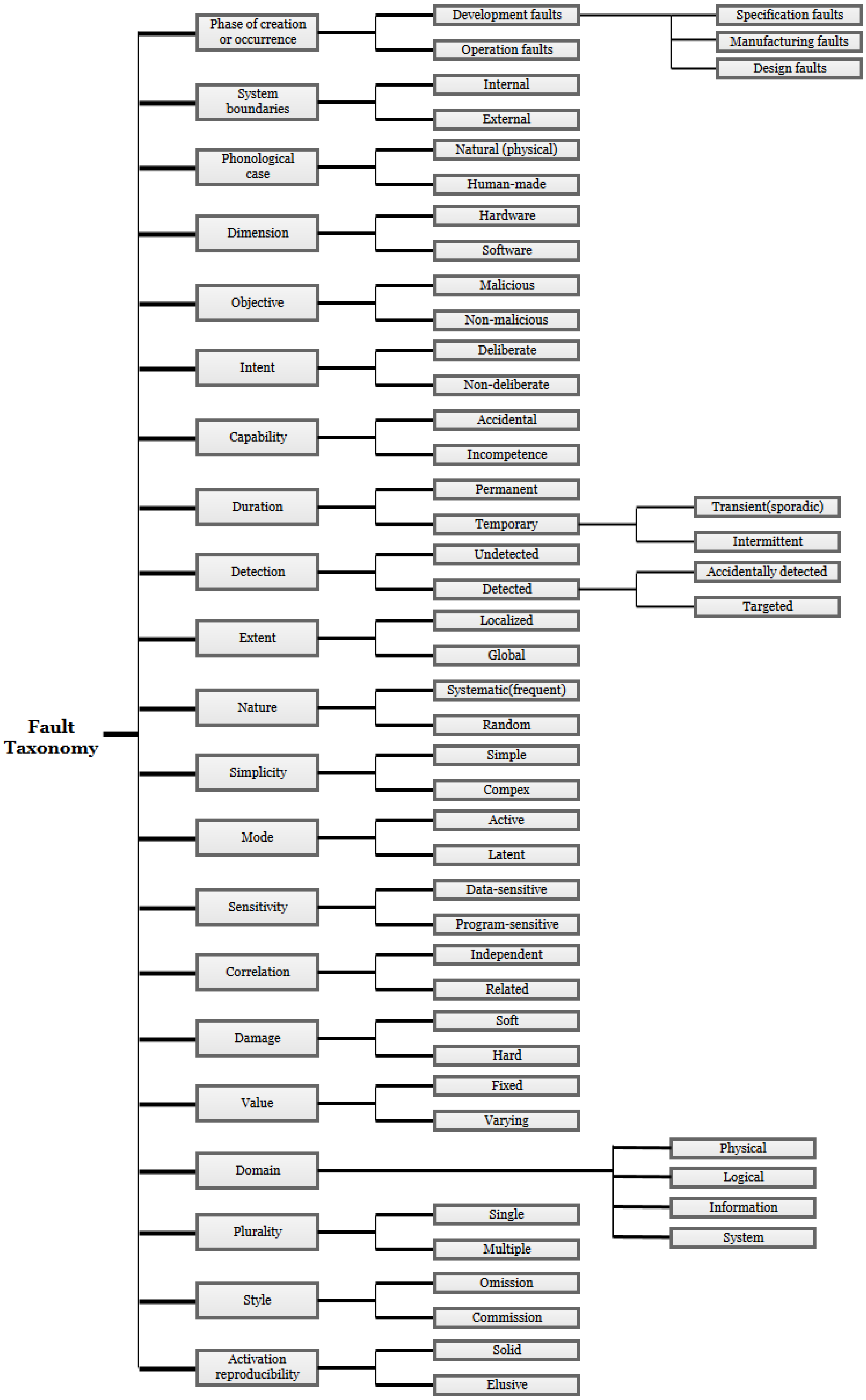

Fig. 4. Taxonomy of faults

4. Conclusion

Nowadays, developing sophisticated fault-tolerant real-time systems is required by public and private bodies. Simultaneous consideration of dependability provides a very convenient tool uniting various concerns within a single conceptual framework. Dependability includes such attributes as availability, reliability, safety, confidentiality, integrity, maintainability. However, despite the wide network of different national and international standards bodies the terminology differences are the largest potential problem. Due to consideration such as need for consistency within a set of standards, intended audience and conceptual organization, dependability as well as its attribute definitions differ in some standards, e.g. MIL HDBK, IEC and DIS. Organizations have different overlapping standards difficult to know which are applicable for a given situation. Moreover, the latest dependability standards do not outline even half of the presented fault classes. Therefore, the fault taxonomy aims to unify and to complement existing fault taxonomies to eliminate inconsistencies, renumbering and overlapping terms. Also, the taxonomy has been created to simplify the verbal description and to improve the adequacy of the models.

This article considers only elementary fault classes, however presented taxonomy does not include a complete picture of faults in fault-tolerant systems (for example authors do not consider the taxonomy of hardware or human faults). Moreover, complemented taxonomy only states the fact of its existence and does not cover the relationship between the presented fault classes.

References

[1] Avizienis A.: Fault-tolerant systems. IEEE Transactions On Computers, vol. 25, no. 12, 2006, 1304–1312.

[2] Avizienis A., Laprie J.-C., Randell B.: Dependability and Its Threats: A Taxonomy, in Building the Information Society. Springer Science + Business Media, 2004, 91–120.

[3] Avizienis A., Laprie J.-C., Randell B., Landwehr C.: Basic concepts and taxonomy of dependable and secure computing. IEEE Transactions on Dependable and Secure Computing, vol. 1, no. 1, 2004, 11–33.

[4] Bozzano M., Villafiorita A.: Design and safety assessment of critical systems. Auerbach Publications, 2010.

[5] Hoyland A., Rausand M.: System reliability theory: Models and statistical methods. John Wiley, 2009.

[6] Isermann R.: Fault-diagnosis applications. Springer-Verlag Berlin Heidelberg, 2014.

[7] Isermann R.: Fault-diagnosis systems. Springer-Verlag Berlin Heidelberg, 2006.

[8] Lee P. A., Anderson T.: Fault tolerance in Dependable Computing and Fault-Tolerant Systems. Springer Vienna, 1990.

[9] Meulen M.: Definitions for hardware and software safety engineers. Springer-Verlag London, 2000.

[10] Spitzer C.: Digital avionics handbook, Second edition – 2 volume set (electrical engineering handbook). 2nd ed. CRC Press, 2006.

[11] Tanenbaum A. S., van Steen M.: Distributed systems: Principles and paradigms. Prentice Hall, 2002.

[12] Veríssimo P., Rodrigues L.: Distributed systems for system architects in Advances in Distributed Computing and Middleware. Springer US, 2001.

D.Sc. Eng. Volodymyr Mosorov
e-mail: w.mosorow@kis.p.lodz.pl

Volodymyr Mosorov received his Ph.D. in 1998 from the State University of Lviv, Ukraine. V.Mosorov was awarded the title of Doctor of Science from AGH University of Science and Technology Krakow Poland in 2009. He is now an associate professor at the Institute of Applied Computer Science of Lodz University of Technology, Poland. His research interests include data mining and clustering. He has published more than 80 technical articles.

M.Sc. Eng. Taras Panskyi
e-mail: tpanski@kis.p.lodz.pl

Graduated from the Department of Theoretical Radio Engineering and Radio Measurement at Lviv Polytechnic National University, Ukraine. Since 2013, he has been a Ph.D. student at the Institute of Applied Computer Science of Lodz University of Technology, Poland. His research interests include data clustering, reliability and availability indexes of embedded systems, educational migration.

M.Sc. Eng. Sebastian Biedron
e-mail: sbiedron@iis.p.lodz.pl

Graduated from the Department of Science and Mathematics at Lodz University. Since 2012, he has been a court expert at the District Court at the Prague. Since 2013, he has been a Ph.D. student at the Institute of Applied Computer Science of Lodz University of Technology. The supervisor of his Ph.D. thesis is Volodymyr Mosorov, D.Sc. (dr hab. inż.), prof. PL.

otrzymano/received: 08.08.2016 *przyjęto do druku/accepted: 11.05.2018*