INTERNATIONAL JOURNAL OF RESEARCH IN COMPUTER APPLICATIONS AND ROBOTICS
ISSN 2320-7345

LITERATURE REVIEW ON DEPENDABLE AND SECURE COMPUTING

B.Kalpana[1], Dr. S. Uma[2]

[1] PG Scholar, PG CSE Department, Hindusthan Institute of Technology, Coimbatore, Tamil Nadu, India, kalpanabalu1982@gmail.com
[2] Head of the Department, PG CSE Department, Hindusthan Institute of Technology, Coimbatore, Tamil Nadu, India, umakarunahind@gmail.com

Abstract

Dependability of computer system plays a major role in the field of security. This paper elaborates the need for dependability and expresses a generic concept including some of the special attributes like reliability, availability, safety, confidentiality, integrity and maintainability. Dependability is also the system property that integrates such special attributes like safety, security and survivability. The major coverage aspect of this paper is to summarize the fundamental concepts of dependability. The view of dependability follows factors like threat, errors, failures, fault tolerance, fault prevention, fault forecasting and fault removal. This paper also discuss about the additional features of dependability with respect to trustworthiness. The aim of this paper also includes to explicate the specific concepts in the field of security.

Keywords*:* Dependability, Threat, Faults, Vulnerability

1. Introduction

The delivery of correct computing and communication services has beena concern of their providers and users since the earliest days. The first generation of electronic computers (late 1940's to mid-50's) used unreliable components, therefore realistic techniques were employed to improve their reliability, such as error control codes, depleting with comparison, triplication with voting, diagnostics to locate failed components, etc. An investigative research on the integration of fault tolerance and the defenses against deliberately malicious faults like security threats was started in the mid-80's. The protection and survival of complex information systems that are embedded as an infrastructure supporting world-wide concern of the priority. Increasingly, individuals and organizations are developing sophisticated computing systems to service a set of cash dispensers, control a satellite, an airplane, a nuclear plant, and radiation therapy device, to maintain the confidentiality of a responsive data bases. The focus will be on differing properties of such services and the ability to avoid failures that could be disastrous to the computer system's environment which should be prevented.

2. Basics of Dependability

The concepts of dependability are broadly classified into attributes and threat which is shown in the following Figure 1.

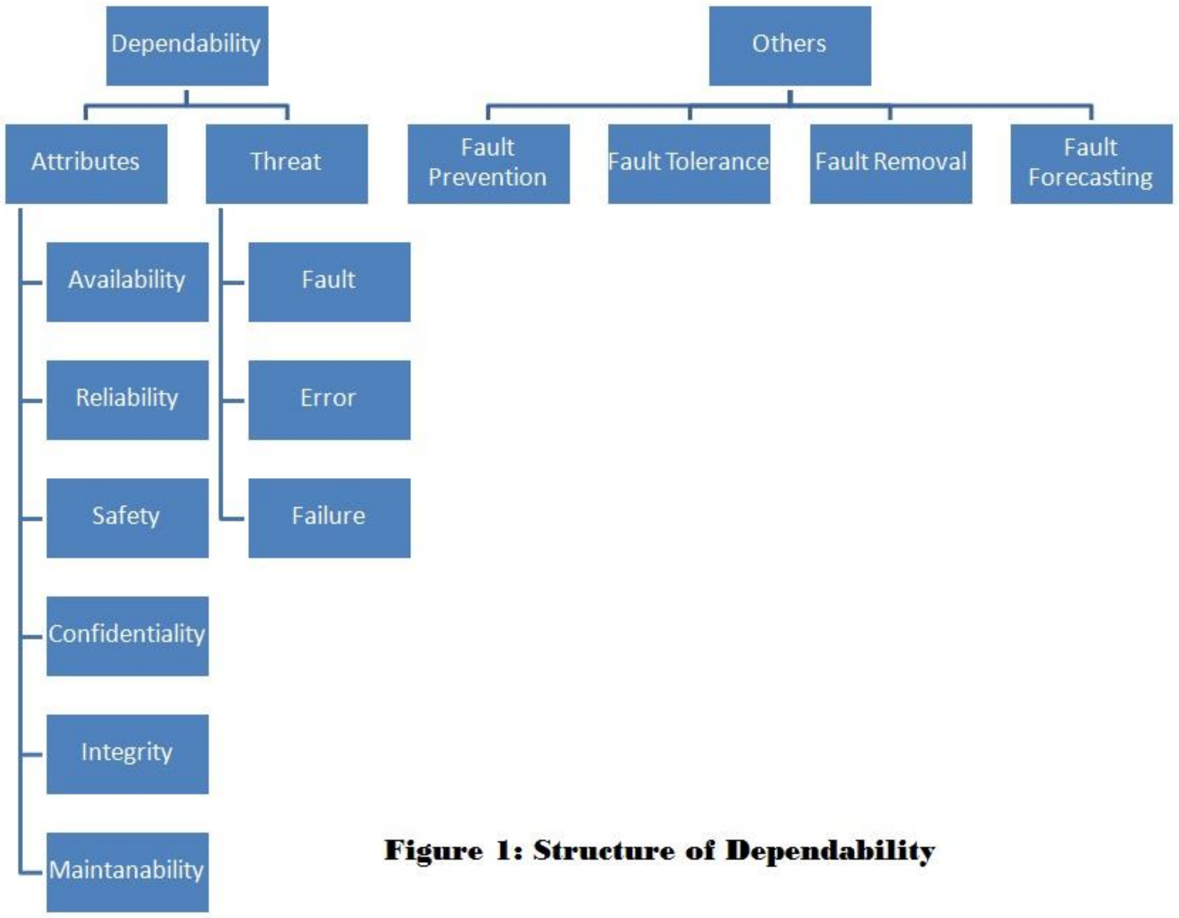

Figure 1: Structure of Dependability

Dependability is an important concept that consists of the following attributes:

- Availability: It specifies the readiness for correct service.
- Reliability: It specifies the continues availability of corrective service.
- Safety: It specifies the unavailability of disastrous consequences in the environment.
- Confidentiality: Unauthorized disclosure of data.
- Integrity: Absence of improper Service/system changes.
- Maintainability: Ability to experience modifications, and repairs.
- Security: It is the concurrent existence of
    - a) Availability for genuine users only.
    - b) confidentiality, and
    - c) Integrity.
- Dependability specification: A system must include the requirements for the dependability attributes in terms of the acceptable frequency and severity of failures for the specified classes of faults and a given use environment. One or more attributes may not be required for specifying dependability.

2.1 Dependability: A brief history

Dependability means an ability to deliver service that can justifiably be trusted in the user environment. It is the ability to avoid service failures that are morefrequent and more severe than it is acceptable. A system is an entity that interacts with other entities, like hardware, software, people, and the external world. These other systems are the environment [4] of the system. The system end points are the common frontier between the system and its associated environment. Computing and communication systems are characterized by four fundamental properties like functionality, dependability, and cost. These four properties are collectively influenced by two other properties namely usability and adaptability. The function of such a system is to explain what the system is intended to do and is described by the functional specification in terms offunctionality and performance. Dependability and cost has separate specifications. The behavior of a system is what the system does toimplement its function and is described by a sequence of states. The totalstate of a given system is the set of the following states: computation, communication, information and interconnection, and physical condition.

2.2 Dependability and System Functions

The part of the system boundary where service delivery takes place is called the service interface. The part of the provider's state is perceivable at the service interface. The delivered service is a sequence of the provider's external states. A system may sequentially or simultaneously be a provider or a user with respect to another system. A system generally implements more than one functionality, and delivers more than one services. Function and service are composed of service items. The structure of a system is what enables it to generate the behavior. A system is a set of components bound together in order to communicate with another system.

The behavior of the system depends on the context and the peripherals or the other system to which it is connected to. The behavior of the system is not static, it changes dynamically with respect to the number of factors like the type of external devices to which it is interfaced, number of users, switching on and off, etc. It also varies due to external interruptions and often it is unable to continue its operation from the place where it was interrupted. Hence it repeats the operations again and again from the beginning which leads to intolerance.

3. Threats encountered during secure computing

A service failure is an event that occurs when the delivered service deviates from service. A service fails when it does not comply with the functional specification. A service failure is a changeover from correct service to incorrect service, i.e., to not implementing the systemfunction. The period of delivery of incorrect service is a service overflow. The changes from improper service to a correct service is called as aservice restoration.The deviation from correct service may assume different forms that arecalled service failure modes and are ranked according to failure severities.

3.1 Service, Fault and Failures

Service: A service is a sequence of the system's external states; a service failure means that at least one (or more) external state of the system deviates from the correct service state. The variation is called an error [6].

Fault: The hypothesized cause of an error is called a fault. In most cases a fault initially causes an error in the service state. An error is the part of the total state of the system that may lead to service failure.

Failure: A fault is dynamic when it causes an error, otherwise it is dormant. The functional specification of a system includes a set of several functions that might leave the system in a degraded mode. The specification may identify such modes like slow service, limited service and emergency service.

4. Faults: A brief history

The faults that may affect a system during its life time are classified into broad basic viewpoints that are shown in Figure 2. These fault classes are called basic faults.

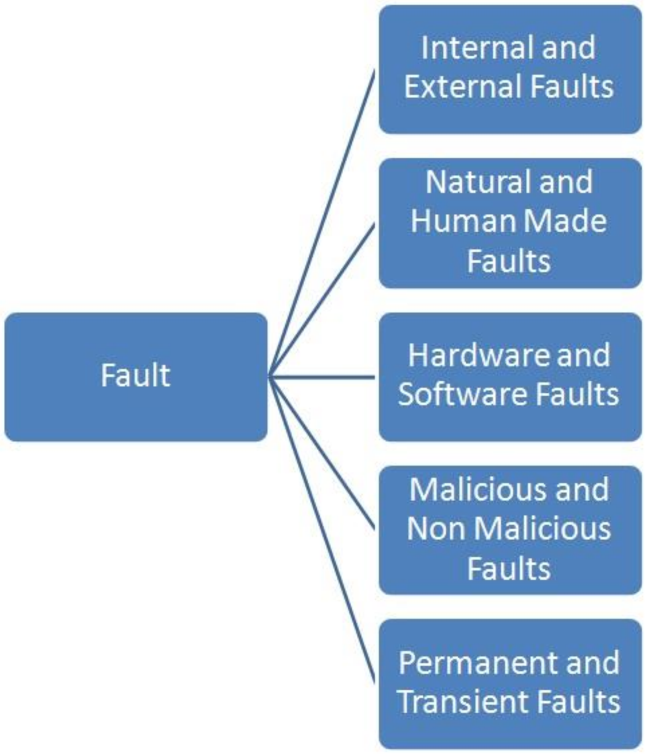
Fault
Internal and External Faults
Natural and Human Made Faults
Hardware and Software Faults
Malicious and Non Malicious Faults
Permanent and Transient Faults

Figure 2: Fault Classes

The broad classification criteria are as follows:

- The *phase* of system life during which the faults originate
  - Development faults that occur during system development.
  - Maintenance faults that occur during the use phase.
  - Operational faults that occur during service delivery of the use phase.

- The *location* of the faults with respect to the system boundary
  - Internal faults that originate inside the system boundary
  - External faults that originate outside the system boundary and propagate errors into the system by interaction or interference.

- The *phenomenological cause* of the faults:
  - Natural faults that are caused by nature without human participation
  - Human-made faults that result from human actions.

4.1 Human-Made Faults

The human-made faults are faults that results from inappropriate actions performed such as omission faults, or simply omissions. Performing wrong actions leads to omission faults. The two classes of human-made faults are:

1. ***Malicious faults***, which are introduced during either system development with the intent to cause harm to the system.
2. ***Non-malicious faults***, which are introduced without malicious objectives.

Malicious human-made faults are introduced by a developer with themalicious objective to alter the functioning of the system during its use. The goals of such faults are:

- To disrupt or halt service.
- To access confidential information.
- To improperly modify the system.

5. The Proactive Methods to attain Dependability

In order to attain the attributes of dependability the following four major categories are followed:

- **Fault prevention**: It means to prevent the occurrence /introduction of faults.
- **Fault tolerance**: It means to avoid service failures in the presence of faults.
- **Fault removal**: It means to reduce the number and severity of faults.
- **Fault forecasting**: It means to estimate the consequences of faults. **[8]**

Fault tolerance and prevention are used to provide the ability to deliver a service that can be trusted, while fault forecasting and fault removal are used to reach the confidence in that ability by justifying that the functional and dependability specifications are adequate and that the system is likely tomeet them.

5.1 Fault Prevention

Fault prevention is attained by quality control techniques that are employed during the design and manufacture of hardware and software. It includes data structured programming, information security, modular programming for software, and design rules for hardware. Shielding, radiation hardening, etc., intend toprevent operational physical faults, while training, rigorous procedures for maintenance.

5.2 Fault Tolerance

Fault tolerance is intended to preserve the delivery of correct service in the presence of active faults. Itis generally implemented by error detection and subsequent system recovery.Error detection originates an error signal or message within the system. An error that is present butnot detected is a latent error.

Error detection techniques are classified as follows:

- **Error detection** is one which takes place during service delivery.
- **Error detection** is one which takes place while service delivery is suspended and it, checks the system for errors and dormant faults.

A recovery of the system transforms a system state that contains one or more errors and faults into a state without detected errors and faults that can be activated again. A recovery consists of error handling and fault handling methodologies. It may take three forms:

- **Rollback** is where the state transformation consists of returning the system back to a saved state and the saved state is called as a checkpoint.
- **Compensation** is where the erroneous state contains redundancy to enable error elimination.
- **Roll forward** is where the state without detected errors is a new state.

Fault handling prevents located faults. The Fault handling involves four steps:

- **Fault diagnosis**, which identifies the cause(s) of error(s).
- **Fault isolation**, which performs physical or logical eradication of the faulty from delivery, i.e., it makes the fault dormant,
- **System reconfiguration**, which reassigns the tasks among non failed components.
- **System reinitialization**, which is used to check, updates and records the new configuration of the system tables and records.

5.3 Fault Removal

Fault removal is executed during the development phase, and during the operational phase of a system. Fault removal during the development phase of a system life-cycle consists of three steps: verification of process, diagnosis of system, correction.

- **Verification** is the process of checking whether the system adhere to its properties and conditions. If it does not, the other two steps follow.
- **Diagnosing** is that the fault(s) that prevented the verification conditions from being fulfilled, and then performing the acute corrections.
- **Correction**, the verification process should be repeated to check for fault removal. The proper checking of the specification is usually referred to as validation.

5.4 Fault Forecasting

Fault forecasting is performed by evaluation of the system behavior with respect to fault occurrence or activation. This type of evaluation has two aspects:

- Qualitative evaluation is one which aims to identify, classify and rank the failure modes that, would lead to system failures.
- Quantitative evaluation is one which aims to evaluate in terms of probabilities and attributes which are measured in terms of dependability.

The prominent measures of dependability are,

- Reliability is the measure of the continuous delivery of correct service.
- Availability is the measure of the delivery of correct service.
- Maintainability is the measure of the time to service restoration since the last failure
- Safety is an extension of reliability when the state is of correct service.

6 Conclusions

The concept of dependability provides a very convenient ways of describing the various concerns which includes some of the special cases such as reliability, availability, safety, integrity, confidentiality and maintainability. The major strength of the dependability concept relies on its attributes. It is a situated concept with accompanying work practices. It is not simply an inherent property of the system. It is a reflexive relationship between the work practices and the computing system. Hence the paper describes some of the important terminologies for the beginners who would like to do a research in dependable and secure computing. The paper also clarifies some of the atomic and meta-concepts of dependability.

REFERENCES

[1] A. Avizˇienis, 1967, “Design of fault-tolerant computers”, in *Proc. 1967 Fall JointComputer Conf., AFIPS Conf. Proc. Vol. 31*, pp. 733-743.

[2] A. Avizienis, J.-C. Laprie, B. Randell, and C. Landwehr, 2004, “Basic Concepts and Taxonomy of Dependable and Secure Computing”, IEEE Trans. on Dependable and Secure Computing, 1(1).

[3] W.H. Pierce, *Failure-Tolerant Computer Design*, Academic Press, 1965.

[4] Wilson, ÒThe STRATUS Computer SystemÓ, in Resilient Computing Systems, (T. Anderson, Ed.), pp.Ê208-231, London, UK, Collins, 1985.

[5] W.E. Baker, R.W. Horst, D.P. Sonnier, W.J. Watson, ÒA flexible ServerNet-based fault-tolerant architectureÓ, in Proc. 25th IEEE Int. Symp. on Fault-Tolerant Computing (FTCS-25), Pasadena, June 1995, pp. 2-11.

[6]N.S. Bowen, J. Antognini, R.D. Regan, N.C. Matsakis, "Availability in parallel systems: automatic processrestart", IBM Systems Journal, vol. 36, no. 2, 1997, pp. 284-300.

[7]C. Hennebert, G. Guiho, "SACEM: a fault-tolerant system for train speed control", Proc. 23rd IEEE Int. Symp.on Fault-Tolerant Computing (FTCS-23), Toulouse, France, June 1993, pp. 624-628.

[8]H. Kantz and C. Koza, ÒThe ELEKTRA Railway Signalling-System: Field Experience with an ActivelyReplicated System with Diversity,Ó Proc. 25th IEEE Int. Symp. on Fault-Tolerant Computing (FTCS-25),Pasadena, June 1995, pp. 453-458.

[9]D. Briere, P. Traverse, "Airbus A320/A330/A340 electrical flight controls Ñ a family of fault-tolerant systems",Proc. 23rd IEEE Int. Symp. on Fault-Tolerant Computing (FTCS-23), Toulouse, France, June 1993, pp. 616-623.

[10]Y.C. Yeh, ÒDependability of the 777 primary flight control systemÓ, in Dependable Computing for CriticalApplications 5, R.K. Iyer, M. Morganti, W.K. Fuchs, V. Gligor, eds, IEEE Computer Society Press, 1997, pp.3-17.

[11]J. Gray, "A census of Tandem system availability between 1985 and 1990", IEEE Trans. on Reliability, vol. 39,no. 4, Oct. 1990, pp. 409-418.

A Brief Author Biography

B Kalpana is an Assistant Professor and Post Graduate Scholar of PG Department of Computer Science and Engineering at Hindusthan Institute of Technology, Coimbatore, Tamil Nadu, India. She received her Bachelor of Technology Degree in Information Technology from Sri Venkateswara College of Engineering Chennai with First Class and Distinction affiliated to Madras University in 2003 and is currently pursuing M E degree (Part Time) at Hindusthan Institute of Technology Coimbatore affiliated to Anna University. She received her Diploma Degree in Computer Science from Panimalar Polytechnic Chennai with First Class and Distinction with a Gold Medal affliated to Directorate of Technical Education in 2000. She has several high-level involvements in the area of Artificial Intelligence and Big data. She has nearly 9 years of academic experience in the field of Engineering and guided many under graduate projects.

Dr S.Uma is Professor and Head of PG Department of Computer Science and Engineering at Hindusthan Institute of Technology, Coimbatore, Tamilnadu, India. She received her B.E., degree in Computer Science and Engineering in First Class with Distinction from PSG College of technology in 1991 and the M.S., degree from Anna University, Chennai, Tamilnadu, India. She received her Ph.D., in Computer Science and Engineering Anna University, Chennai, Tamilnadu, India with High Commendation. She has nearly 24 years of academic experience. She has organized many National Level events like seminars, workshops and conferences. She has published many research papers in National and International Conferences and Journals. She is a potential reviewer of International Journals and life member of ISTE professional body. Her research interests are pattern recognition and analysis of nonlinear time series data.