

Dependability foundations

Supporting reading

A. Avizienis, J.C. Laprie, B. Randell, C. Landwehr
Basic Concepts and Taxonomy of Dependable and Secure Computing
IEEE Transactions on Dependable and Secure Computing, Vol. 1, N. 1, 2004

Dependability: a definition

Dependability is “that property of a computer system such that reliance can **justifiably** be placed on the service it delivers”

If the system stops delivering the intended service, we call this a **failure**.

Trust in a computer controlled system could be justified through numbers that show the quality level of the system.
Such numbers are obtained using probabilities and statistical methods

For example in the avionic field, a rate of occurrence of failures of 10^{-9} was set as a design target

Dependability attributes

Dependability is a concept that encompasses multiple properties

- **Availability**
readiness for correct service

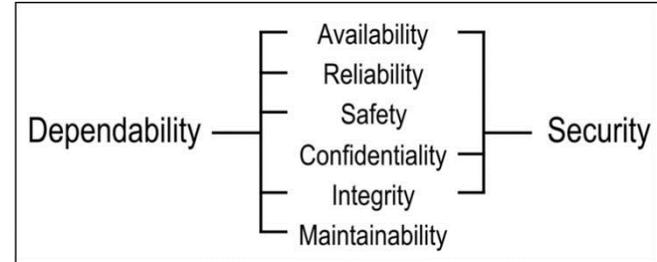
- **Reliability**
continuity of correct service

- **Safety**
absence of catastrophic consequences on the user(s)
and the environment

- **Confidentiality**
the absence of unauthorized disclosure of information

- **Integrity**
absence of improper system alterations

- **Maintainability**
ability to undergo modifications and repairs



What is a system?

System: entity that interacts with other entities, i.e., other systems, including

- hardware,
- networks,
- operating systems software,
- application software,
- humans, and
- the physical world with its natural phenomena.

These other systems are the environment of the given system.

The **system boundary** is the common frontier between the system and its environment.

Fundamental properties of a system:

functionality, performance, **dependability** and security, and cost.

System

Function of a system:

what the system is intended to do and is described by the functional specification in terms of functionality and performance.

Behavior of a system:

what the system does to implement its function and is described by a sequence of states.

Total state of a system:

is the set of the following states: computation, communication, stored information, interconnection, and physical condition.

Structure of a system:

what enables it to generate the behavior.

A system is composed of a set of components bound together in order to interact, where each component is another system, etc. The recursion stops when a component is considered to be atomic

The total state of a system is the set of the (external) states of its **atomic components**.

System

Service delivered by a system (in its role as a provider):
its behavior as it is perceived by its user(s)

A user is another system that receives service from the provider. The part of the provider's system boundary where service delivery takes place is the **provider's service interface**.

The part of the provider's total state that is perceivable at the service interface is its **external state**; the remaining part is its **internal state**.

The **delivered service** is a sequence of the provider's **external states**.

A system may sequentially or simultaneously be a provider and a user with respect to another system, i.e., deliver service to and receive service from that other system.

Use interface: the interface of the user at which the user receives service

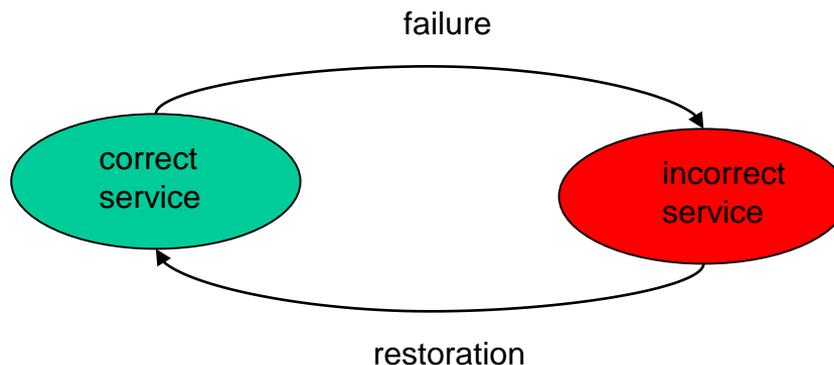
Threats to Dependability: Failures, Errors and Faults

Correct service is delivered when the service implements the system function.

A **service failure**, often abbreviated failure, is an event that occurs when the delivered service deviates from correct service. A service fails either because it does not comply with the functional specification, or because this specification did not adequately describe the system function.

Failure is a transition from correct service to incorrect service

Restoration is the transition from incorrect service to correct service



Threats to Dependability: Failures, Errors and Faults

A service failure means that at least one (or more) external state of the system deviates from the correct service state. The deviation of a state from the correct state is called an **error**.

The deviation from correct service may assume different forms that are called **service failure modes** and are ranked according to **failure severities**.

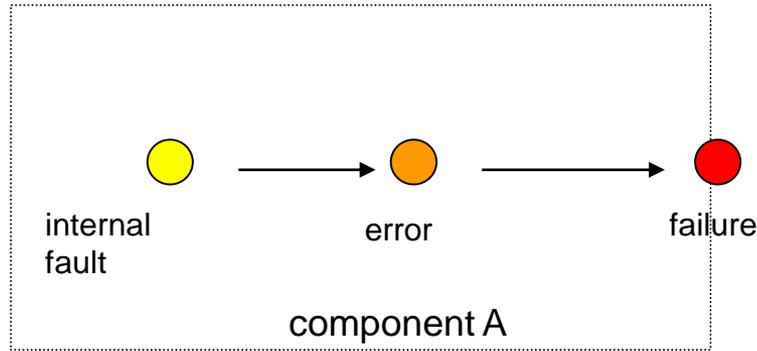
Fault: the adjudged or hypothesized cause of an error.

Faults can be **internal** or **external** of a system. The prior presence of a vulnerability, i.e., an internal fault that enables an external fault to harm the system, is necessary for an external fault to cause an error and possibly subsequent failure(s).

A **fault** first causes an error in the service state of a component that is a part of the internal state of the system and the external state is not immediately affected.

For this reason, the **definition of an error** is the part of the total state of the system that may lead to its subsequent service failure. It is important to note that many errors do not reach the system's external state and cause a failure.

Threats to Dependability: Failures, Errors and Faults



A fault causes an error in the internal state of the system.
The error causes the system to fail

Partial failure: Services implementing the functions may leave the system in a degraded mode that still offers a subset of needed services to the user. The specification may identify several such modes, e.g., slow service, limited service, emergency service, etc. Here, we say that the system has suffered a partial failure of its functionality or performance.

Means for achieving dependability

- A combined use of methods can be applied as means for achieving dependability. These means can be classified into:

1. Fault Prevention techniques

to prevent the occurrence and introduction of faults

- design review, component screening, testing, quality control methods, ...
- formal methods

2. Fault Tolerance techniques

to provide a service complying with the specification in spite of faults

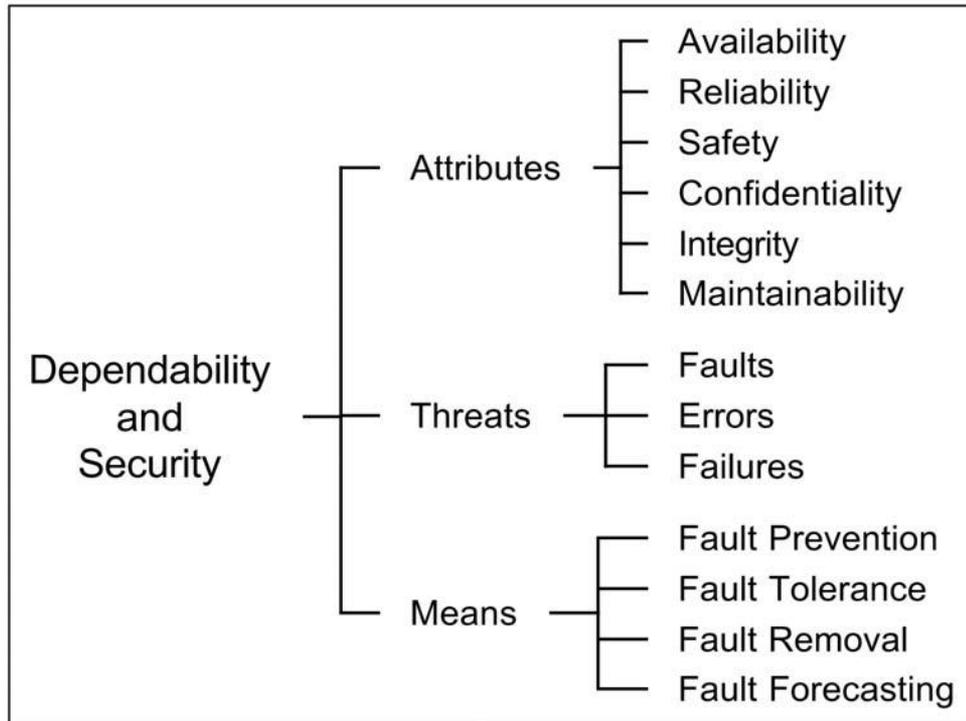
3. Fault Removal techniques

to reduce the presence of faults (number,seriouness, ...)

4. Fault Forecasting techniques

to estimate the present number, the future incidence, and the consequences of faults

Dependability tree



From A. Avizienis, J.C. Laprie, B. Randell, C. Landwehr. Basic Concepts and Taxonomy of Dependable and Secure Computing, IEEE Transactions on Dependable and Secure Computing, Vol. 1, N. 1, 2004

(* Security: Availability, Confidentiality, Integrity

System life cycle

The life cycle of a system consists of two phases:

- development
- use

Development phase includes all activities from presentation of the user's initial concept to the decision that the system has passed all acceptance tests and is ready to deliver service in its user's environment. During the development phase, the system interacts with the **development environment** and **development faults** may be introduced into the system by the environment.

The **development environment** of a system consists of the following elements:

1. the physical world with its natural phenomena
2. human developers, some possibly lacking competence or having malicious objectives
3. development tools: software and hardware used by the developers to assist them in the development process
4. production and test facilities

The **use phase** of a system's life begins when the system is accepted for use and starts the delivery of its services to the users.

The system interacts with its **use environment**.

System life cycle

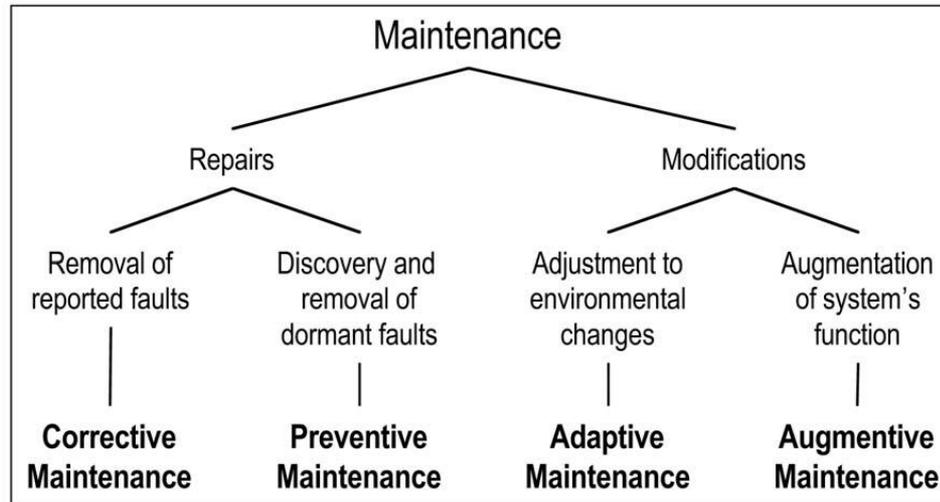
Use consists of alternating periods of **correct service delivery** (to be called **service delivery**), service outage, and service shutdown.

A **service outage** is caused by a service failure. It is the period when incorrect service (including no service at all) is delivered at the service interface.

A **service shutdown** is an intentional halt of service by an authorized entity.

Maintenance actions may take place during all three periods of the use phase. Maintenance, following common usage, includes not only repairs, but also all modifications of the system that take place during the use phase of system life.

Forms of Maintenance



From A. Avizienis, J.C. Laprie, B. Randell, C. Landwehr. Basic Concepts and Taxonomy of Dependable and Secure Computing, IEEE Transactions on Dependable and Secure Computing, Vol. 1, N. 1, 2004

Maintenance involves the participation of an external agent, e.g., a repairman, test equipment, remote reloading of software.

Furthermore, repair is part of **fault removal** (during the use phase), and **fault forecasting usually considers repair situations**. In fact, repair can be seen as a fault tolerance activity within a larger system that includes the system being repaired and the people and other systems that perform such repairs.

Types of faults

Failures may have many different causes or faults:

- chip suffers permanent electrical damage
- undersized fan (design fault) allows overheating on hot days
- Chip malfunction (physical fault)
- The machine works ok after cooling down (the fault is transient)
- Operator pushes the wrong button
- Cosmic ray particle causing transient upset in execution
- Defect in software

.....

Types of faults

Caused by what?

Physical faults

Human-Made faults

Why?

Accidental faults

Intentional non malicious faults / Intentional malicious faults

When?

Development faults: design, manufacturing, ...

Operational faults: in use or maintenance

Where (with respect to the system)?

Internal faults

External faults

How long?

Permanent faults

Transient faults