Defect, Fault, Error, ... , or Failure?

Behrooz Parhami, *Fellow IEEE*
University of California, Santa Barabara

Roderick Rees [1] expands on Ram Chillarege's commentary on software failure [2] by pointing out that "failure is a matter of function only [and is thus] related to purpose, not to whether an item is physically intact or not." I completely agree with Rees's view and made this same point in a 1994 survey paper [3] that contained a multi-level model for evaluating & counteracting the various causes of unreliability in computing and information systems.

After quoting a few relevant passages from the proposed model [3], three examples are used to show the relevance of such a viewpoint in a wider context.

" ... dependability of a computer system may be defined as justifiable confidence that it will perform specified actions or deliver specified results in a trustworthy and timely manner. ... [From the viewpoint of an end user who] is mainly concerned with triggered actions and computation results, ... a system can be in one of seven states (see figure 1): Ideal, Defective, Faulty, Erroneous, Malfunctioning, Degraded, or Failed. ... Briefly, a hardware or software component may be defective ... Certain system states will expose the defect, resulting in the development of faults defined as incorrect signal values or decisions within the system. If a fault is actually exercised, it may contaminate the data flowing within the system, causing errors. Erroneous information or states may or may not cause the affected subsystem to malfunction, depending on the subsystem's design and error tolerance. A subsystem malfunction does not necessarily have a catastrophic, unsafe, or even perceivable service-level effect. Finally, degradation of service could eventually lead to system failure. ... Initially, a system may start up in any of the seven states depending on the appropriateness and thoroughness of [specification and] validation efforts. Once in the initial state, the system moves from one state to another as a result of deviations and remedies. Deviations are events that take the system to a lower (less desirable) state, while remedies are measures that enable a system to make the transition to a higher state. As shown in figure 1, each state can be entered through the sideways transitions initially, from above due to a deviation, or from below as a result of a remedy." [3]

This model accommodates Roderick Rees's observation that some failures can result from unsatisfiable requirements rather than from any discernible malfunction. Examples 1 - 3 clarify the meaning of the states and state-transitions in figure 1. Example 2 is similar to example 1, but it better illustrates the lateral transitions of figure 1 and multi-level tolerance techniques. Example 3 illustrates both tolerance and avoidance techniques.

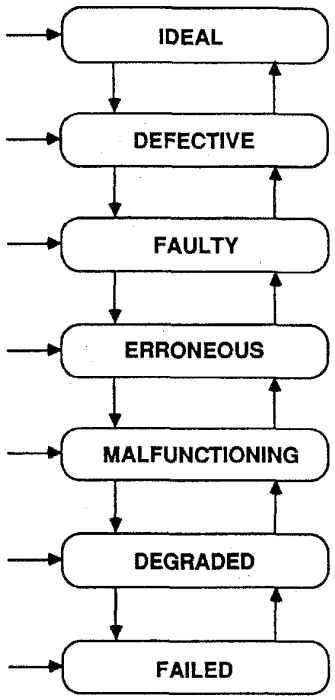

Figure 1. System States and State Transitions in the Multi-Level Model of Reliability

Example 1

An automobile brake system with a weak joint in the brake fluid piping (*eg*, caused by a design flaw or a road hazard) is *defective*. If the weak-joint breaks down, the brake system becomes *faulty*. A careful (off-line) inspection of the automobile can reveal the fault. However, the driver does not automatically notice the fault (on-line) while driving. The brake-system state becomes *erroneous* when the brake fluid level drops dangerously low. Again, the error is not automatically noticed by the driver, unless a working brake-fluid indicator-light is present. A *malfunctioning* brake system results from the improper state of its hydraulics when the brake pedal is applied. With no brake-fluid indicator light, the driver's first realization that something is wrong comes from noticing the *degraded* performance of the brake system (higher force needed or

0018-9529/97/$10.00 ©1997 IEEE

lower deceleration). If this degraded performance is insufficient for slowing down or stopping the vehicle when the need arises, the brake system has *failed* to act properly or deliver the anticipated result.

Example 2

Consider an automobile with one tire that has a weak spot on its road surface. The *defect* could be a result of corrosion or due to improper manufacture & inspection. Use of multiple layers or steel reinforcement constitute possible defect tolerance techniques. A hole in the tire is a *fault.* It could result from the weak-spot or could be caused directly by a nail. Low tire-pressure due to the hole, or directly as a result of improper initialization, is viewed as an *error.* Automatic steering compensation leads to error tolerance (at least for a while). A tire that is unfit for use, either due to its pressure dropping below a threshold or because it was unfit to begin with (*eg,* too small), leads to a *malfunction.* A vehicle with multiple axles or twin tires can tolerate some tire malfunctions. In the absence of tolerance provisions, one can still drive an automobile having a flat or otherwise unfit tire, but the performance (speed, comfort, safety, *etc*) is seriously *degraded.* Even a vehicle with several axles suffers performance degradation in terms of load capacity when a tire malfunctions. Finally, as a result of the above sequence of events or because someone forgot to install a vital subsystem, the entire automobile system can *fail.*

Example 3

Consider a small organization. *Defects* in the organization's staff promotion policies can cause improper promotions, viewed as *faults.* The resulting ineptitudes & dissatisfactions are *errors* in the organization's state. The organization's personnel or departments probably begin to *malfunction* as a result of the errors, in turn causing an overall *degradation* of performance. The end result can be the organization's *failure* to achieve its goals. Many parallels exist between organizational procedures and dependable computing terms such as:

- *defect removal* (external reviews),
- *fault testing* (staff evaluations),
- *fault tolerance* (friendly relations, teamwork),
- *error correction* (openness, alternate rewards),
- *self-repair* (mediation, on-the-job training).

For defect-induced failures, the sequence of transitions from defect to failure can be very slow, due to large inter-level latencies, or so quick as to defy detection. Ordinary inter-level latencies can be:

a. increased through tolerance provisions, or

b. reduced for making the deviations more readily observable (since deviations at lower levels of figure 1 are more easily detected).

The methods in #a are referred to as defect tolerance, fault tolerance, error tolerance, malfunction tolerance, degradation tolerance, and failure tolerance, while the methods in #b are useful for defect testing, fault testing, error testing, *etc.*

REFERENCES

[1] R. Rees, "What is a failure", *IEEE Trans. Reliability,* vol 46, 1997 Jun, p 163.

[2] R. Chillarege, "What is software failure", *IEEE Trans. Reliability,* vol 45, 1996 Sep, pp 354-355.

[3] B. Parhami, "A multi-level view of dependable computing", *Computers & Electrical Engineering,* vol 20, 1994, pp 347-368.

AUTHOR

Dr. Behrooz Parhami; Dep't of Electrical & Computer Engineering; University of California; Santa Barbara, California 93106-9560 USA.

Internet (e-mail): parhami@ece.ucsb.edu

Behrooz Parhami (S'1970, M'1973, SM'1978, F'1997) received his PhD (1973) in Computer Science from the University of California, Los Angeles. From 1974 to 1988, he was with Sharif University of Technology in Tehran, Iran. He was the Founding President of the Informatics Society of Iran (1979-84), served as Chair'n of the IEEE Iran Section (1977-1986), and received the IEEE Centennial Medal in 1984. He was elected a Fellow of the British Computer Society in 1995 and of IEEE in 1997. He is now a Professor at the University of California, Santa Barbara, where, in parallel with other activities, he is putting the finishing touches on his textbooks on Parallel Processing (Plenum Publishing Corporation, 1998) and Computer Arithmetic (Oxford Universiry Press, 1998).

Manuscript TR97-195 received 1997 August 14.
Responsible editor: R.A. Evans
Publisher Item Identifier S 0018-9529(97)09456-6