Dependable Systems

4. Chapter
Impairments

Prof. Matthias Werner
Operating Systems Group

4.1 Technical Terms

Failure / Fault / Error

- In general: Behavior differs from specification or expectation
- **Impairment**: Holistic view, considers also load, weaknesses etc. .

Definitions by LAPRIE

- **Failure:** Occurs when the delivered service deviates from the specified service. Failures are caused by errors
- **Error:** Derivation of the system's state from expected state (program or data); deviation from the expected result of computation (incorrect result) .
- **Fault:** Adjudged or hypothesized cause of an error

* Beachten Sie bitte, dass in der deutschen (Umgangs-)Sprache alle Konzepte mit “Fehler” bezeichnet werden.

Characterization of Faults

High diversity in possible sources and types

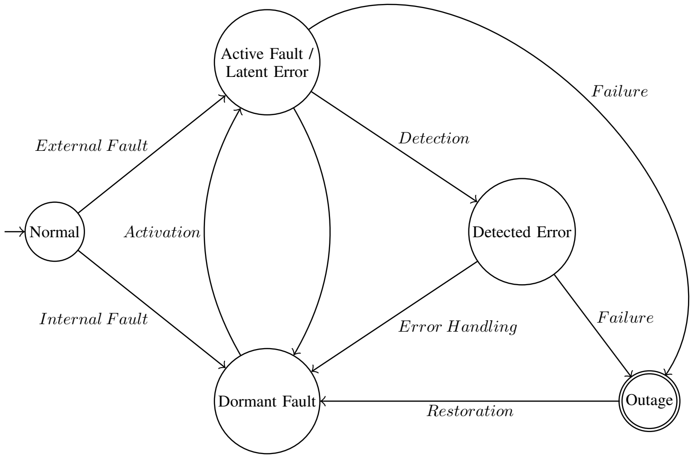
- **Fault nature:**
  - **Accidental** faults vs. **intentional** faults
- **Fault origin viewpoints**
  - Phenomenological causes: **physical** faults vs. **human-made** faults
  - System boundaries: **internal** faults vs. **external** faults
  - Phase of creation: **design** faults vs. **operational** faults
- **Temporal persistence:**
  - **Permanent** faults vs. **temporary** faults

Temporal Characterization

- A fault is **active** when it produces an error
- A non-active internal fault is a **dormant** fault
  ➡ often cycling between dormant and active
- Temporary external accidental physical faults are also called **transient faults**
- Temporary internal accidental faults are also called **intermittent faults**
- **Examples**:
  - Pattern-sensitive memory hardware, system overload
  - Arbitrary concept-dependent faults with unknown activation condition

Fault Automaton

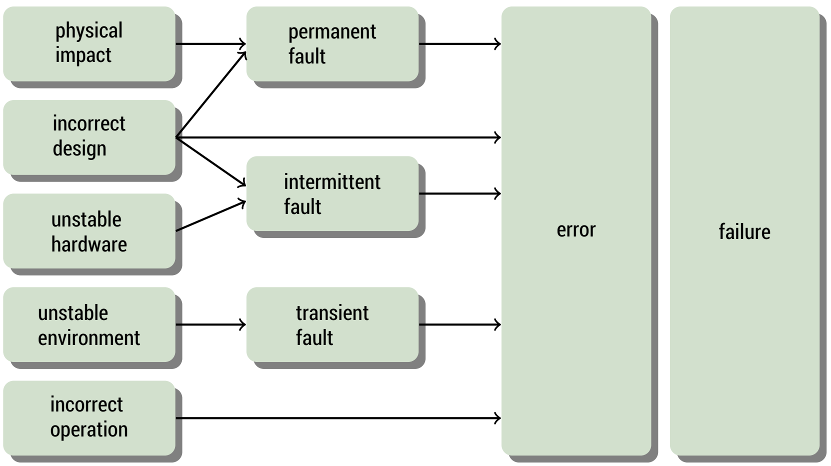
- The temporal relations can be described by a **fault automaton**



Discussion

- An external fault is a design fault – inability or refusal to foresee all situations
  - **Example**: performance or timing faults (derivation from expected load / timing)
- Design faults are created during system development, system modification, or operational procedure creation and establishment
- Physical faults are **accidental faults**
- Intentional and design faults are human-made faults
- Many specialized versions of the term “fault”, e.g. **bug**
  - **Heisenbug** – Intermittent software fault
  - **Bohrbug** – Permanent software fault
  - **Mandelbugs** – Appear chaotic due to many dependencies

Sources of Error



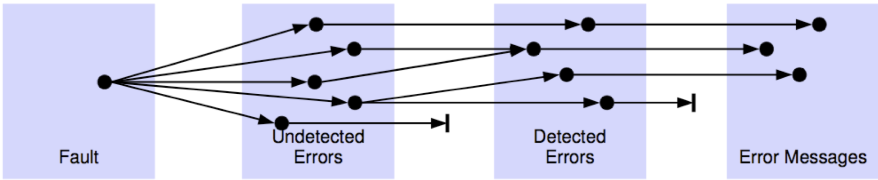
(Siewiorek/Swarz)

Errors

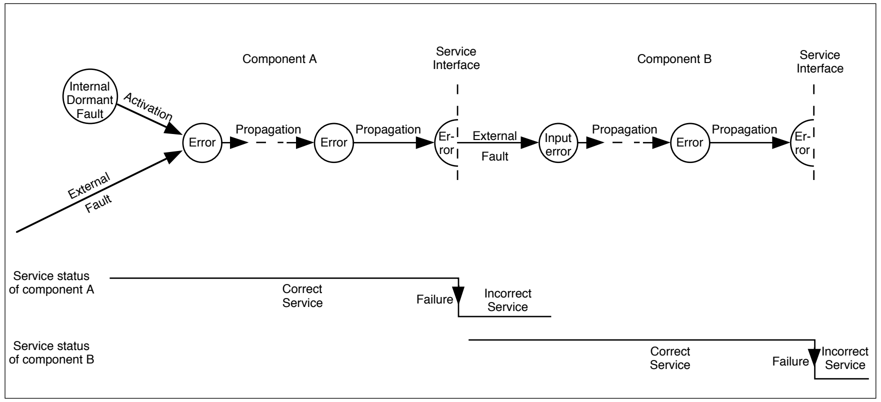
- State of the system, not an event
- Escalates to failure depending on
  - Intentional / unintentional redundancy
  - System activity
  - User‘s definition of a failure
    Examples: maximum outage time, acceptable delay, retransmission rate
  - Latent (not recognized) vs. detected error coming from an active fault

Error Messages (HANSEN & SIEWIOREK)

- Same fault can lead to different errors
- Detected errors might not be logged



Propagation



- **Source:** Algirdas Avizienis et al. “Basic Concepts and Taxonomy of Dependable and Secure Computing”. In: *IEEE Transactions on Dependable and Secure Computing* 1.1 (2004), pp. 1–23

Fault Failure Model

Alternative view: KOPETZ

No states are considered, but events only

- **fault:** undesired event
- **failure:** event of derivation in function; may constitute *fault* in the next level of abstraction

4.2 Impairment Models

System and Fault Model

- **Fault tolerant system design**: Seems to be a contradiction in terms:
  - Design requires specification (what to expect)?
  - Faults are deviations from specification
- **Solution**: Specification for fault free case + additional fault
- **Caution:** Interdependencies between system model and fault specification ➡ **fault model**

Coverage

Coverage

Coverage of a fault model is the ratio of the number of faults that can be described by the model to the number of faults that occur in reality.

- Used to evaluate a test's performance
- Tests are always based (at least **implicitly**) on a specific fault model
- Usually one fault model is evaluated against another that covers more faults
- The actual coverage can only be derived empirically

Fault Models

Fault models can be described at different levels of abstractions:

- (Physics (unusual))
- Circuit level
- **Switching [circuit] level**
- Register transfer level
- PMS-level (processor-memory-switch)
- **System level**

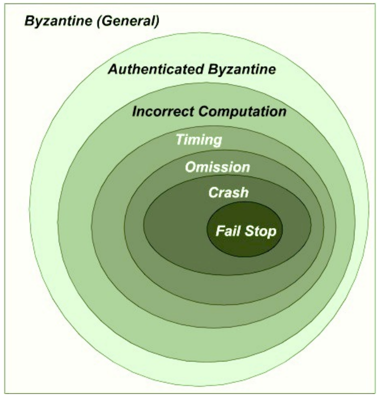
Fault Models at Switching Level

- Fault models at switching level are mainly used in two areas:
  - Design of circuits (for us not very interesting)
  - Communication
- Logical faults at wires are considered:
  - **Stuck-at-X faults**: Signal is always X ($X \in \{0, 1, valid, \ldots\}$)
  - **Bridging faults** (short circuits)
  - **Stuck-at-open faults** (undefined signal)

Fault Models for Distributed Systems

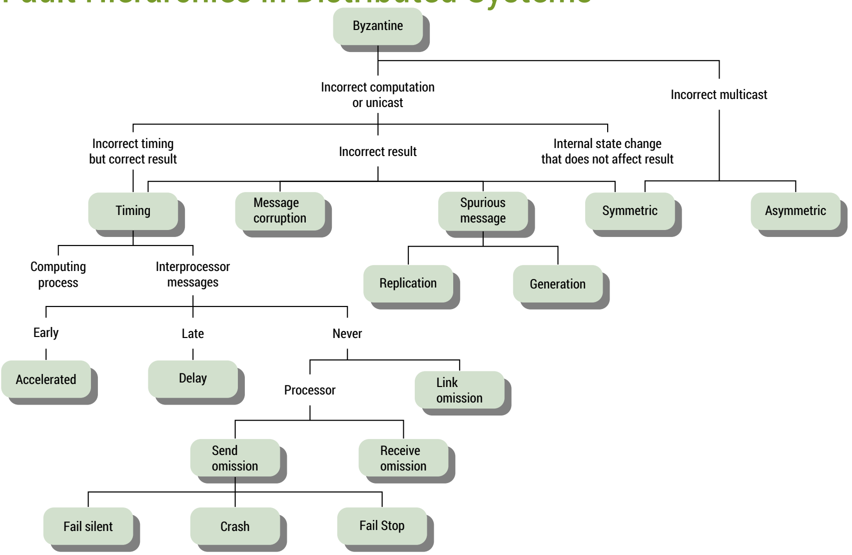
- Crash fault
- Omission fault
- Timing fault
- Incorrect computation/communication
- Arbitrary/ byzantine fault†

Link faults are mapped onto component faults of the sender or receiver



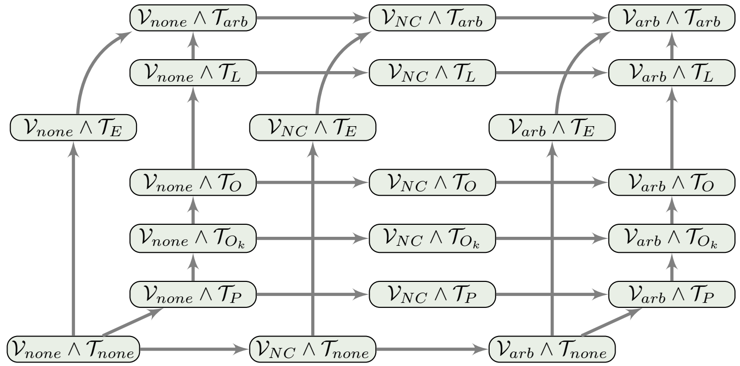
†In case of authenticated Byzantine faults, one assumes that no message can be inwardly altered.

Fault Hierarchies in Distributed Systems



Partial Order of System Faults

- Arrows represent decreasing strictness of assumptions
- $\mathcal{V}$: value domain, $\mathcal{T}$: time domain



P – permanent; O_k – sequential omission; O – omission; L – belate; E – premature; NC – coding ; arb – arbitrary

Statistical Assumptions

- Usually it is not sufficient to know the **impact** of faults
- Statements regarding **occurence** are necessary
  - How many faults at the same time (maximum)?
  - How often?
  - Independence of faults
- **Independence** of faults
  - Are faults occuring independent from each other?
  - Are faults correlated?
  - Are faults intended (attack)?

Statistical Assumptions (cont.)

- Statistical assumption usually consider mean values
  - Mean time to failure (MTTF, cf. Chapter 2):

Mean time to failure

$f(t)$ is the densitiy of the probability that **no failure** occurs until time $t_0 + t$.

$$\bar{t}_F = \int_0^\infty t \cdot f(t)dt$$

  - For failures with exponential distribution: $\bar{t} = MTTF = \frac{1}{\lambda}$

Faults as Stochastic Process

- Arrival of faults (or load/requests) is often described as **stochastic process**
- **Recap:**

Definition 4.1 (Stochastic process)

A **stochastic (or random) process** is a set $\{X(t)\}$ of random variables with a shared value domain (state space) and a shared index parameter t.

- Each instance of index $X(t)$ is a random variable for fixed t
- Usually, t is interpreted as time (especially for arrival processes)

Poisson Process

- Counting events of a discrete process until $t_0 + t$
- Number is $N(t)$
- It holds:
  - $N(t_0) = 0$
  - Independence in not overlapping intervals
  - Probability of an event only depends on t, not on t_0
  - $\lim_{t \to 0} \Pr\{\text{event in } t\} \sim t$

Then this is called a **Poisson-Prozess** and for $N(t)$ holds:

$$\Pr\{N(t) = m\} = \frac{(a \cdot t)^m}{m!} e^{-a \cdot t}$$

Poisson Process (cont.)

- $N(t)$ represents numbers of faults in interval
- Special case: Time to the **first** fault

$$\begin{aligned}\Pr\{N(t) = m = 0\} &= \frac{(a \cdot t)^m}{m!} e^{-a \cdot t} \\ &= \frac{(a \cdot t)^0}{0!} e^{-a \cdot t} \\ &= e^{-a \cdot t}\end{aligned}$$

➡ equals known probability of survival for constant fault rate (exponential)

➡ More in Chapter 7

Discussion: Fault and Load

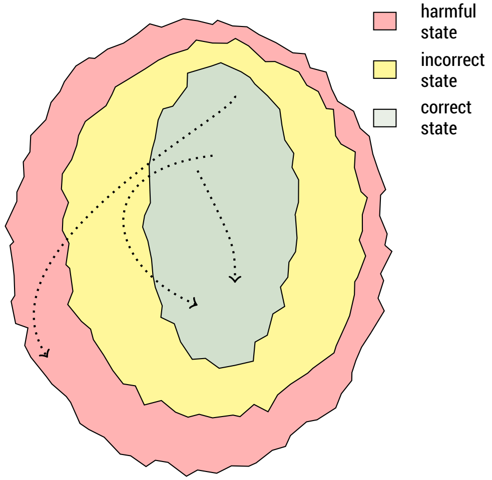
Interaction between load and classical failures

- Usually positive correlated:
  - Increasing load leads to **wearout**
    ➡ Failure rate increases
  - Higher load shows cause of failure faster
    ➡ $\Pr\{\text{error}|\text{fault}\}$ increases
  - (Recognized) faults lead to recovery measures
    ➡ Load increases
- Feedback possible
- Sometimes load **decreases** failure rate

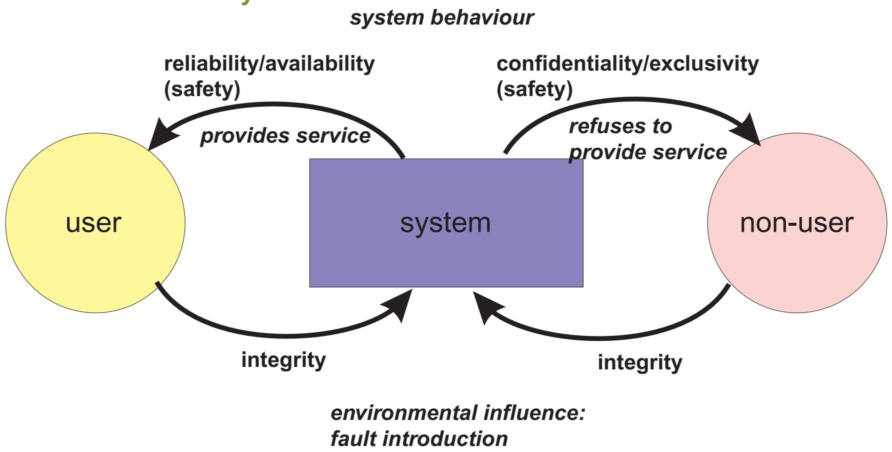
4.3 Alternative Descriptions

Description of Behavior

- Considers computer as automaton
- States may include time
- Sequence of states describes correct or incorrect behavior

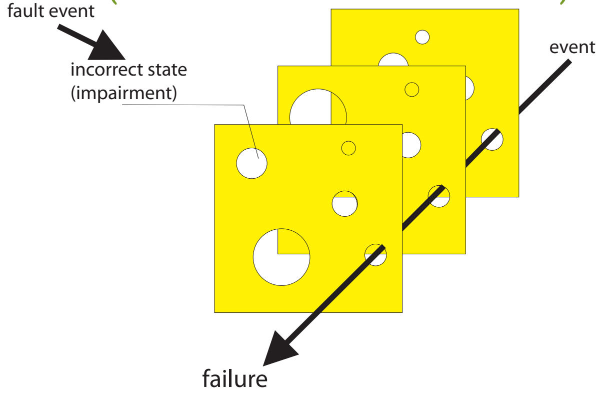


Holistic Model by Jonsson



- Attempt of standardization
- Includes *security*

Multiple Causes (D. Orlandella and J. Reason)



- Sometimes it is not possible to detect **the** root cause but the concurrence of several causes
- Description of potential impairments
- Swiss-Cheese-Model