# Fundamental Concepts of Dependability

Algirdas Avizienis, Jean-Claude Laprie, Brian Randell

Presented By

Neeraj Krishnan

CS 553, Spring 2003

# Authors

All have decades of research in fault tolerant computing behind them.

Avizienis, esp. on software faulttolerance

Laprie on control, □-electronics

Randell on HPCS, and dependability

# What we'll do

Dependability

Attributes (Where we want to go)

Threats (What we are up against)

Means (How we overcome)

Conclude

# Attributes of Dependability

Integrity

Availability, Reliability, Safety

Integrity + Availability + Confidentiality = Security

Maintainability

# Attributes of Dependability

More mathematical definitions:

Reliability: Probability that the system survives throughout [0,t] (MTTF)

Maintainability: Probability that a system will be repaired in time less than t. (MTTR)

Likewise for others.

Availability: MTTF/(MTTF + MTTR)

# Attributes of Dependability

Any design contradictions ?

What about availability and safety ?

Any examples ?

Maintainability as a measure of the tradeoff ?

Why a time measure for everything ?

# Threats

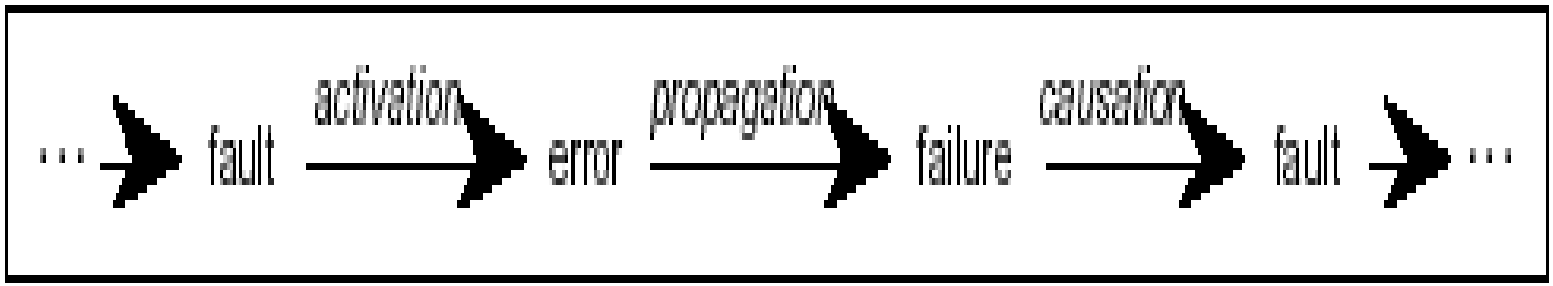
What are we up against ?

Failure: Deviation from specified service

Error: System state that can lead to a failure

Fault: Cause of an error

# Threats

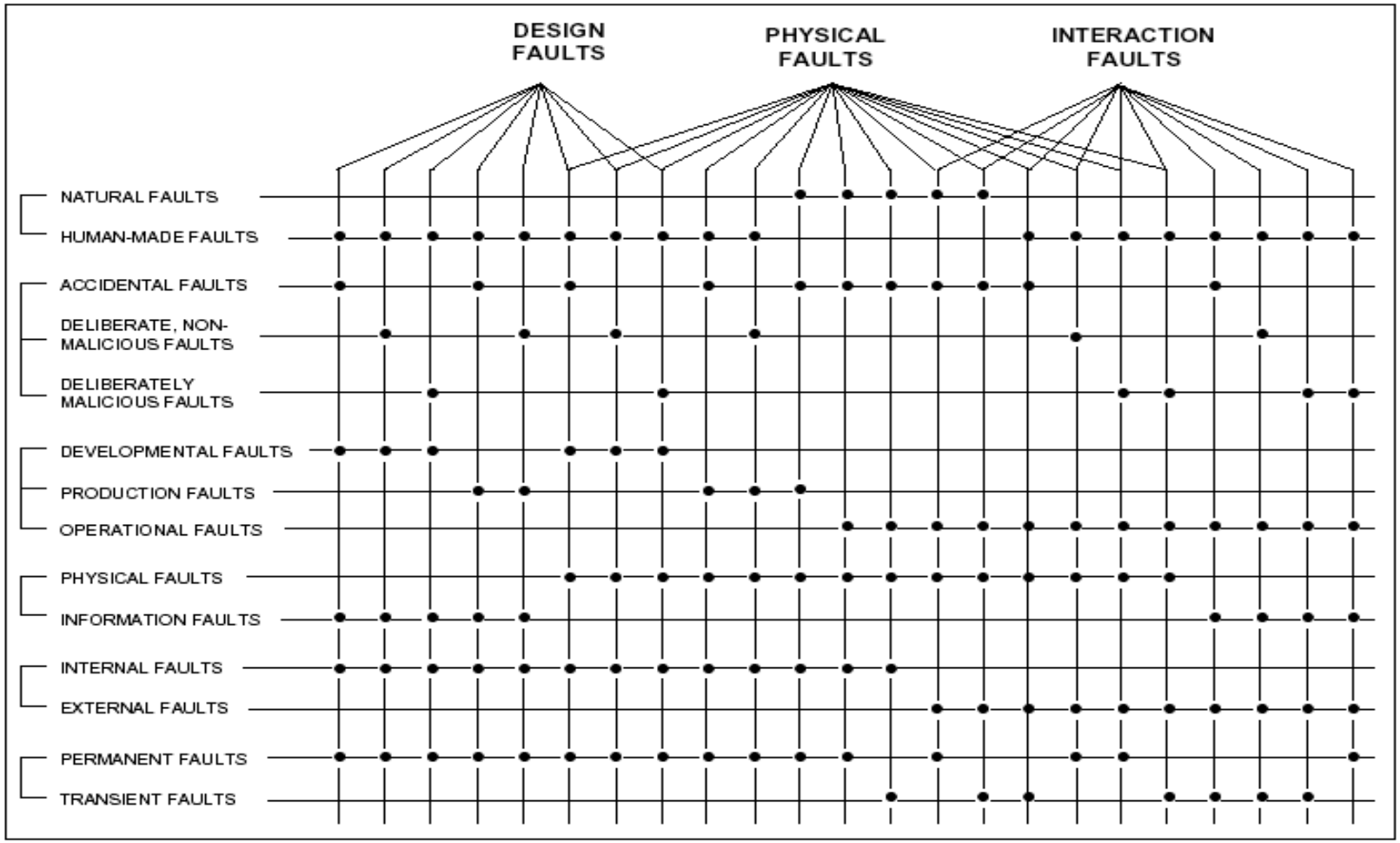

...
fault
activation
error
propagation
failure
causation
fault
...

# Threats

Examples:

1. Link failure(!) - no connection - no webpage (physical)
2. Dos - low b/w - no service (interaction)
3. Math library - incorrect float - divide by 0 (design)

# Fault Classes



# Means

How do we overcome ?

Fault prevention: QC, Classical s/w engineering principles. Modularity, information hiding, etc.

Fault tolerance: Deliver service in the presence of faults. Detect and recover, or maybe recover without detecting. Mask.

Fault removal: test by injecting faults, system must be verifiable.