II

DEPENDABILITY - CONCEPTS, STATE-OF-THE-ART, CHALLENGES

J-C Laprie

Rapporteur: C Sala-Oliveras

Dependability – Concepts*, State-of-the-Art, Challenges

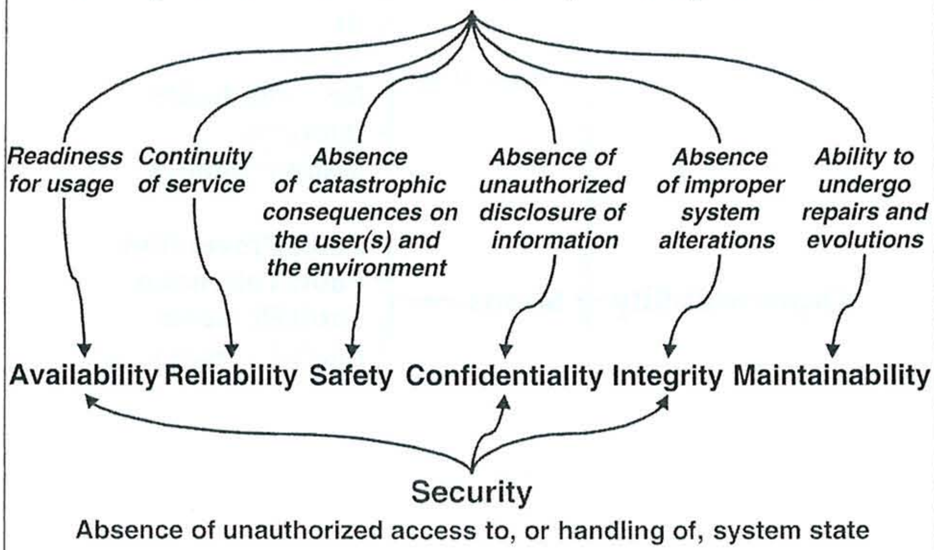
Jean-Claude Laprie

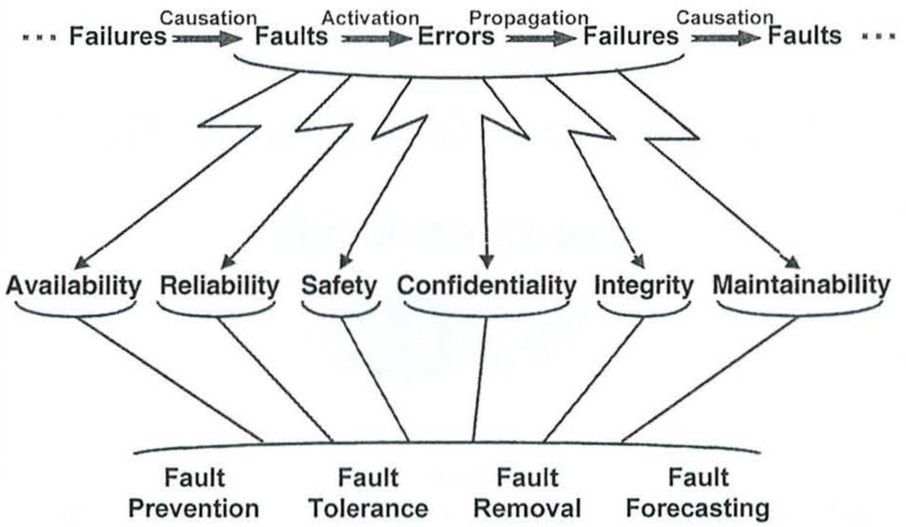
* Based on
A. Avizienis (UCLA), J.C. Laprie, B. Randell (Univ. Of Newcastle upon Tyne): *Fundamental Concepts* of *Dependability*

International Seminar — University of Newcastle upon Tyne
September 3-7, 2001

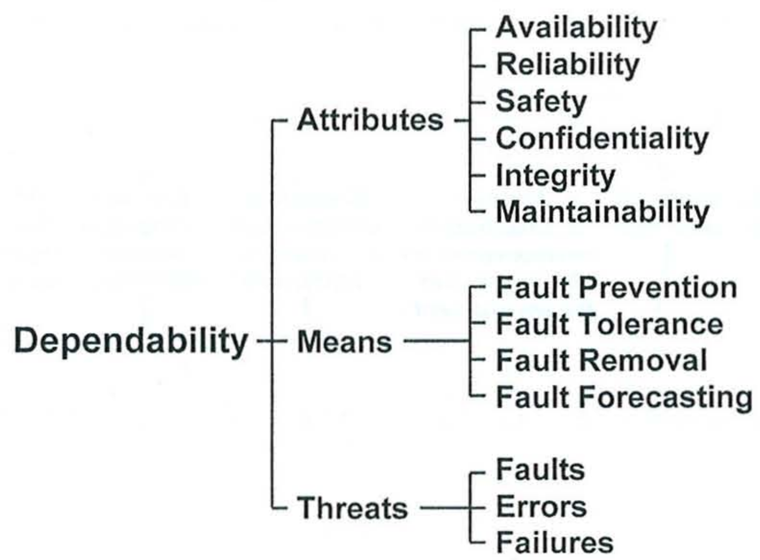
Dependability

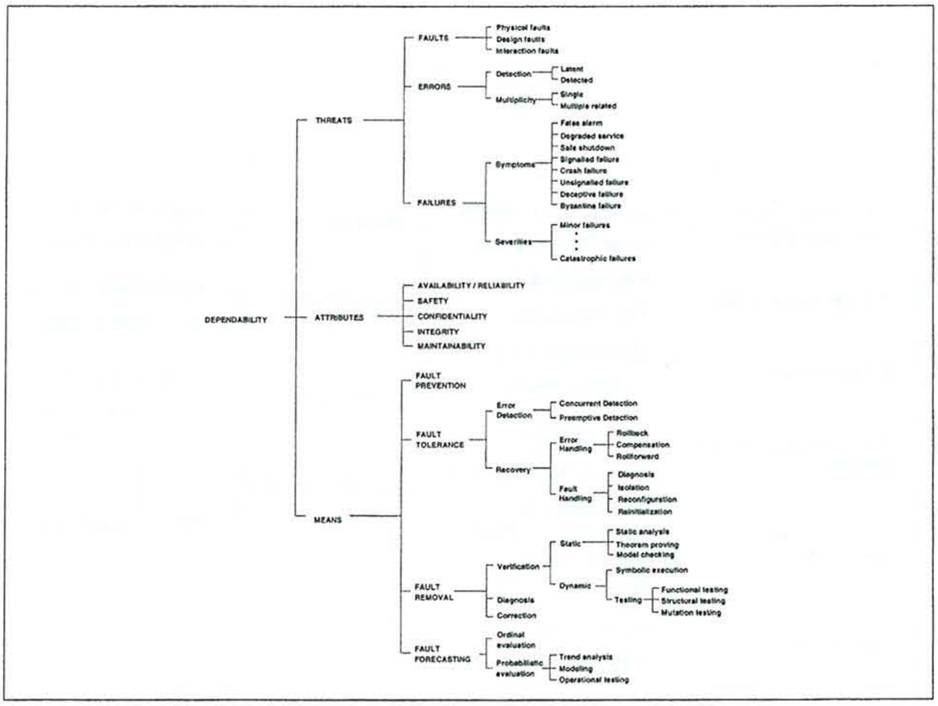
ability to deliver service that can justifiably be trusted

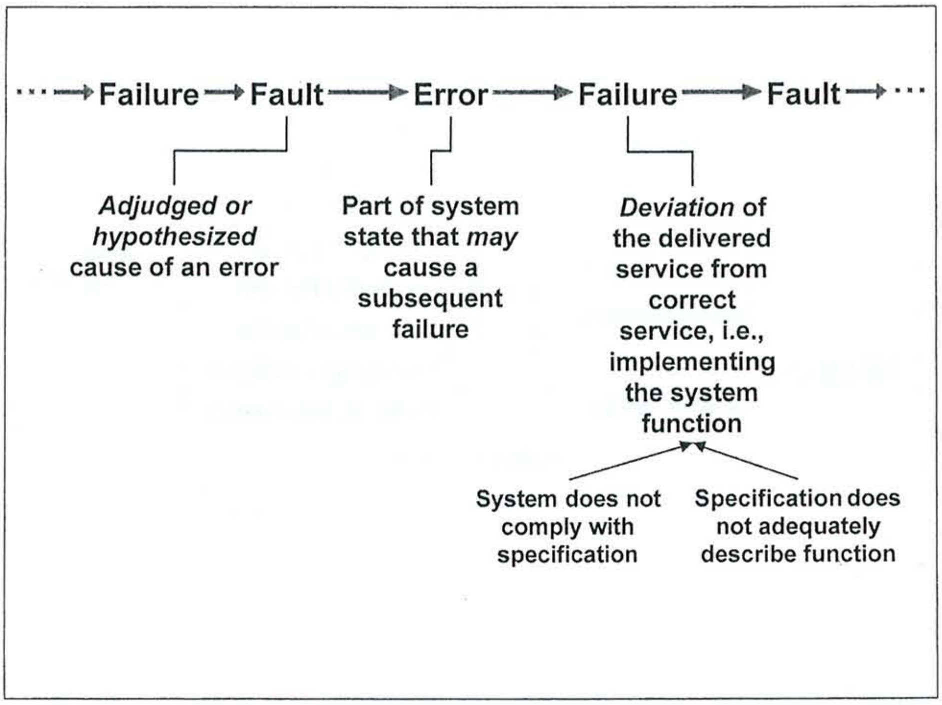


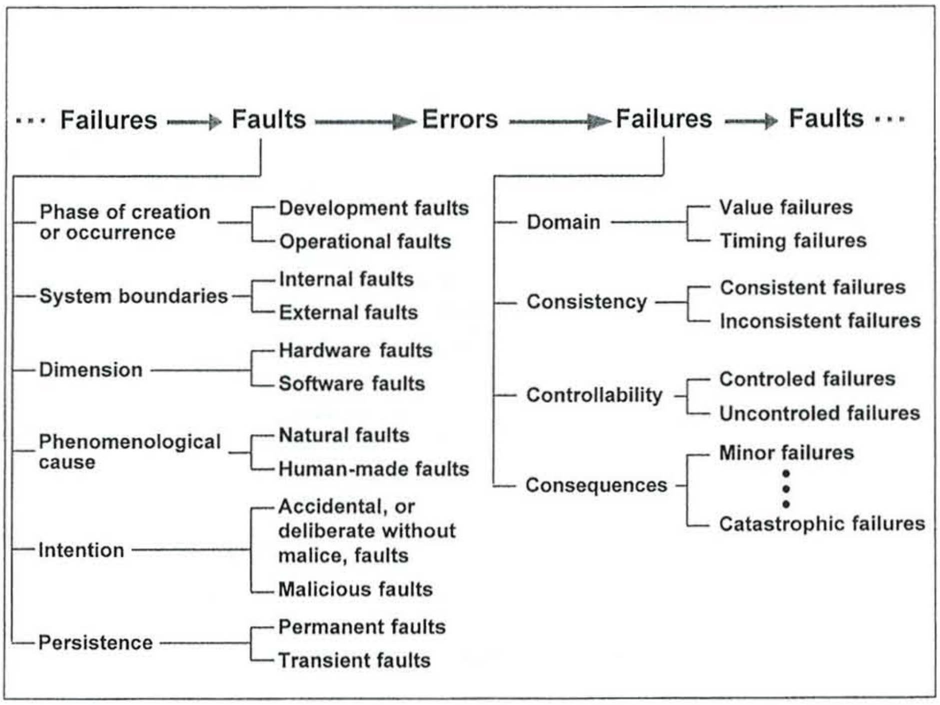


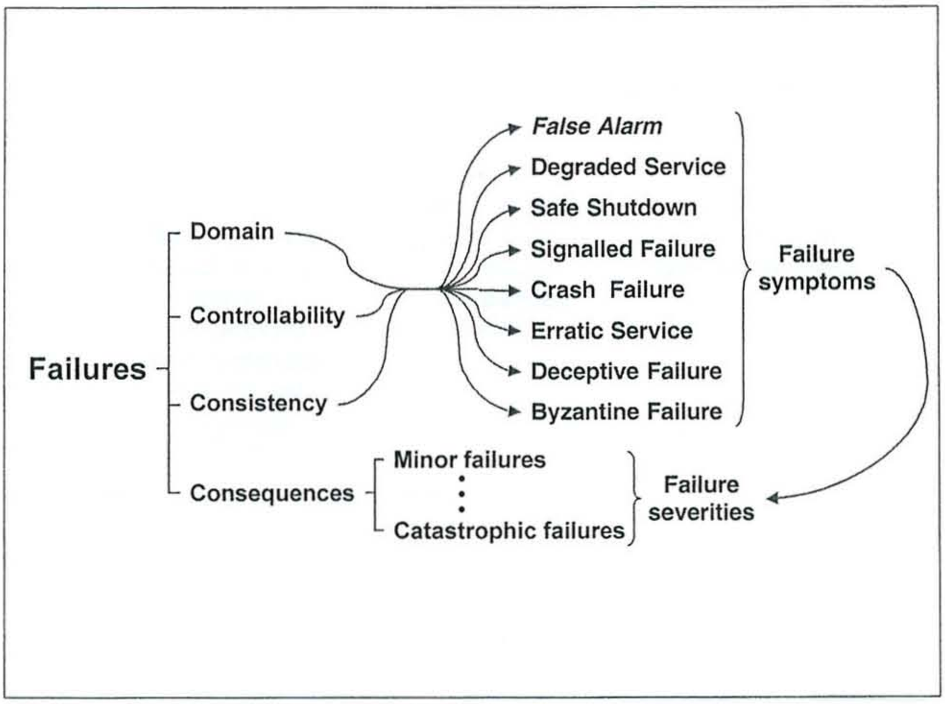
Dependability: ability to avoid failures that are more frequent or more severe, and outage durations that are longer, than is acceptable to the user(s)

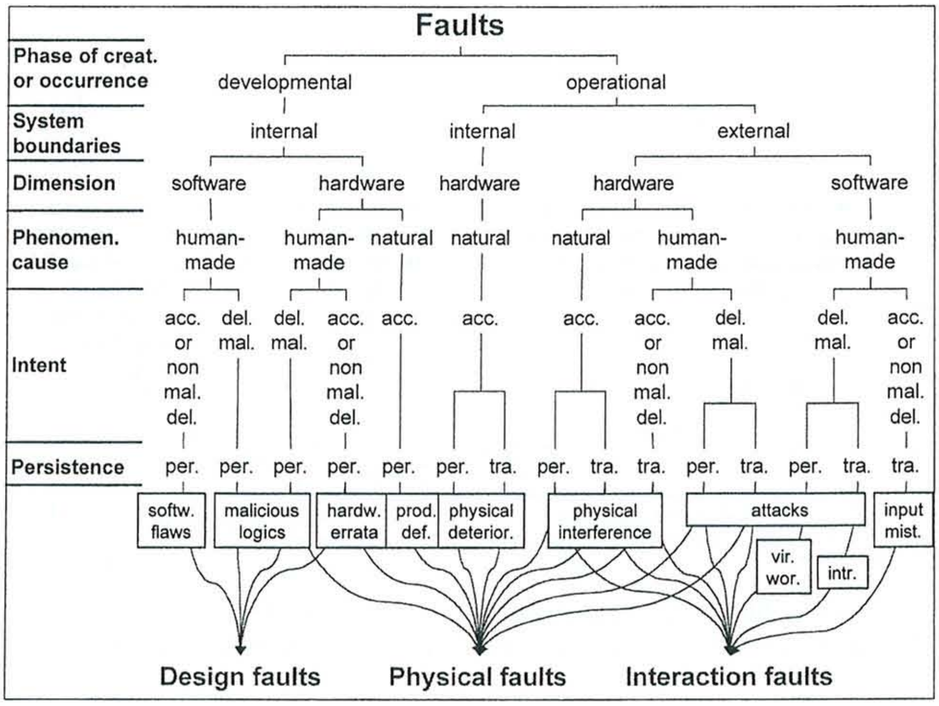


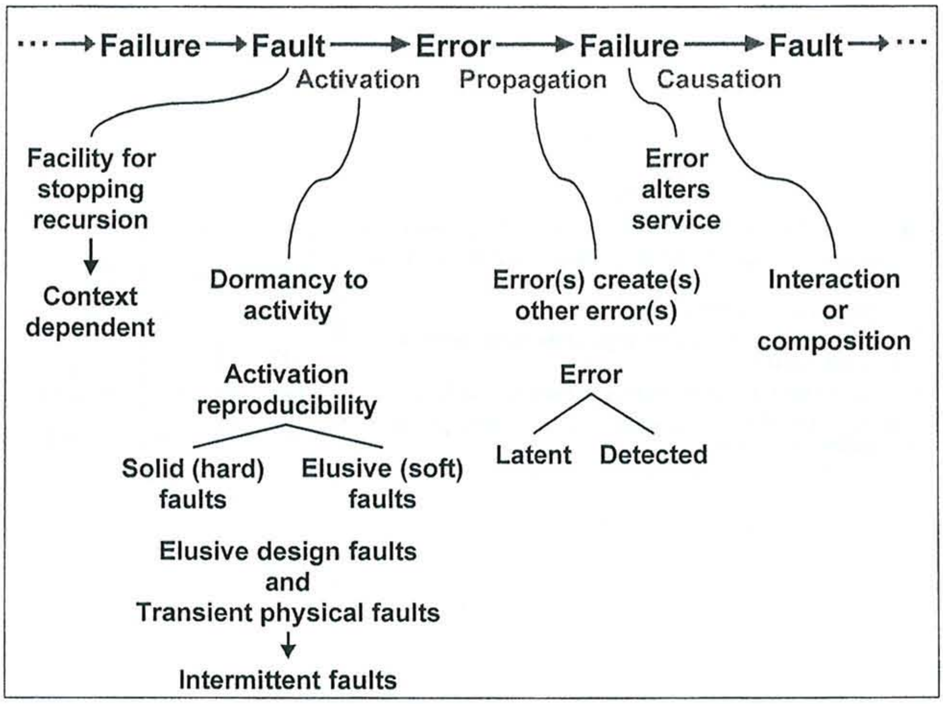

DEPENDABILITY
THREATS
FAULTS
Physical faults
Design faults
Interaction faults
ERRORS
Detection
Latent
Detected
Multiplicity
Single
Multiple related
FAILURES
Symptoms
False alarm
Degraded service
Safe shutdown
Signalled failure
Crash failure
Unsignalled failure
Deceptive failure
Byzantine failure
Severities
Minor failures
Catastrophic failures
ATTRIBUTES
AVAILABILITY / RELIABILITY
SAFETY
CONFIDENTIALITY
INTEGRITY
MAINTAINABILITY
MEANS
FAULT PREVENTION
FAULT TOLERANCE
Error Detection
Concurrent Detection
Preemptive Detection
Recovery
Error Handling
Rollback
Compensation
Rollforward
Fault Handling
Diagnosis
Isolation
Reconfiguration
Reinitialization
FAULT REMOVAL
Verification
Static
Static analysis
Theorem proving
Model checking
Dynamic
Symbolic execution
Testing
Functional testing
Structural testing
Mutation testing
Diagnosis
Correction
FAULT FORECASTING
Ordinal evaluation
Probabilistic evaluation
Trend analysis
Modeling
Operational testing

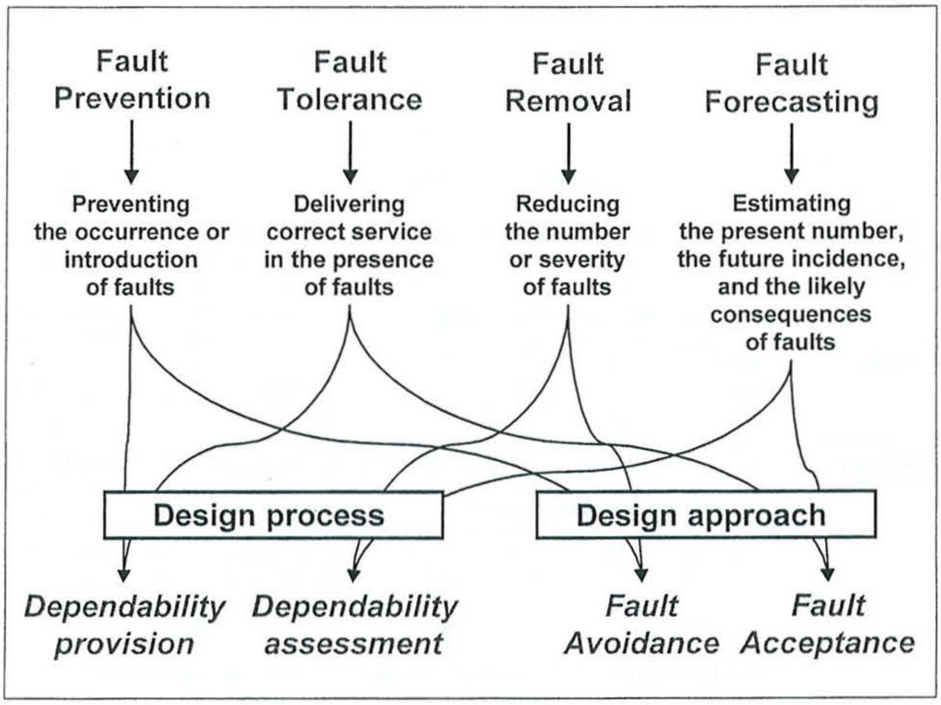

··· Failure → Fault → Error → Failure → Fault → ···
Adjudged or hypothesized cause of an error
Part of system state that *may* cause a subsequent failure
Deviation of the delivered service from correct service, i.e., implementing the system function
System does not comply with specification
Specification does not adequately describe function


··· Failures → Faults → Errors → Failures → Faults ···
Phase of creation or occurrence
Development faults
Operational faults
System boundaries
Internal faults
External faults
Dimension
Hardware faults
Software faults
Phenomenological cause
Natural faults
Human-made faults
Intention
Accidental, or deliberate without malice, faults
Malicious faults
Persistence
Permanent faults
Transient faults
Domain
Value failures
Timing failures
Consistency
Consistent failures
Inconsistent failures
Controllability
Controled failures
Uncontroled failures
Consequences
Minor failures
Catastrophic failures


Failures
Domain
Controllability
Consistency
Consequences
False Alarm
Degraded Service
Safe Shutdown
Signalled Failure
Crash Failure
Erratic Service
Deceptive Failure
Byzantine Failure
Failure symptoms
Minor failures
Catastrophic failures
Failure severities


Faults
Phase of creat. or occurrence
System boundaries
Dimension
Phenomen. cause
Intent
Persistence
developmental
operational
internal
internal
external
software
hardware
hardware
hardware
software
human-made
human-made
natural
natural
natural
human-made
human-made
acc. or non mal. del.
del. mal.
del. mal.
acc. or non mal. del.
acc.
acc.
acc.
acc. or non mal. del.
del. mal.
del. mal.
acc. or non mal. del.
per.
per.
per.
per.
per.
per.
tra.
per.
tra.
tra.
per.
tra.
per.
tra.
tra.
softw. flaws
malicious logics
hardw. errata
prod. def.
physical deterior.
physical interference
attacks
vir. wor.
intr.
input mist.
Design faults
Physical faults
Interaction faults


··· → Failure → Fault → Error → Failure → Fault → ···
Activation
Propagation
Causation
Facility for stopping recursion
Context dependent
Dormancy to activity
Activation reproducibility
Solid (hard) faults
Elusive (soft) faults
Elusive design faults and Transient physical faults
Intermittent faults
Error alters service
Error(s) create(s) other error(s)
Error
Latent
Detected
Interaction or composition



	Faults			Failures		Availability/ Reliability	Safety	Confidentiality
	Physical	Design	Interaction	Localized	Distributed			
June 1980: False alerts at NORAD	✓			✓		✓		
April 1981: First launch of the Space Shuttle postponed		✓		✓		✓		
June 1985 - January 1987: Excessive radiotherapy doses (Therac-25)		✓	✓	✓	✓	✓	✓	
August 1986 - 1987: the "wily hacker"		✓	✓	✓				✓
15 January 1990: 9 hours outage of the long-distance phone in the USA		✓			✓	✓		
February 1991: Scud missed by a Patriot (Gulf War)		✓	✓	✓		✓	✓	
November 1992: Communication crash of the London ambulance service		✓	✓		✓	✓	✓	
26 and 27 June 1993: Denial of credit card operations in France	✓	✓			✓	✓		
4 June 1996: Flight 501 failure of Ariane 5		✓		✓		✓		
17 July 1997: Internet .com domain mixed up			✓		✓	✓		
13 April 1998: Crash of AT&T data network		✓	✓		✓	✓		
February 2000: Distributed denials of service on large Web sites		✓	✓		✓	✓		
May 2000: virus "Iloveyou"		✓	✓		✓	✓		

Accidental (and non-malicious deliberate) faults

Number of failures [consequences and outage durations highly-application dependent]	Computer systems (e.g. Transactions, Electronic switching)		Larger, controlled, systems (e.g. Commercial airplanes; telephone network)	
	Rank	Proportion	Rank	Proportion
Physical internal	3	~ 10%	2	15-20%
Physical external	3	~ 10%	2	15-20%
Human-machine interaction *	2	~ 20%	1	40-50%
Design	1	~ 60%	2	15-20%

* Forensics evidence that interaction faults can often be traced back to design faults

Persistence	Solid	Intermittent
Physical and design	~ 10%	~ 90%

Deliberately malicious faults

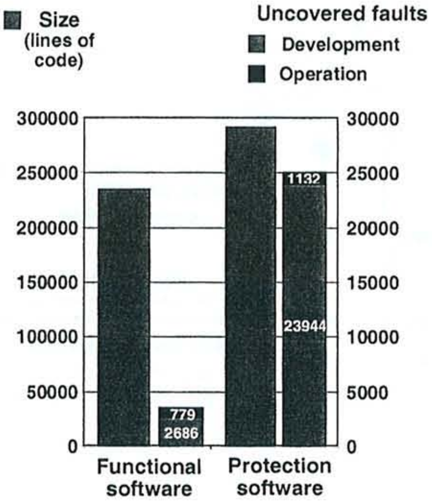
[Ernst & Young, 1998 ; 1200 companies in 32 countries]

Companies having experienced frauds during the last 12 months

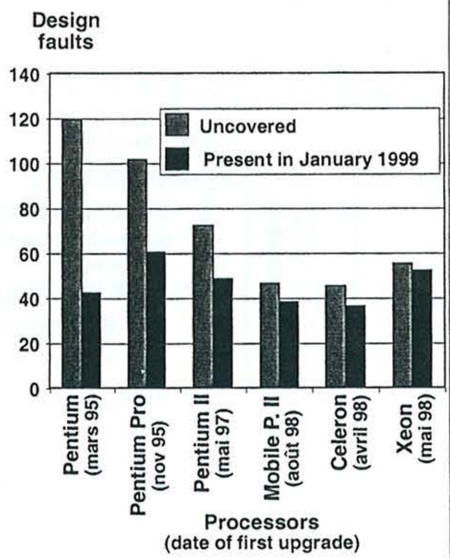
one at least: 66 % more than 5: 17 %

+ 85 % of frauds by employees

Upgrades of AT&T ESS-5

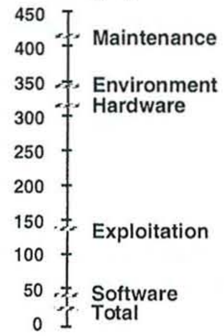


Intel processors (UCLA)



Tandem Fault Tolerant Systems

	Number	Duration (yrs)
Clients	2000	7000
Systems	9000	30000
Processors	25500	80000
Disks	74000	200000
Reprted outages		438
MTBF System		**21 yrs**



FAA Airworthiness Directives (AD)
(Brooklyn Polytechnic University)

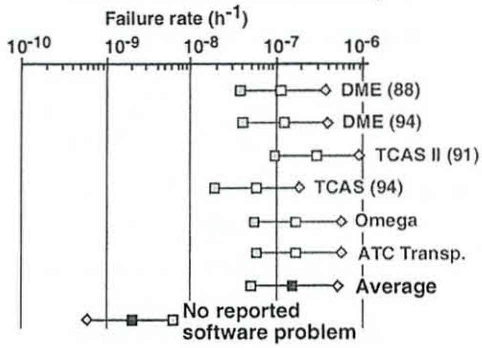
January 1st, 1980 - Sept. 21, 1994

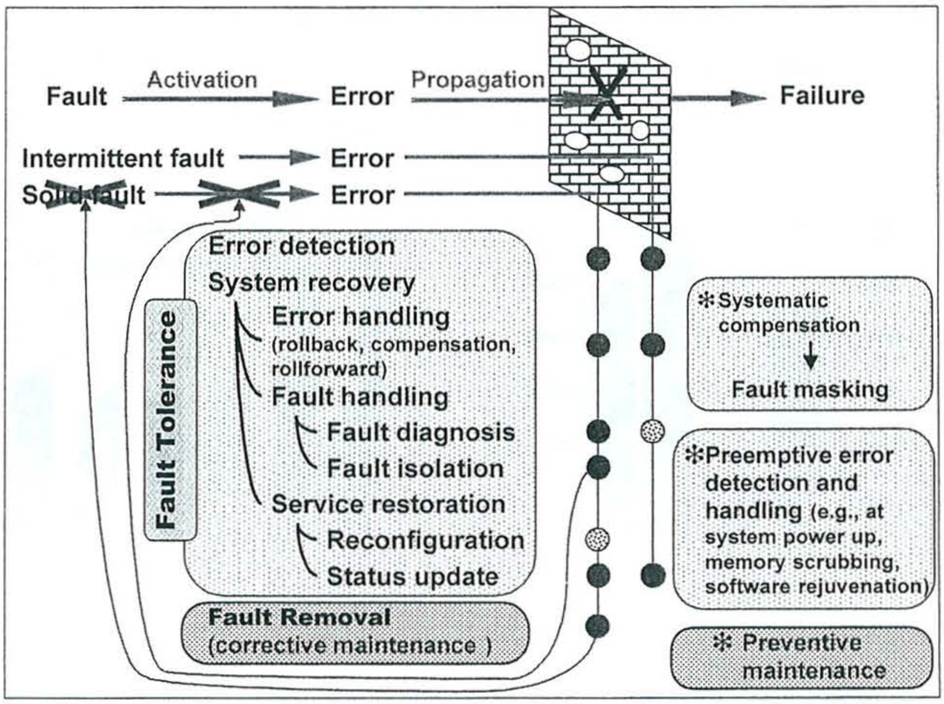
Confirmed avionics AD : 33

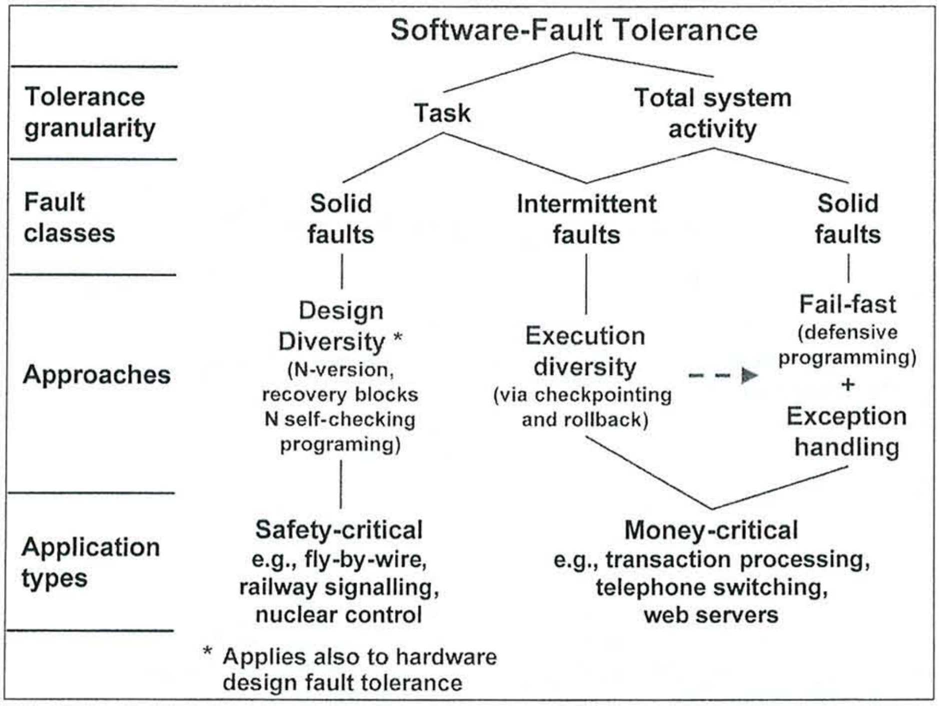
Hardware : 20 Software : 13

Equipments: Rockwell/Collins, Bendix/King, Honeywell/Sperry, Tracor Aerospace

Estimation of software reliability

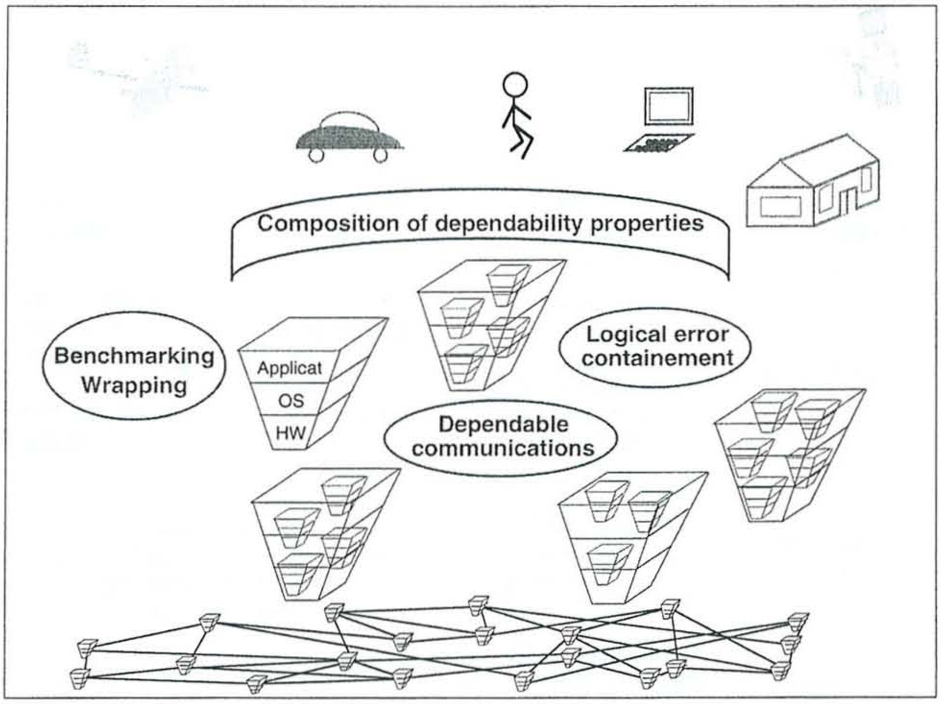


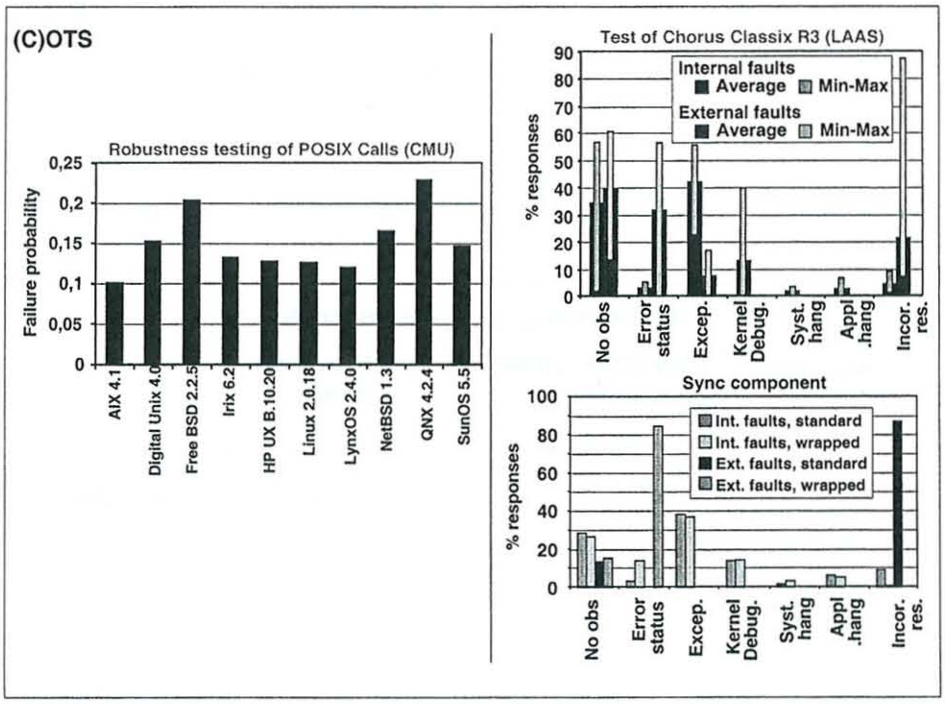




Malicious-Fault Tolerance

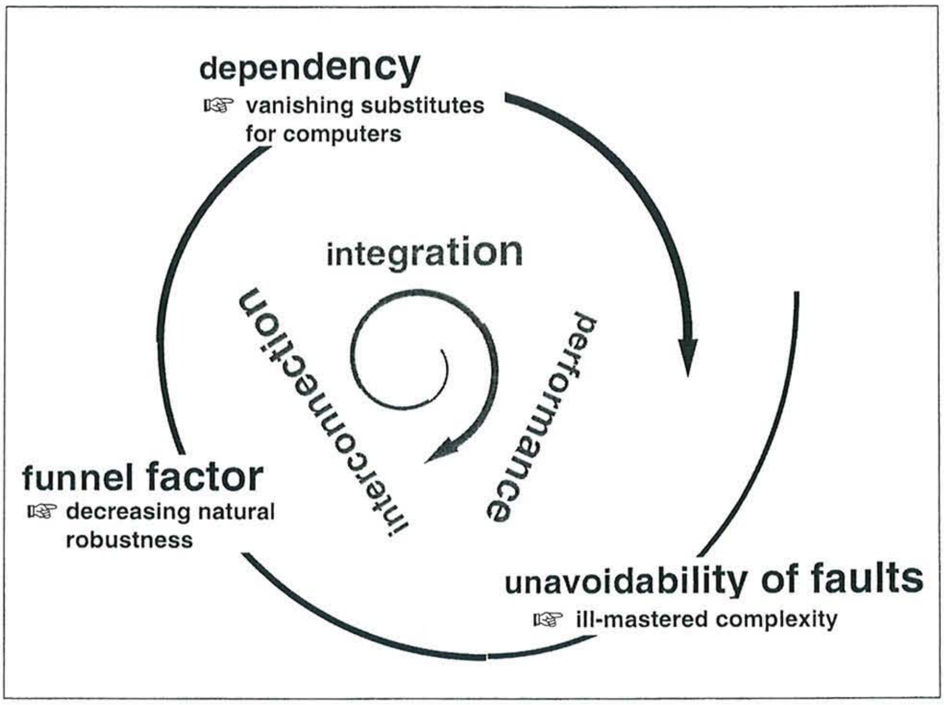
Fault classes	Malicious logics	Intrusions	Non-intrusive Attacks (wire-tapping, inference, covert channels, Tempest)
Detection	Access control Execution flow control	Access control User behavior analysis	
Detection-Recovery or Masking	Design Diversity	Encryption Fragmentation-scattering Deception	Encryption Fragmentation-scattering Jamming


Composition of dependability properties
Benchmarking
Wrapping
Applicat
OS
HW
Logical error
containement
Dependable
communications


(C)OTS
Robustness testing of POSIX Calls (CMU)
Failure probability
0,25
0,2
0,15
0,1
0,05
0
AIX 4.1
Digital Unix 4.0
Free BSD 2.2.5
Irix 6.2
HP UX B.10.20
Linux 2.0.18
LynxOS 2.4.0
NetBSD 1.3
QNX 4.2.4
SunOS 5.5
Test of Chorus Classix R3 (LAAS)
% responses
Internal faults
Average
Min-Max
External faults
Average
Min-Max
No obs
Error status
Excep.
Kernel Debug.
Syst. hang
Appl .hang
Incor. res.
Sync component
Int. faults, standard
Int. faults, wrapped
Ext. faults, standard
Ext. faults, wrapped

Dependability

- Subsumes concerns in reliability, availability, safety, confidentiality, integrity, maintenability — the *attributes of dependability* — within a unified conceptual framework; enables the appropriate balance between the attributes to be addressed
- *Means for dependability* — fault prevention, fault tolerance, fault removal, fault forecasting — provide an orthogonal classification of development activities; essential for abstract and discrete systems (nonexistent or vanishing safety factor)
- Causal chain of *threats to dependability* — fault - error - failure
  - Central to understanding and mastering various threats likely to affect a system
  - Provides for a unified presentation of those threats, though preserving their specificities via the various classes
- Rigorous terminology — not just definitions: a model
  - abstraction structuration recursion
  - Avoiding intellectual confusion(s)
  - Focusing on scientific problems and technical choices



❊ Cost of computer failures

	France [Insurers' association, private businesses]	USA [Find/SVP, large businesses]	UK [Insurers' association]
Accidental (and non-malicious intentional) faults	BFF 5 / Yr	B$ 4 / Yr	
Deliberately malicious faults	BFF 6 / Yr		B£ 1,25 / Yr

- ☞ Average cost per hour of downtime (lost revenue in banking, retail, manufacturing, health insurances, securities, reservations, etc.): $78,000
- ☞ Estimate of total yearly cost (USA): B$ 80

❊ Maintenance costs

- ☞ On-board Space Shuttle software: M$ 100 / year

❊ Undeployed software cost (development process failure)

☞ USA [Standish Group — 8380 projets]	Successful	Challenged	Cancelled
	1360 - 16%	4416 - 53%	2604 - 31%

~ B$ 81 lost yearly due to cancellations

	1983 estimate	1988 (contract awarded) estimate	1994 estimate	Schedule slippage (1994 estimate)
☞ FAA AAS	B$ 1	B$ 4	B$ 7	6 - 8 years

DISCUSSION

Rapporteur: C Sala Oliveras

Lecture One

Dr Laprie was talking about fault classification, especially those faults which are human made when Dr Ross pointed out that in the previous session (Dr Rushby's 2nd Lecture: Analyzing human factors with formal methods) they had a great discussion about the human mental image (mental model), and, sometimes, this human mental model is different from what is actually going on in the system. Dr Ross wondered if Dr Laprie's fault dimensions could be extended to include what is going on in the minds of people who are collaborating with the system, or if it would be another dimension in the sense of hardware and software versus training or cognition. Dr Laprie answered that he does not intend to look at what is happening in people's mind when they interact with systems, however one could always regard larger systems composed of computer systems plus operators. Professor Malek emphasized that one does have human-made faults. Dr Laprie agreed that indeed one has human-made faults, which are neither ergonomic nor interface related but more cognitive faults. So, regarding the classification of faults one clearly can have a classification of the operators' faults and indeed it is recognized that there is also this type of human fault dimension where the model that the operator has of the system does not match with the actual system behaviour. At this point, Dr Ross agreed that these faults are beyond the human interface. Dr Laprie went on to say that these faults are most difficult to detect and to correct (for the operator and for the system).

Dr Laprie was talking about wrapping and their benefits when Dr Lomet asked whether the wrappers tested the arguments of the calls. Dr Laprie responded that the wrappers test both the inputs and the outputs.

Mr Warne questioned if there was any good reason why Dr Laprie did not put timeliness as a fault in his taxonomy. Dr Laprie argued that timeliness, in the level of abstraction of his model, can be seen in the concept of continuity of service from a system.

Dr Laprie also pointed out that perhaps they were talking about different concepts of timeliness.

Dr Laprie was talking about losses in project cancellations due to software faults upon which Professor Malek questioned what the percentage of project cancellations were due to technological changes over time rather than to software faults. Professor Malek also commented that, for instance, a lot of military projects were simply cancelled because of technological changes and reasons other than software faults. Dr Laprie responded that he believed that most of the time, project cancellations are due to specification changes and/or bad service system performance.