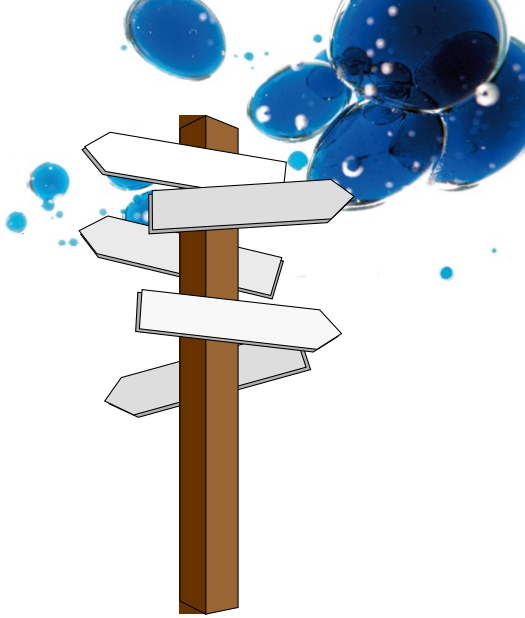
Robustness and Dependability

Seminar on Evolution, Complexity and Cognition (ECCO), October 1,2009

Wilfried Elmenreich
Senior Researcher

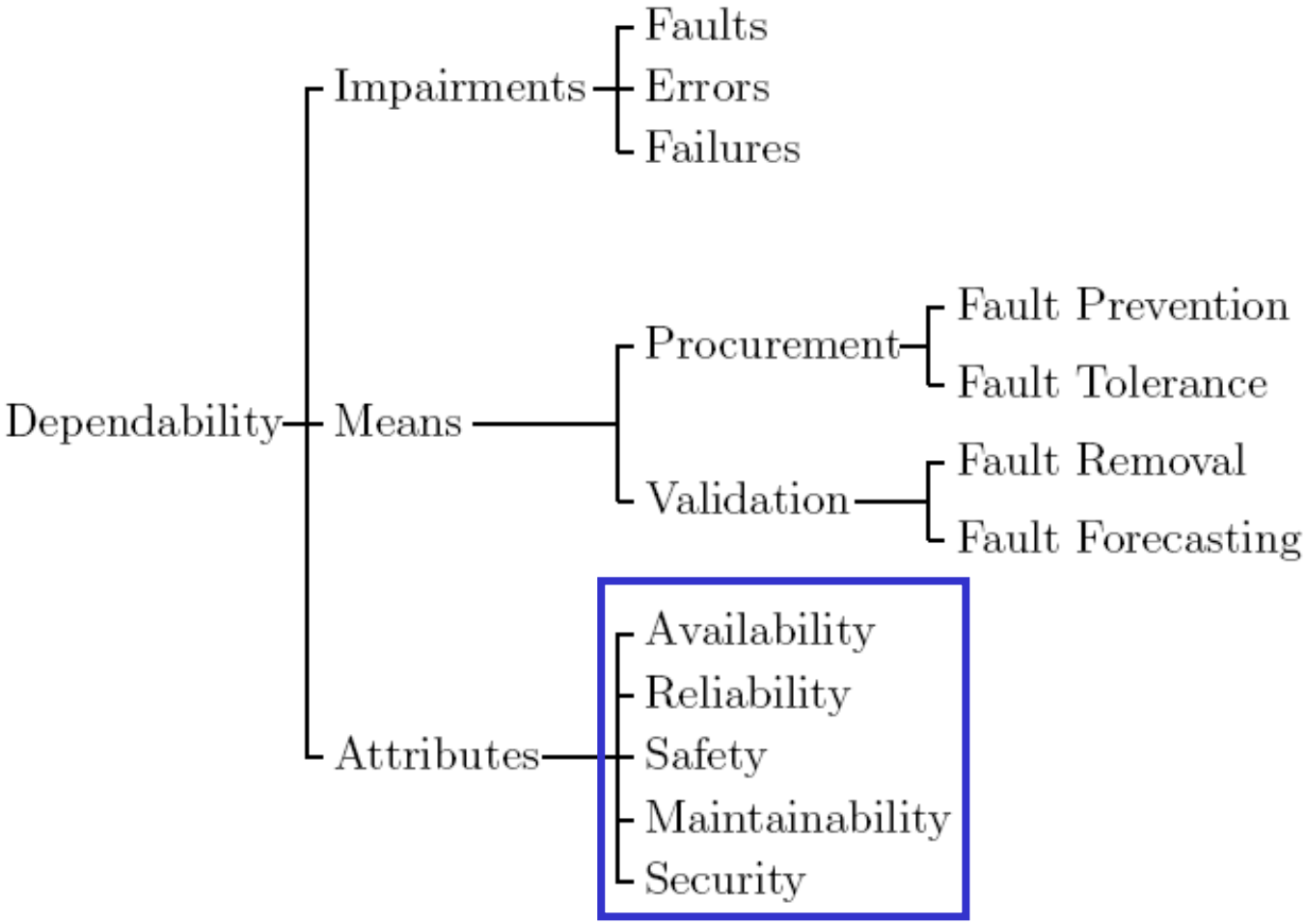
Mobile Systems Group/Lakeside Labs

Overview



- Dependability
  - Attributes
  - Means and impairments
  - Fault tolerance and fault hypothesis
- Robustness
  - Definition
  - Resilience, adaptability, recovery
- Summary

Means and Impairments of Dependability



Five Attributes of Dependability (1-3) [Lap92]

- Availability
  - Dependability with respect to the readiness for usage
  - The availability A(t) of a system is defined by the probability that the system is operational at a given point in time t.
- Reliability
  - Dependability with respect to the continuity of service
  - The reliability R(t) of a system is the probability that the system is operational during a given interval of time [0; t).
- Safety
  - Dependability with respect to the avoidance of catastrophic consequences
  - The safety S(t) of a system is the probability that no critical failure occurs in a given interval of time [0; t).

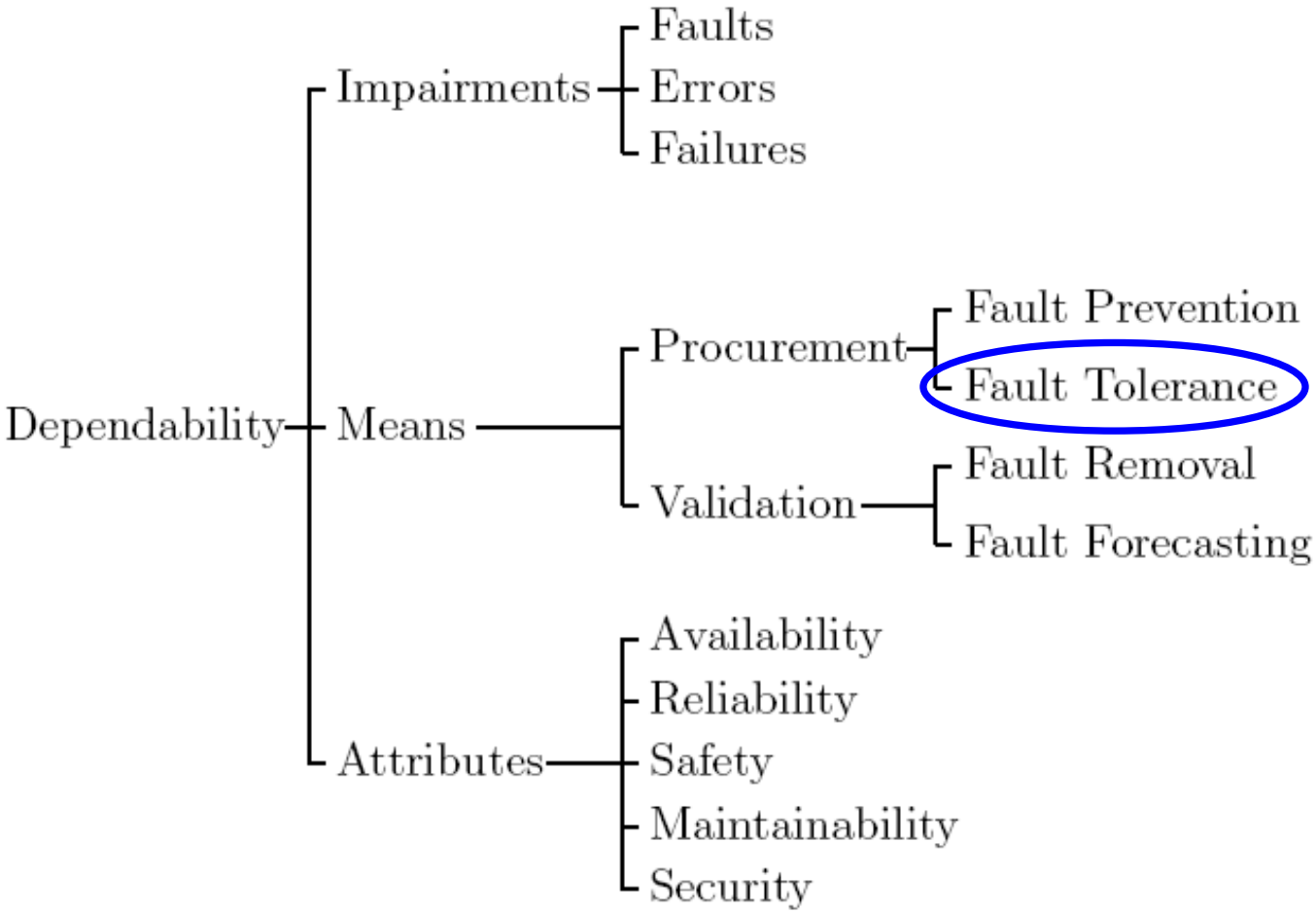
[Lap92] J. C. Laprie. Dependability: Basic Concepts and Terminology. In Dependable Computing and Fault Tolerant Systems, volume 5, pages 257{282. Springer Verlag, Vienna, 1992.

Five Attributes of Dependability (4-5)

- Maintainability
  - Measure of the time required to repair a system after the occurrence of a (benign) failure
  - Originally without quantification
  - Maintainability is defined as the probability M(d) that the system is restored within the duration d after failure.[Kop97]
- Security
  - Confidentiality
  - Integrity
  - No quantification

[Kop97] H. Kopetz. Real-Time Systems, Design Principles for Distributed Embedded Applications. Kluwer Academic Publishers, Boston, Dordrecht, London, 1997.

Means and Impairments of Dependability



Fault Tolerance

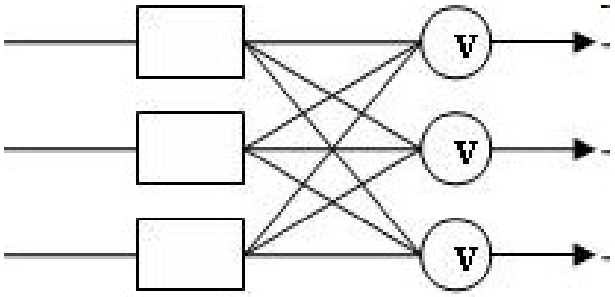
- Fault tolerance encompasses methods and techniques that enable the system to operate properly despite the presence of faults
- Real-Time Systems Community
  - Fault-tolerant systems mask failures (no system degradation)
- Wikipedia about fault tolerance
  - Possible reduced service after fault (graceful degradation)
- Main mechanism of fault tolerance is *redundancy*

Engineering of Fault-Tolerant Systems

- Specification of a *Fault Hypothesis* assuming type, frequency and number of fault scenarios
- All fault scenarios within the fault hypothesis are *covered faults*
- System is build to tolerate all faults within the fault hypothesis
- The assumption coverage gives the probability that faults occur within the fault hypothesis
- *No guaranties on the system behavior for uncovered faults*

Triple Modular Redundancy

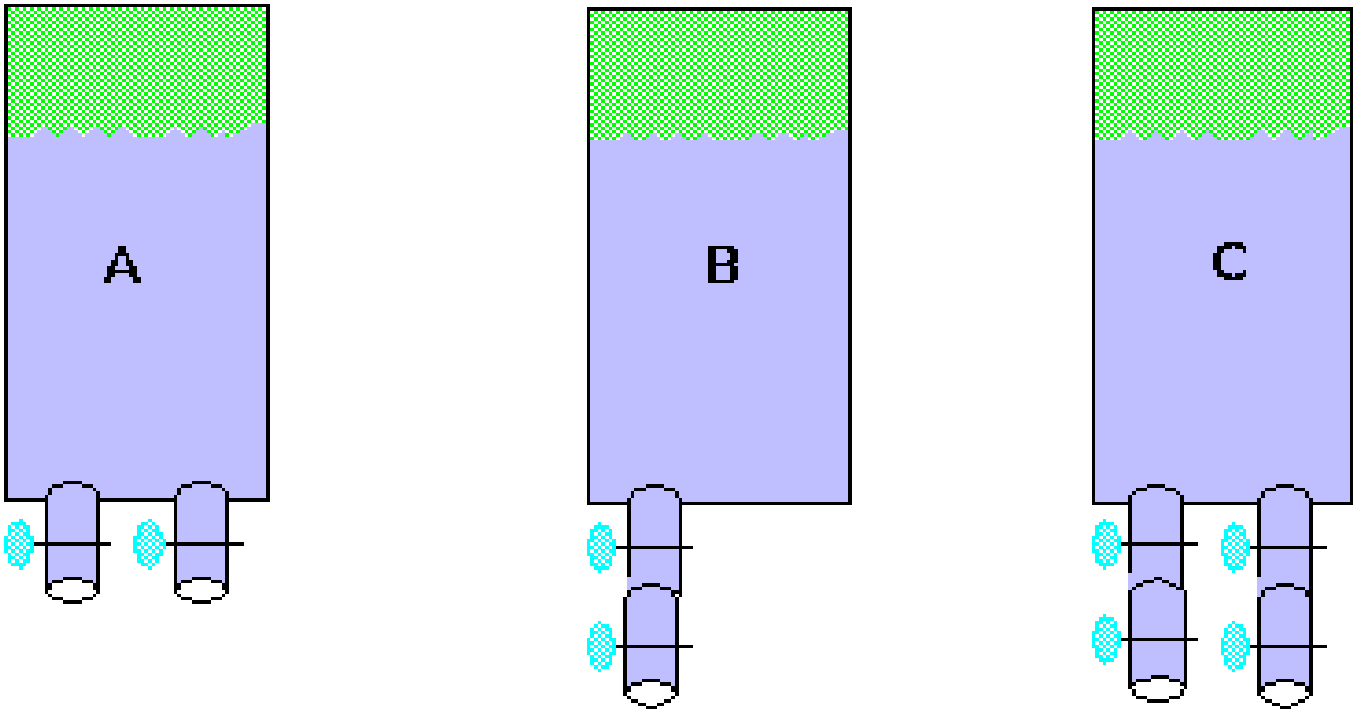
- Proposed by János Neumann (aka John von Neumann) in 1956
- Implements fault tolerance via triple redundancy and voters



- TMR unit forms a fault-containment region (TMR)
- Generalized as NMR (n-modular redundancy)

Fault-tolerant Example 1

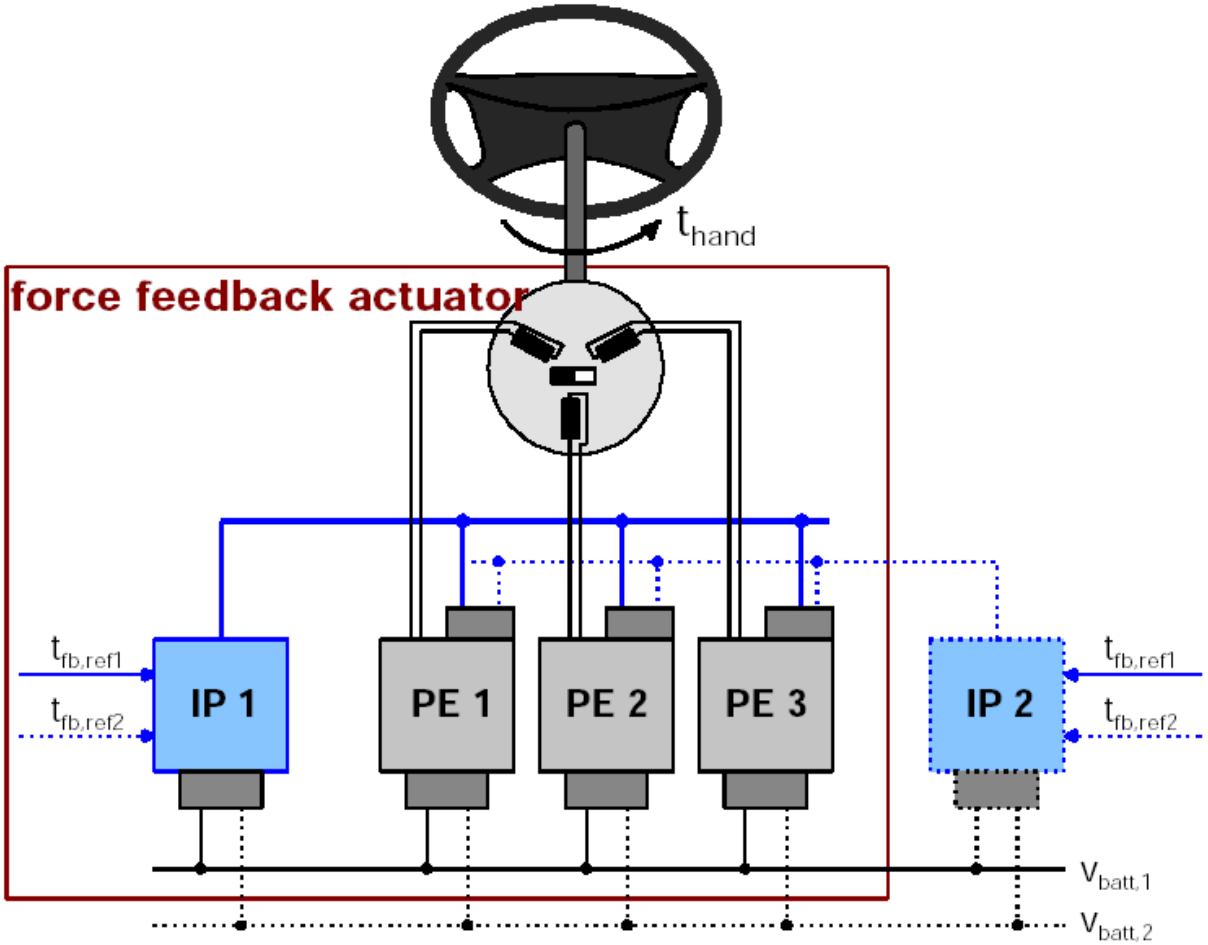
Simple watertank:



Fault-position: close open none

Fault-tolerant Example 2

Force feedback actuator



bustness


RIDICULOUSLY ROBUST
100% summer guarantee
SAFEWAY
Ingredients for life.


robust...

Robustness

- Robustness is the capability of a system to maintain an *acceptable* level of service despite various *unexpected* perturbations [Kit04,Ste04]
- Implemented via three main concepts [Mik09]
- Robustness in biology: „systems with a large associated neutral space of equivalent solutions to a given problem" [Wag07]

[Kit04] H. Kitano. Biological robustness. Nature, 5(11):826–837, Nov 2004.

[Ste04] J. Stelling, U. Sauer, Z. Szallasi, F. J. Doyle, and J. Doyle. Robustness of cellular functions. Cell, 118:675–685, Sep. 2004.

[Mik09] V. Mikolásek. Dependability and Robustness: State of the Art and Challenges. Workshop on Software Technologies for Future Dependable Distributed Systems, Tokyo, Japan, 2009.

[Wag07] A. Wagner. Robustness and Evolvability in Living Systems. Princeton Univ Press. 2007

Open questions

- When is a system robust?
  - Imagine a system running a very robust speech recognition software
  - What happens if you pull the plug?
- What is an acceptable level of service?
- (Do) we need a kind of fault hypothesis for designing robust systems(?)
  - We must at least name the class of expected faults
  - = „robust according to a particular property“

Why are we Interested in Robust Systems?

- Traditional dependability approaches require many resources (e.g. TMR)
- Complex systems become even more complex due to redundant computation and voting
- A robust system design could achieve the same with fewer resources and lower complexity

Engineering Robust Systems

- Design patterns:
  - Self-stabilization/self-organization
  - Reuse known concepts from biology or literature
  - Evolve system with robustness in the fitness function
- How to validate robustness?
  - Which test cases (how to expect „unexpected“ errors)
  - Difficult to specify a coverage

Resilience, Adaptability, Recovery [Mik09]

- Resilience
  - Capability of a component to compensate for a temporary degradation in a requested service
  - Example: humans understand a speech even in a noisy environment
- Adaptability
  - Capability to change system behavior or its logical/physical structure in order to compensate for internal or external perturbations and changing requirements
  - Example: Communication system with flexible routing
- Recovery
  - Avoid accumulation of errors
  - Example: Scheduled periodical reboot of internet servers

Interdependency of Resilience, Adaptability, Recovery

- Without recovery degradation is permanent
  → *recovery supports resilience*
- System must continue to work until repair
  → *resilience supports recovery*
- Adapting to a new configuration can cause a system degradation
  → *resilience supports adaptability*
- Integration of components
  → *common requirement for recovery and adaptability*

Robustness in Biology

- Robust systems are systems with a large associated neutral space of equivalent solutions to a given problem[Wag05]
  - Natural selection can further increase robustness by incremental evolution of a system within a neutral space
- Living cells are capable of maintaining their functionality under a variety of genetic changes and external perturbations[Dub08]
  - Intrinsic stability of attractors to achieve fault-tolerant computation

[Wag07] A. Wagner. Robustness and Evolvability in Living Systems. Princeton Univ Press. 2007
[Dub08] E. Dubrova. Self-Organization for Fault-Tolerance. IWSOS 2008.

Examples for Robust Systems

- The Internet
  - Very resilient against random node failures
  - Nobody made an exact fault hypothesis
- P2P systems
  - Breakdown of servers decrease performance, but service is still upright
- Self-Stabilizing Operating Systems
  - Proposes an automatic periodic or event-triggered re-install of the operating system components

[Dol04] S. Dolev, R. Yagel. Toward Self-Stabilizing Operating Systems. Proceedings of the Database and Expert Systems Applications. 2004

Conclusion

- Robustness and dependability are concepts for making a system less fragile
- Robustness is promising for more resource-efficient solutions
  - Especially for complex systems
- Engineering robust systems is difficult
  - No straightforward way to implement robustness
  - Open questions on testing and validation
- *A field for research!*