Design for Dependability 1

Takashi Nanya

- Building a Hierarchical Defense: An Immune System Paradigm for Fault-Tolerant System Design by Al Avizienis

- Dependable Computer Networks
  by Dimiter R Avresky

- Creating Robust Software Interfaces: Fast, Cheap, Good - Now you can get all three
  by Phil Koopman

Building a Hierarchical Defense: An Immune System Paradigm for Fault-Tolerant System Design by Al Avizienis

- Goal:

  Exploiting hierarchical fault-tolerant hardware infrastructure to assure system dependability or survivability.

- Arguments:

  Develop and employ the immune system analogy to explain the concept of the fault tolerance infrastructure.

  Difficulty in distinguishing between hardware and software.

- Main results expected:

  Convenient analogy to explain a set of design principle for fault-tolerant systems that are based on the separation of hardware and software.

Dependable Computer Networks
by Dimiter R Avresky

- Goal:

to build computer networks, based on COTS , that are inexpensive, scalable and dependable

Arguments:

- Computer networks are created by composing a complex set of hardware and software components that are heterogeneous and subject to continuous upgrade, replacement and scaling their numbers.

- Main results expected:

Transparent fault-tolerance.

Validation with a real-life application.

Introducing and developing detection and recovery-oriented technique for achieving high availability of computer networks.

Creating Robust Software Interfaces: Fast, Cheap, Good - Now you can get all three by Phil Koopman

- Goal:

to make sure that software is indeed written to be robust.

- Arguments:

People ignore software robustness improvement.

The big future problem for “near-stationary” robustness isn’t technology, It’s awareness and training

- Main results expected:

making software interface robust with little performance overhead.