Murphy Was an Optimist

Kevin R. Driscoll

Kevin.Driscoll@Honeywell.com

Visegrád

Murphy's Law

- Murphy's Law says:
  "If anything can go wrong, it will go wrong."

- Revised for Critical Embedded Systems:
  "… and, if anything *can't* go wrong, it will go wrong anyway."

Requirements Are Beyond Our Experience

- A typical designer has much less than 5,000 hours of real hands-on experience
- So, when a designer says that the failure can't happen, this means that it hasn't been seen in the designer's lifetime of observation (i.e., under 5,000 hours)
- But 5,000 is an insignificant fraction of the 10,000,000 or 1,000,000,000 hours corresponding to typical high dependability requirements

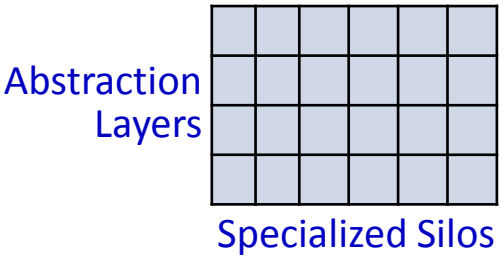
➔ ***We cannot rely on our experience-based intuition to determine whether a failure can happen within required probability limits***

- Why not use the literature to gain virtual experience?

  ... Because, papers about real occurrences of rare faults are difficult to find
  - Designers, maintainers, and users don't have "write papers" in their job descriptions
  - Organizations and people don't want to admit failures
  - The feeling that "once-in-a-lifetime" faults are not worth reporting
    - Most potential authors and some referees/reviewers have this feeling
  - Some referees/reviewers have "If I don't know about it, it doesn't exist" arrogance
  - Hard to find a popular venue and category for papers about single failures
    - Not enough material for a paper (1 to 2 pages), even for an "Experience" paper
    - Not really a "Quick Abstract"
    - Best fit maybe is a "Note" (but, where?)
    - Risks Forum (Digest): web page, email list, comp.risks newsgroup
      - Not a peer-reviewed publication (some posts are published in journal columns)
      - Most contributions are hearsay and don't have enough detail to be useful to desigmers

  ➔ ***Where should such "papers" be published and made known to designers?***

Complexity Division Boundaries



- To manage increasing complexity, we:
  - Abstract
    - Divide the problem into layers of less/more detail
    - Failures can happen in layers that designers don't see
  - Specialize
    - Divide the problem into slices (silos) of different technology disciplines
    - Failures can hide in the "cracks" between disciplines
- Phenomena crossing a sufficient number of boundaries is indistinguishable from magic*

* Apologies to Arthur C. Clark and his 3rd law of prediction

Transmogrification* Examples

- A diode became a capacitor, causing all four NASA space shuttle processors to disagree amongst themselves (Byzantine problem) during the countdown for mission STS-124
- An integrated circuit input become an output, causing panic at the most important secure switchboard in country
- Conversion of "stuck at" failures to oscillatory failures
  - RS-485 (driver → oscillator), escaped intended fault containment zone
  - MIL STD 1553 (amplifier → oscillator → phase-lock-loop)
- Example of "partogenesis"
  - Partogenesis = creation of a component that didn't exist before
  - Capacitor added to an integrated circuit

* Transmogrification definition: the act of changing into a different form or appearance (especially a fantastic or grotesque one), often as if by magic

Some Other Byzantine Failures

- Previously reported in 2003 Safecomp and 2004 DASC
  - Another space shuttle example
    - Different triggering cause (wrong bus termination resistor)
    - Slightly different symptom (2:2 disagreement)
  - Multi-Microprocessor Flight Control System (M^2FCS)
  - Potential grounding of an entire aircraft fleet of 150+ airplanes
  - Heavy ion fault injection in an early version of TTP/C silicon
- Two other Byzantine failures
  - Mid-value select (MVS)
    - Shows asynchronous, inexact voting is not immune
  - Command/Monitor (COM/MON)
    - Lesson learned:
      It may be impossible to create a COM / MON (or any wrap-back fault detection mechanism) which can observe all failures that might escape!

Some “Out of Band” Fault Propagations

- Reset propagated through ground
- Self-inflicted shrapnel
  - Jet engines (A-380, Sioux City DC-10, ...)
  - Exploding capacitor
- All ICs in an avionics box died, iff the plane flew a greater than 3-G climbing right-turn with side slip and was above 10,000 feet

Some Software Examples

- Software “evaporated” from memory only at a certain temperature range
- Bad software caused CPU to lock up (even reset wouldn’t work)
- Exhaustively tested software started causing a problem when some other software was changed
- Miscompare between identical copies of software running on identical hardware
- Software didn’t operate correctly even though the source code and compiler were bug free.

Arthur C. Clark's 1st law of prediction:
"When a distinguished but elderly scientist states that something is possible, he is almost certainly right. When he states that something is impossible, he is very probably wrong."

→ Any statement of "that can't happen" should be given a great deal of skepticism.

If you have any stories about rare failure modes, please e-mail them to me: Kevin.Driscoll@Honeywell.com
I will be posting some stories on a web site for NASA.
Similarly, we are collecting questions that should be asked when reviewing a system designed high dependability.