Formal Methods at Scale
Formal + Informal

Paul Miner
NASA Langley Research Center
p.s.miner@nasa.gov

25 September 2019

Issues of scale

- One niche in aerospace is formal analysis of specific models and requirements codified into industry-wide standards
  - When possible, design models to be reusable for similar applications (open source release)
- Three recent examples:
  - SAE AS 6802: Time-Triggered Ethernet
    - Formalization of fault-tolerance properties of a safety-critical data network
  - (SC 228) RTCA DO-365: Minimum Operational Performance Standards [MOPS] for Detect and Avoid (DAA) Systems
    - Formalization of well-clear requirements and DAA alerting logic
  - (SC 186) RTCA DO-260C (draft): MOPS for ADS-B
    - Formalization of requirements and reference implementation of Compact Position Reporting algorithm

Time-Triggered Ethernet

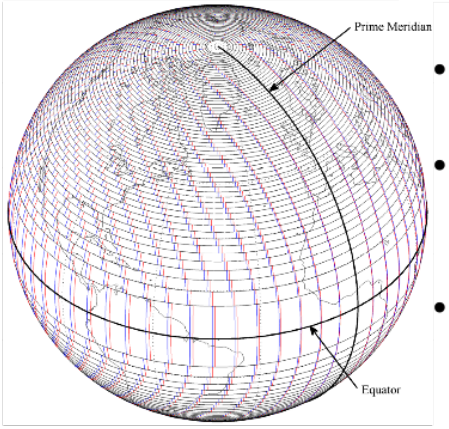
- Fault-tolerant data network for safety-critical applications
- Formal modeling and analysis using SAL and PVS during development of AS 6802 standard (by SRI, TTTech, & Honeywell)
- Analysis using PVS helped identify and fix a design defect. The fix was incorporated into the published standard:
  - B. Dutertre, A. Easwaran, B. Hall, and W. Steiner, Model-Based Analysis of Timed-Triggered Ethernet, presented at the 31st Digital Avionics Systems Conference (DASC), October 2012.

Detect and Avoid Alerting Logic For Unmanned Systems (DAIDALUS)

- Reference implementation of NASA's **detect and avoid** concept for the integration of **Unmanned Aircraft Systems** into civil airspace (RTCA DO 365).
- **Formally verified** core algorithms that:
  - Determine the current pairwise well-clear status (Detection Logic).
  - Compute maneuver guidance to maintain or regain well-clear status (Maneuver Guidance Logic).
  - Determine alert type (Alerting Logic).
- DAIDALUS core algorithms have been implemented as an Application Programming Interface (API) library in Java and C++ (≈ 44k lines of code).
- DAIDALUS API provides a highly configurable interface:
  - Aircraft performance limits (acceleration, turn rate, etc.)
  - Wind information (simple wind-field model)
  - Alerting and guidance thresholds
- Code is released under NASA Open Source Agreement:
  http://github.com/nasa/wellclear

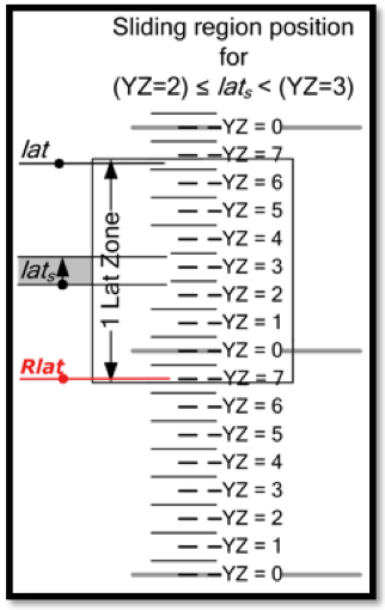
DAIDALUS

Compact Position Reporting – ADS-B Positioning



CPR divides the globe into "zones," and transmits only the target's position within the zone. The receiver has to determine the correct zone for proper decoding.

- CPR is used to save message space in ADS-B position messages
- Formal analysis led to tightening of decoding requirements, and simplified calculations.
- Spurred development of a PRECiSA, a tool for formal analysis of floating point (IEEE-754 spec) programs
- Formally verified implementations in floating point (double) and fixed point (single).
- Changes from formal analysis and verified implementation to be in revision C of DO-260 (ABS-B MOPS)



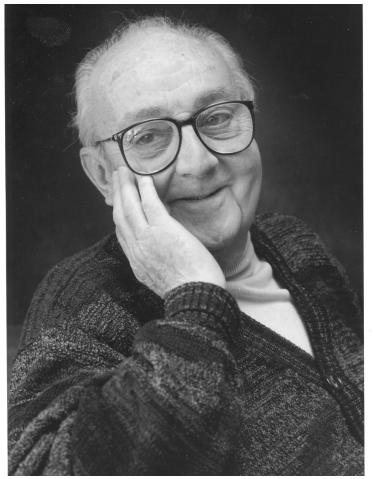
Visualization of loose requirement for decoding. Target is within stated distance threshold, but decodes incorrectly

https://shemesh.larc.nasa.gov/fm/CPR/
https://shemesh.larc.nasa.gov/fm/PRECiSA/

3

On (formal) models …

- *"Essentially, all models are wrong, but some are useful"*
  - George Box



Formal or Informal models?

Benefits

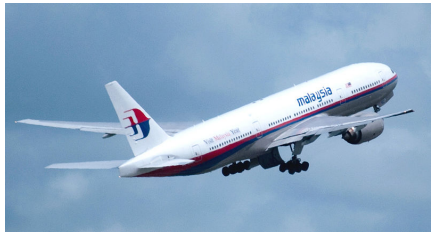
- Explore system behavior earlier in lifecycle
- Ability to verify properties that cannot be effectively demonstrated by test
  - Airborne separation
  - Robust partitioning for Integrated Modular Avionics
  - No memory leaks, buffer overflows, etc.
- …

Risks

- Invalid assumptions
- Unstated assumptions
- Tendency to conflate model with reality
- Maintaining consistency between multiple models (with different underlying abstractions)
- Incompatibility between models
  - Especially design models vs. failure models
- …

Representative Avionics Incidents

- Ariane 501 – 4 June 1996; Software defect in initialization routine for the inertial reference unit resulted in shutdown of both IRU and subsequent loss of rocket 37 seconds into launch
- B777 Malaysia Airlines Flight 124 – 1 August 2005; Latent software defect in ADIRU startup routine forgot prior failure of an accelerometer; second failed accelerometer resulted in incorrect data output from ADIRU to other critical systems (ADIRU was supposed to fail silent in this case).
- F-22 International Date Line – February 2007; multiple software-related systems failures when crossing the 180th meridian; failures resulted in simultaneous loss of navigation and communication; clear weather allowed squadron to follow tankers back to Hawaii.



Common to these incidents is presence of a software defect coupled with error propagation affecting critical functions unrelated to the software failure..

First Picture of a Byzantine Fault?

Honeywell

At 12:12 GMT 13 May 2008, a NASA Space Shuttle was loading hypergolic fuel for mission STS-124 when a 3-1 split of its four control computers occurred. Three seconds later, the split became 2-1-1. During troubleshooting, the remaining two computers disagreed (1-1-1-1 split). ***Complete system disagreement.*** But, none of the computers or their intercommunications were faulty! The ***single fault**** was in a box (MDM FA2) that sends messages to the 4 computers via a multi-drop data bus that is similar to the MIL STD 1553 data bus. This fault was a simple crack (fissure) through a diode in the data link interface.

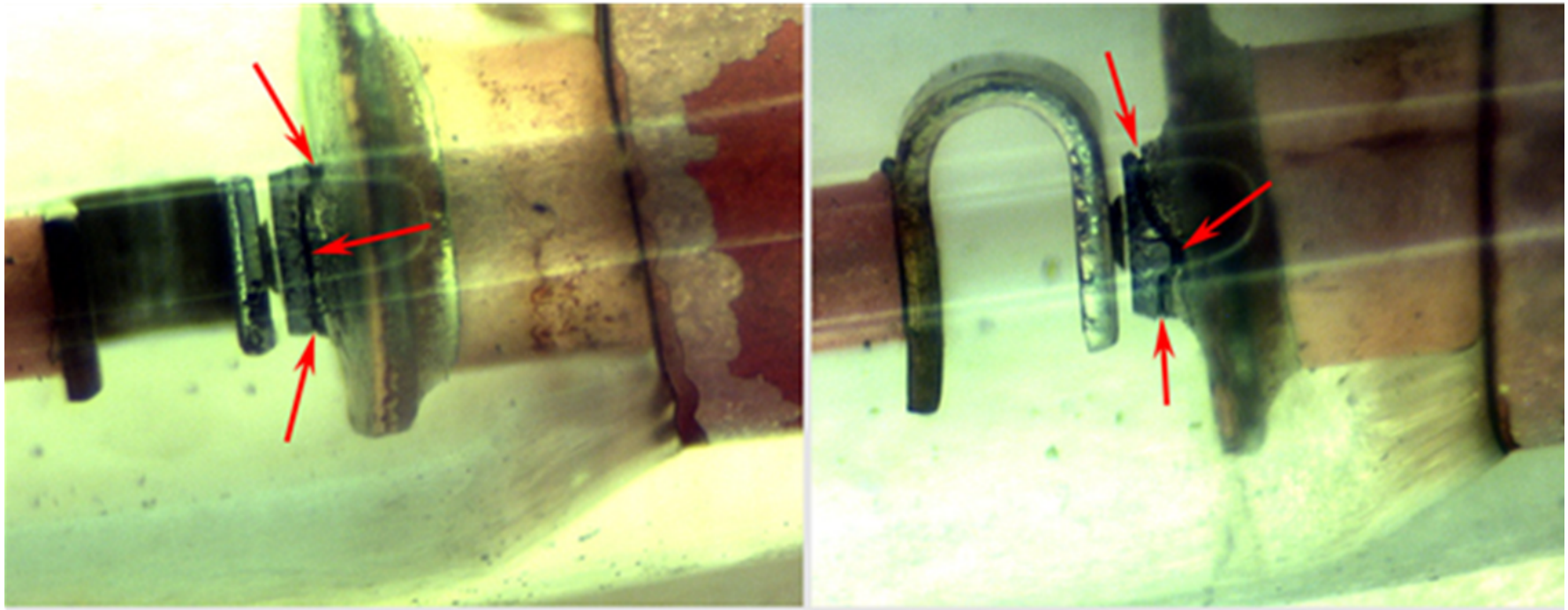
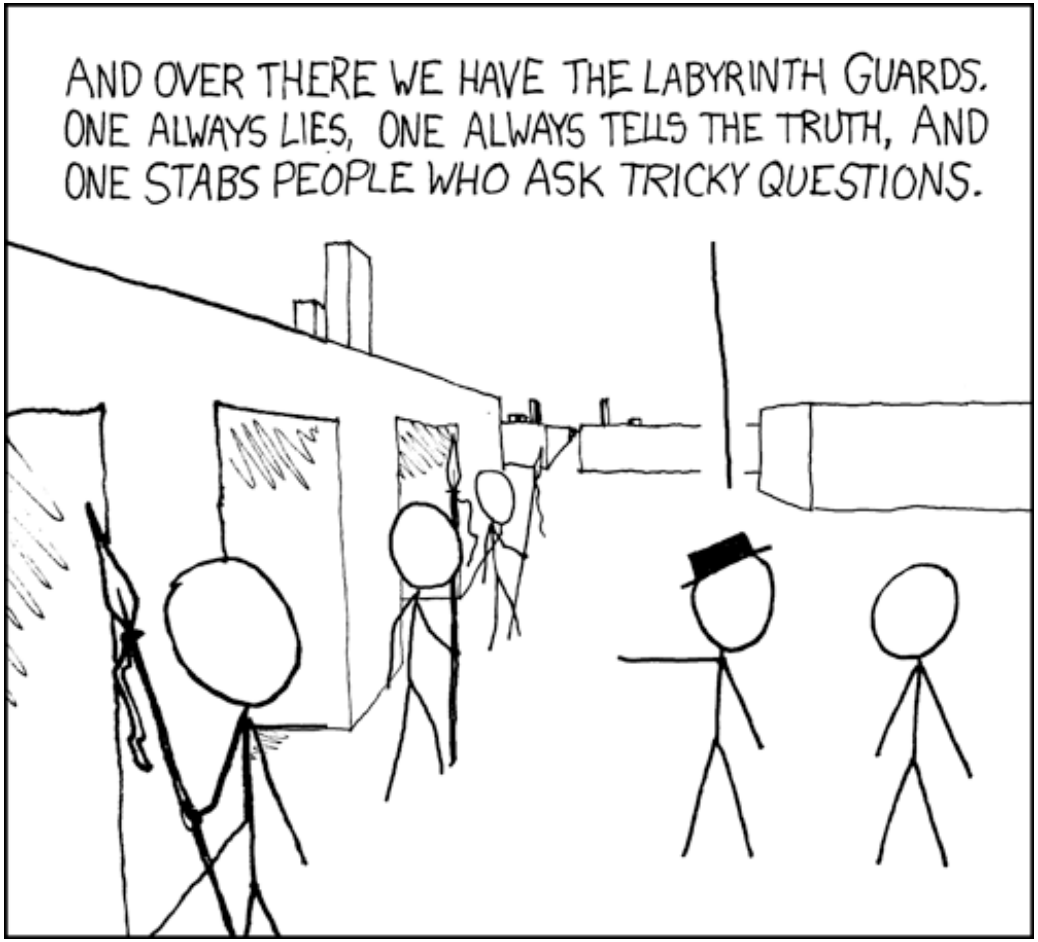


Figure 1. Two views (90 degrees apart) of a fissure that appears to go through the silicon - Red arrows.

* the Byzantine Assassin

From https://c3.nasa.gov/dashlink/static/media/other/ObservedFailures4.html

Questions?



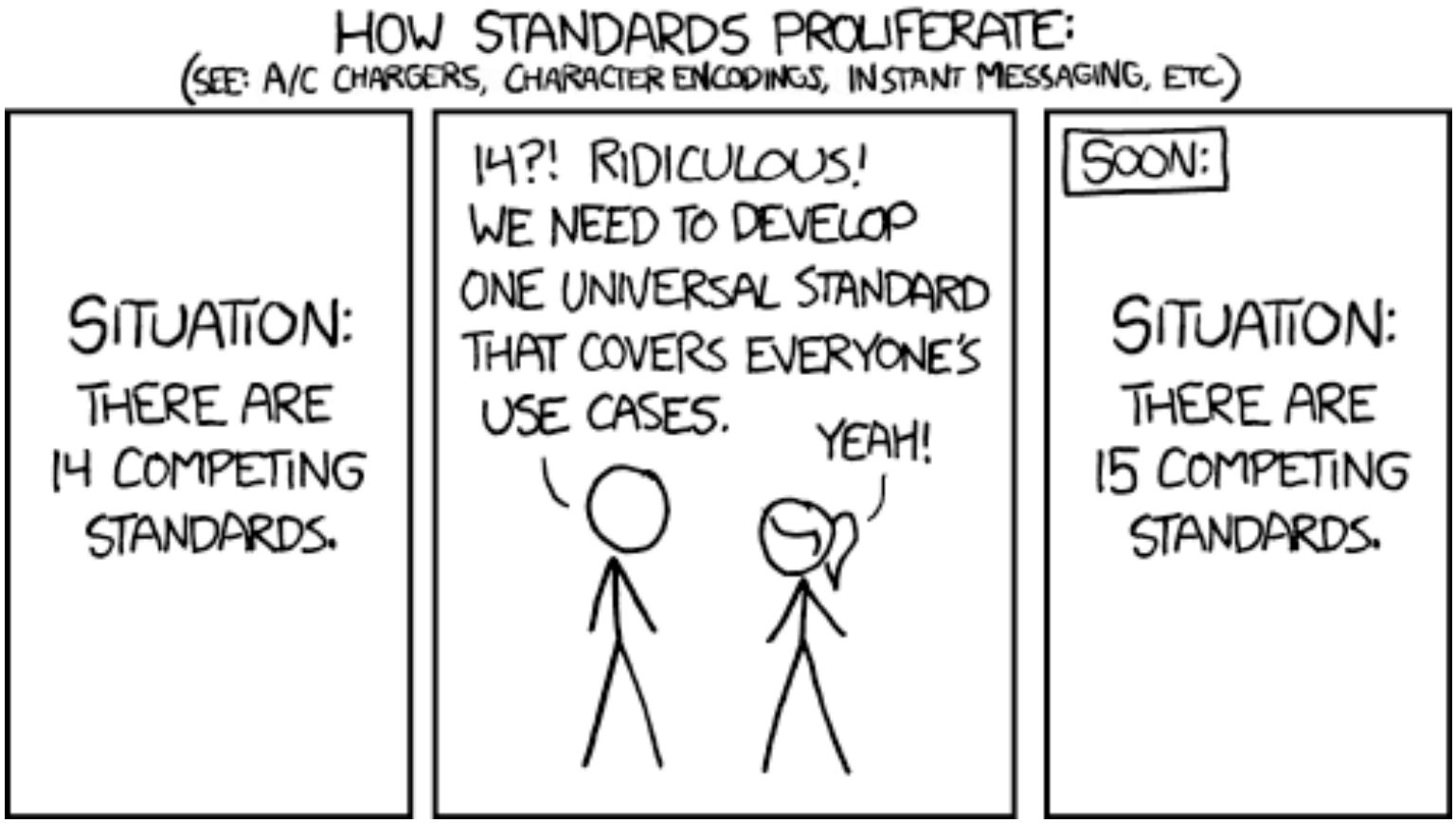
Downloaded from http://xkcd.com/246/

Backup Slides

An assumption will remain valid only until you come to depend on it*.

* http://www.ece.mtu.edu/faculty/rmkieckh/Kieckhafer-top-ten.htm (version 11.1; law 4.2)

On Standards



Downloaded from https://xkcd.com/927/

Design Verification vs. Certification

Design Verification

- Focus on functional correctness, desired properties, and performance
- Emphasis on average case behavior (e.g., for performance)
- Intended interactions between components & environment
  - Presumption that the only interaction is through defined interfaces

Certification

- Focus on non-functional requirements – Safety, Security, etc.
- Emphasis on worst-case behavior
- Preclude adverse interaction between components & environment
  - In addition to failure propagation through defined interfaces, must also consider “out-of-band” failure modes

"How Systems Fail"

Honeywell

- **Assumed importance order**
  - **Assumed/known fault hypothesis violated**
    - exhaustion of resources (known fault hypothesis)
  - **Single point of failure**
    - unknown fault hypothesis
    - forgotten failure mode
    - underestimated probability of occurrence
  - **Fault propagation = domino effect (fault containment)**
- **Real occurrence frequency order**
  - **Chain or domino effect (missing fault containment)**
    - E.g. TTP membership; shown to be a fault propagation path [Ademaj, Sivencrona]
  - **Single point of failure (unknown fault hypothesis)**
    - E.g. quad-redundant control system (termination of bus)[2003]
  - **Exhaustion of resources (known fault hypothesis)**