CHALLENGES IN AUTONOMOUS VEHICLE TESTING AND VALIDATION

(CLIFF'S NOTES VERSION FOR IFIP)

Philip Koopman, Carnegie Mellon University
Michael Wagner, Edge Case Research LLC

Carnegie Mellon University

NREC National Robotics Engineering Center

Electrical & Computer ENGINEERING

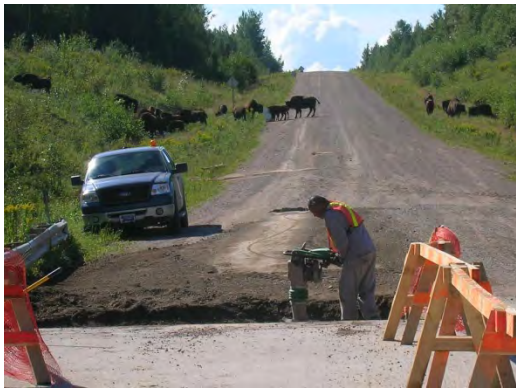
Getting Obvious Cases Covered Is Challenging

Extreme contrast

No lane infrastructure

Poor visibility

Unusual obstacles



Construction

Water (note that it appears flat!)

[Wagner 2014]

Validating High-ASIL Systems via Testing Is Challenging

Need to test for at least ~3x crash rate to validate safety

- Hypothetical fleet deployment: New York Medallion Taxi Fleet
  - 13,437 vehicles, average 70,000 miles/yr = 941M miles/year
    - 7 critical crashes in 2015

    [2014 NYC Taxi Fact Book]

    [Fatal and Critical Injury data / Local Law 31 of 2014]

    ➔ 134M miles/critical crash (death or serious injury)
- Assume testing representative; faults are random independent
  - $R(t) = e^{-lamba*t}$ is the probability of not seeing a crash during testing
- Illustrative: How much testing to ensure critical crash rate is at least as good as human drivers? ➔ (At least 3x crash rate)
  - These are optimistic test lengths…
    - Assumes random independent arrivals
    - Is simulated driving accurate enough?

Testing Miles	Confidence if NO critical crash seen
122.8M	60%
308.5M	90%
401.4M	**95%**
617.1M	99%

Using chi-square test from: http://reliabilityanalyticstoolkit.appspot.com/mtbf_test_calculator

But, Then There Is The Weird Stuff…
(Weirder than any one person can imagine)

http://piximus.net/fun/funny-and-odd-things-spotted-on-the-road
http://blogs.reuters.com/photographers-blog/2012/11/26/house-in-the-middle-of-the-road/
http://edtech2.boisestate.edu/robertsona/506/images/buffalo.jpg

Machine Learning Might Be Brittle & Inscrutable

Legibility: can humans understand how ML works?

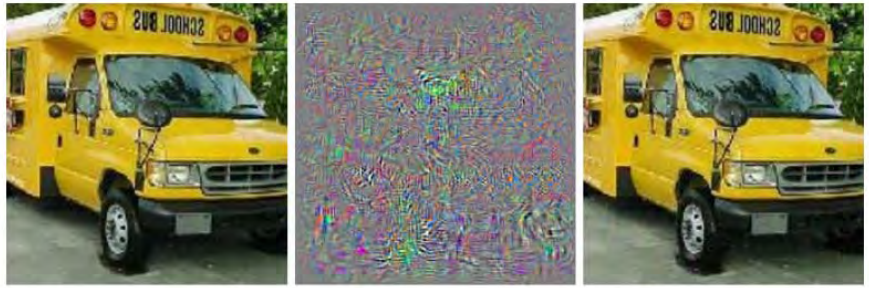
- Machine Learning “learns” from training data
  - Result is a weighted combination of “features”
- Commonly the weighting is inscrutable, or at least not intuitive
  - There is an unknown (significant?) chance results are brittle
    - E.g., accidental correlations in training data, sensitivity to noise

QuocNet:

Car | **Not a Car** | *Magnified Difference*

AlexNet:

Bus | *Magnified Difference* | **Not a Bus**

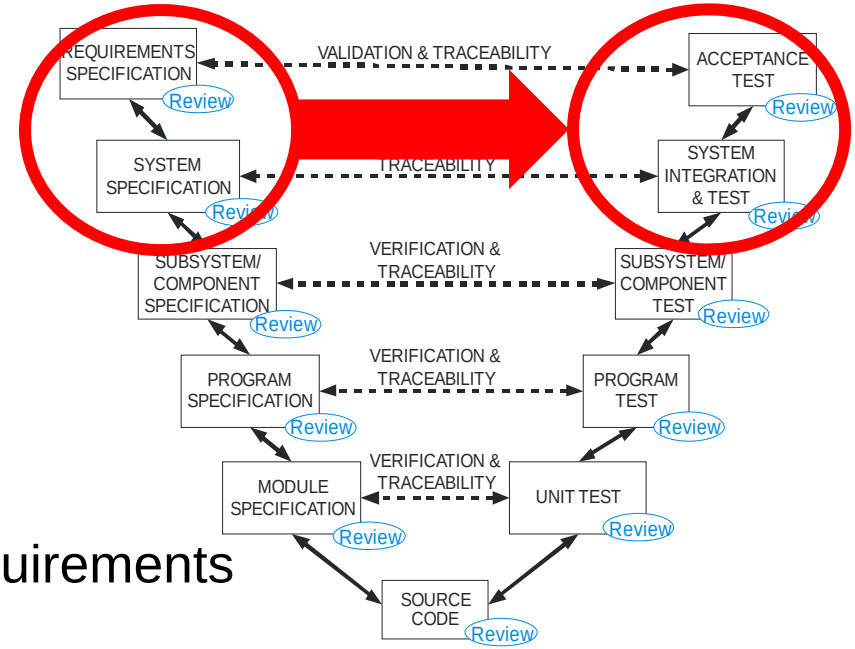


Szegedy, Christian, et al. "Intriguing properties of neural networks." *arXiv preprint arXiv:1312.6199* (2013).

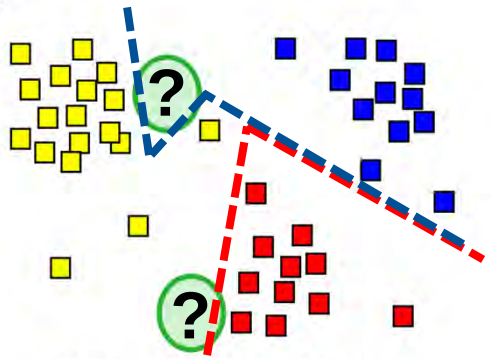
Where Are the Requirements for Machine Learning?

Machine Learning requirements are the training data

- V model traces reqts to V&V
- Where are the requirements in a machine learning based system?
  - ML system is just a framework
  - The **training data** forms de facto requirements



- How do you know the training data is "complete"?
  - Training data is safety critical
  - What if a moderately rare case isn't trained?
    - It might not behave as you expect
    - People's perception of "almost the same" does not necessarily predict ML responses!

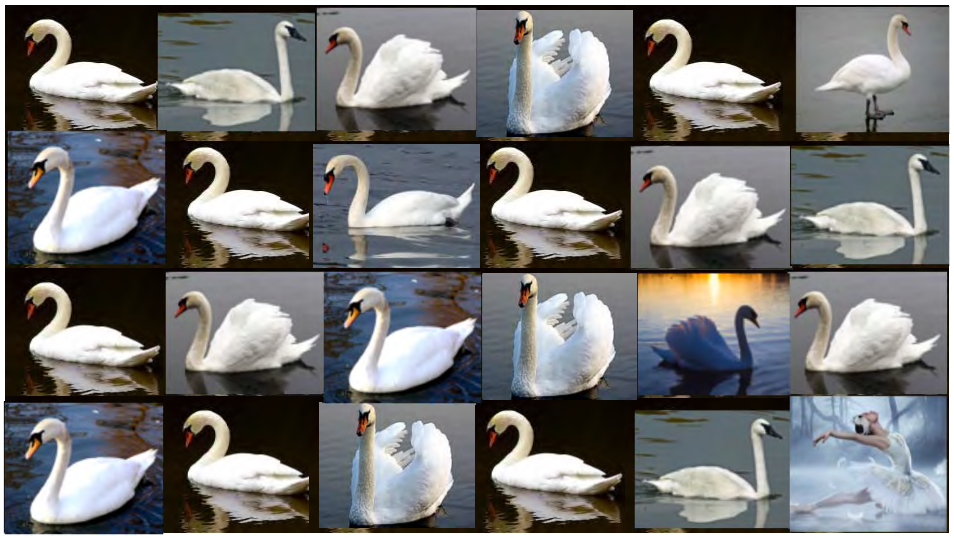


Cluster Analysis

The Black Swan Meets Autonomous Vehicles

Suggested Philosophy for Testing Autonomous Vehicles:

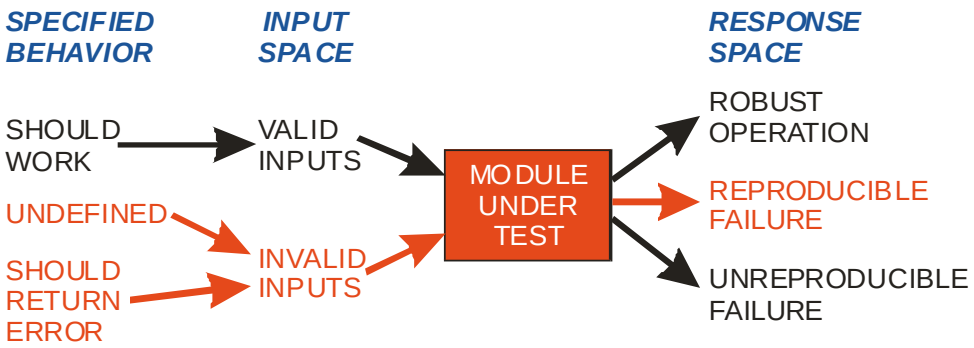
- Some testing should look for proper functionality
  - But, some testing should attempt to ***falsify a correctness hypothesis***
- Much of vehicle autonomy is based on Machine Learning
  - ML is inductive learning… which is vulnerable to "black swan" failures
  - We've found robustness testing to be useful in this role



Thousands of miles of "white swans"…

Make sure to fault inject some "black swans"

ASTAA:
Automated Stress Testing of Autonomy Systems



+

Ballista Stress-Testing Tool

Robustness testing of defined interfaces

- Most test cases are exceptional
- Test cases based on best-practice software testing methodology
- Detects software hanging or crashing

Earlier work looked at stress-testing COTS operating systems

Uncovered system-killer crash vulnerabilities in top-of-the-line commercial operating systems

NREC Safety Monitor

Monitors safety invariants at run-time

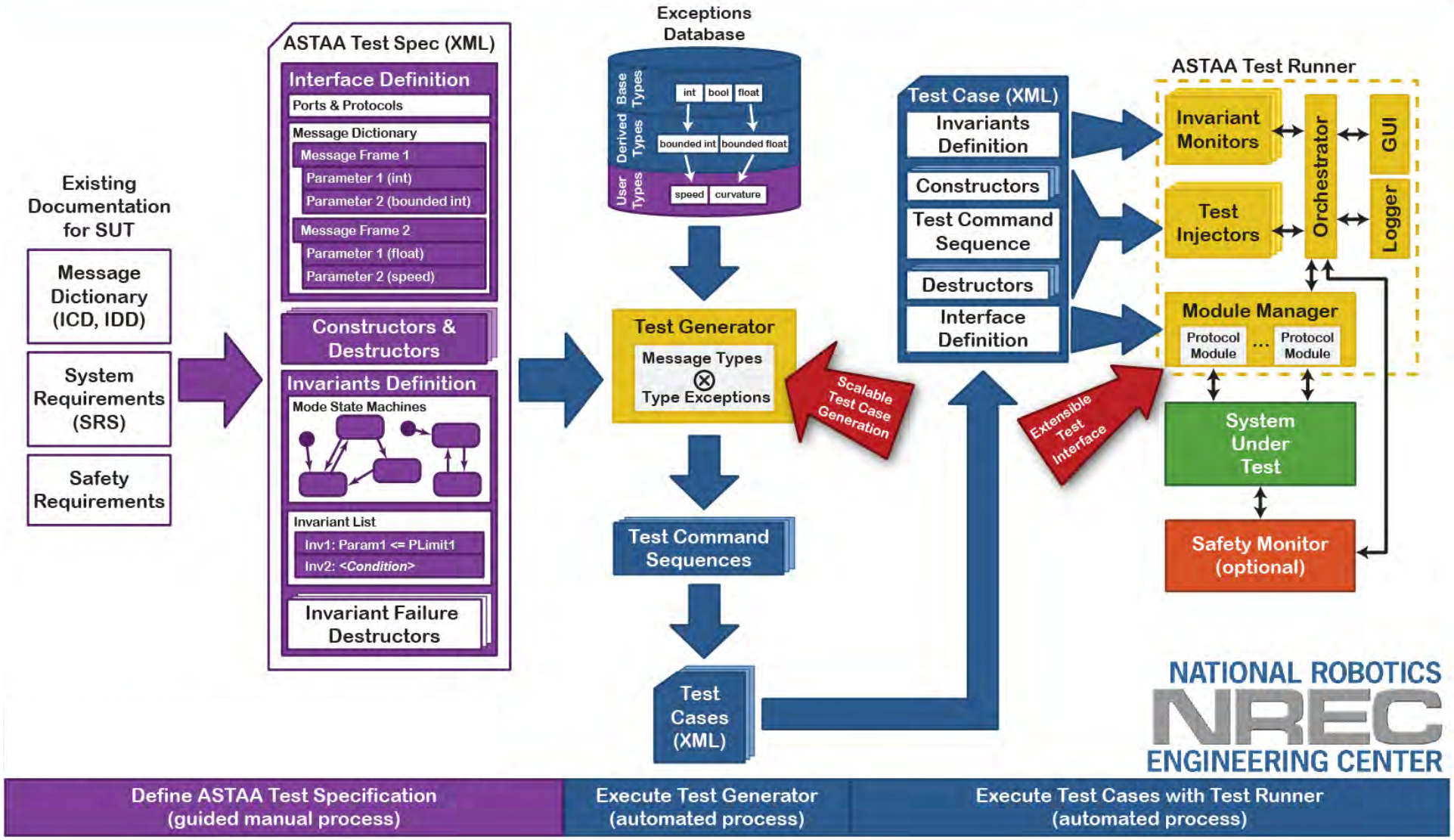
- Designed as run-time safety shutdown box for UAS applications

Independently senses system state to determine whether invariants are violated

Firewalls safety-criticality into a small, manageable subset of a complex UAS; prototype deployed on Autonomous Platform Demonstrator (APD), a 9-ton UGV capable of reaching 80 km/hr

Automated Stress-Testing for Autonomy Architectures

Test Specification and Execution Overview



Example Autonomous Vehicle Defects Found via Robustness Testing

ASTAA Project at NREC found system failures due to:

Improper handling of floating-point numbers:

- Inf, NaN, limited precision

Array indexing and allocation:

- Images, point clouds, etc…
- Segmentation faults due to arrays that are too small
- Many forms of buffer overflow, especially dealing with complex data types
- Large arrays and memory exhaustion

ROS

Time:

- Time flowing backwards, jumps
- Not rejecting stale data

Problems handling dynamic state:

- For example, lists of perceived objects or command trajectories
- Race conditions permit improper insertion or removal of items
- Vulnerabilities in garbage collection allow memory to be exhausted or execution to be slowed down