Carnegie Mellon University
Software Engineering Institute

Verification and Validation in Artificial Intelligence

Eliezer Kanal

Technical Manager, CERT

Software Engineering Institute
Carnegie Mellon University
Pittsburgh, PA 15213

[DISTRIBUTION STATEMENT A] This material has been approved for public release and unlimited distribution. Please see Copyright notice for non-US Government use and distribution.

CMU SEI – a DoD Federally Funded Research and Development Center (FFRDC)

- Our mission: Engineering and securing software
- Established in 1984 at Carnegie Mellon University
- ~700 employees
- Offices in Pittsburgh and DC, with locations near customer facilities in MA, MD, TX, and CA
- ~$145M in annual funding (~$20M USD(R&E) 6.2 and 6.3 Line funding)

AI V&V in a nutshell

Motivating question:

"How can I gain confidence in my AI technique?"

Expectation:

- Metrics
- Algorithms

Finding:

- V&V is a Comp Sci concept
- Multiple *fields* of research, multiple lines in each field
- Still need some definitional work

V&V – Definition

3.4539

verification and validation (V&V)

Process of determining whether:

1. the requirements for a system or component are complete and correct,
2. the products of each development phase fulfill the requirements or conditions imposed by the previous phase, and
3. the final system or component complies with specified requirements

V&V – Definition

Verification – confirmation, through the provision of objective evidence, that specified requirements have been fulfilled

Validation – confirmation, through the provision of objective evidence, that the requirements for a specific intended use or application have been fulfilled

Said differently:

Verification: Did I solve the problem correctly?

Validation: Did I solve the correct problem?

Early AI V&V – Expert Systems

Early AI practitioners borrowed concepts from contemporary Computer Science literature [citation needed]

Expert Systems:

> "An expert system is a computer program that embodies **expertise** about a particular domain, and can use **symbolic reasoning** techniques to solve problems in this domain; problems that would need the assistance of a human expert in the real world. "

Expertise ➞ Knowledge Representation (e.g., RDF triplets)

Symbolic reasoning ➞ Logic

M. J. Rijckaert, V. Debroey, and W. Bogaerts, "Expert Systems: The State of the Art," in *Mathematical Models for Decision Support*, Springer Berlin Heidelberg, 1988, pp. 487–517.

Early AI V&V – Expert Systems

Frank has the job of student

object attribute value

Verification: Is the data & logic internally consistent?

e.g., Redundant rules:

$$\mathtt{PROFESSOR}(x) \rightarrow \mathtt{HAS\_DEGREE}(x, PhD)$$
$$\mathtt{HAS\_DEGREE}(x, PhD) \rightarrow \mathtt{HAS\_DEGREE}(x, BSc)$$
$$\mathtt{HAS\_DEGREE}(x, BSc) \rightarrow \mathtt{GRADUATE}(x)$$
$$\mathtt{PROFESSOR}(x) \rightarrow \mathtt{GRADUATE}(x)$$

A. D. Preece, R. Shinghal, and A. Batarekh, "Principles and practice in verifying rule-based systems," *Knowl. Eng. Rev.*, vol. 7, no. 2, pp. 115–141, Jun. 1992.

Early AI V&V – Expert Systems

Ambivalence:

For example, assume $E = \{$GRADUATE(x), UNDERGRAD$(x)\}$ is an impermissible set, and we have the following rules:

$$\text{ENROLLED\_IN}(x,y) \wedge \text{GRAD\_COURSE}(x) \rightarrow \text{GRADSTUDENT}(y)$$

$$\text{ENROLLED\_IN}(x,y) \wedge \text{UNDERGRAD\_COURSE}(x) \rightarrow \text{UNDERGRAD}(y)$$

$$\text{GRAD\_COURSE}(CS661)$$

$$\text{UNDERGRAD\_COURSE}(CS445)$$

If the environment $\{$ENROLLED_IN$(CS661, Marvin)$,ENROLLED_IN$(CS445, Marvin)\}$ is a permissible environment, we would be able to infer the set:

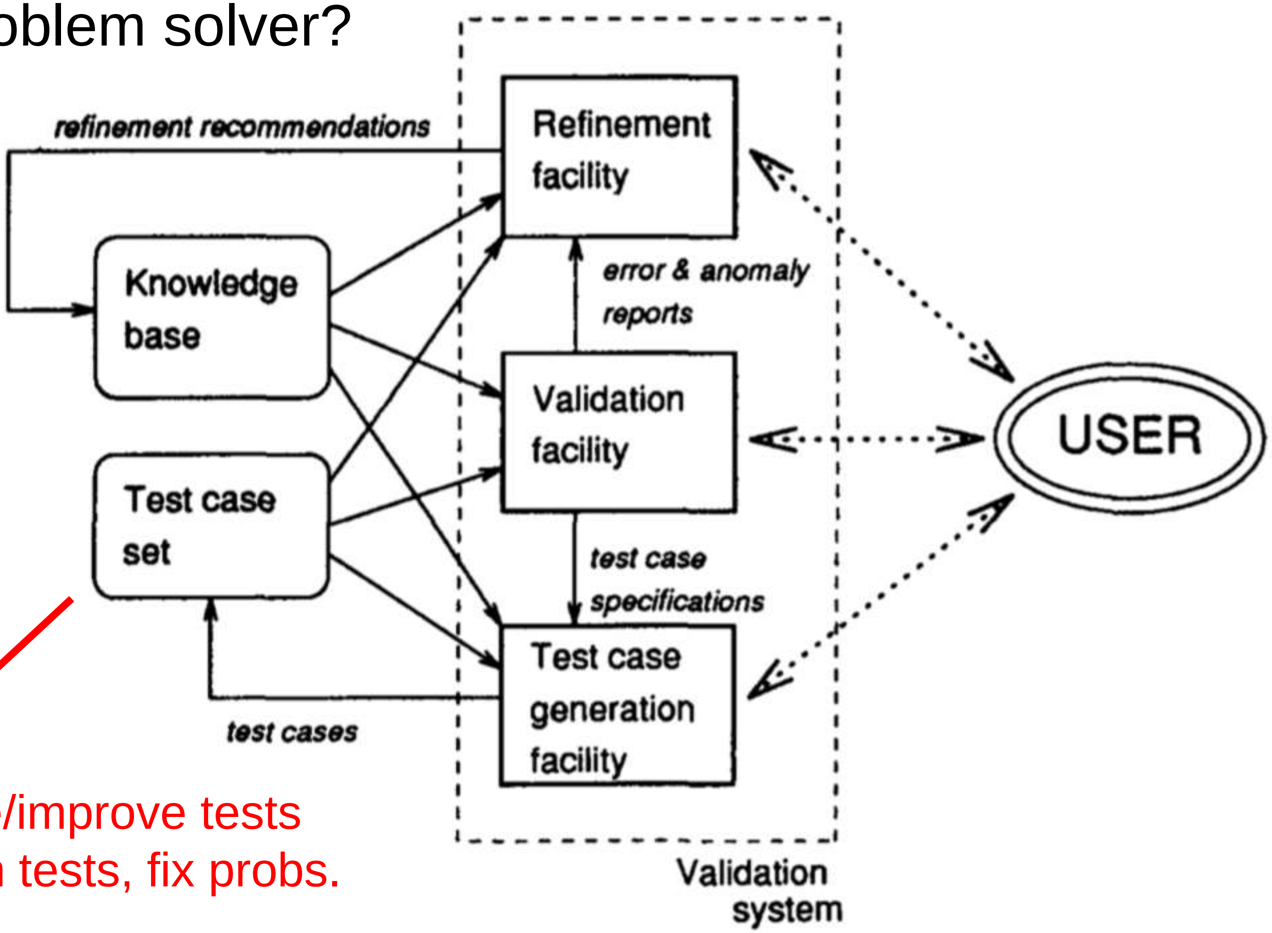
$$\{\text{GRADSTUDENT}(Marvin), \text{UNDERGRAD}(Marvin)\}$$

This is the impermissible set $E\sigma$ where $\sigma = \{Marvin/x\}$.

A. D. Preece, R. Shinghal, and A. Batarekh, "Principles and practice in verifying rule-based systems," *Knowl. Eng. Rev.*, vol. 7, no. 2, pp. 115–141, Jun. 1992.

Early AI V&V – Expert Systems

Validation: Does the expert system serve it's function as a problem solver?



- Static – create/improve tests
- Dynamic – run tests, fix probs.

N. Zlatareva and A. Preece, "State of the art in automated validation of knowledge-based systems," *Expert Syst. Appl.*, vol. 7, no. 2, pp. 151–167, Apr. 1994.

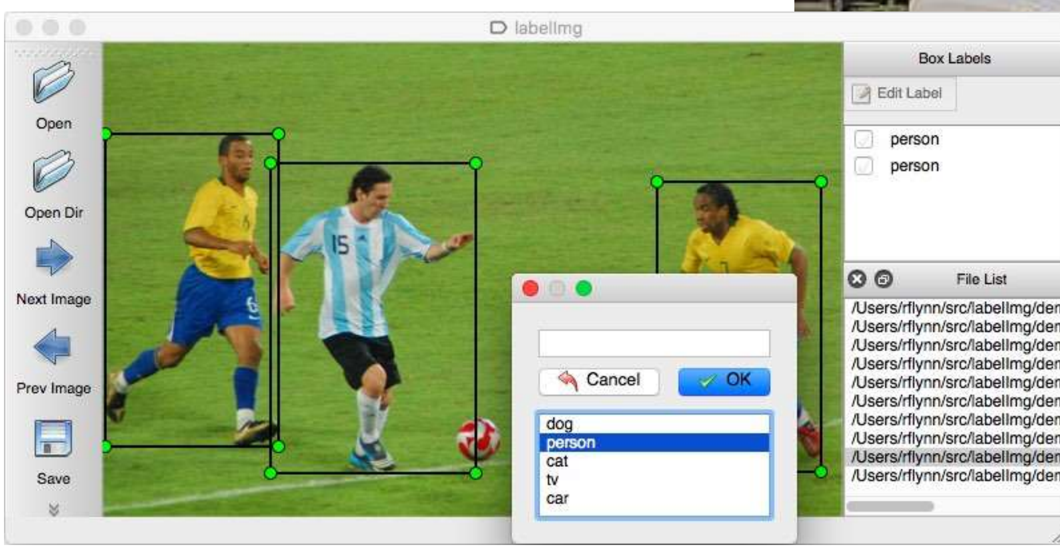
Early AI V&V – Expert Systems

Many tools created to support these activities

- SEEK, SEEK2
- Rule Checker Program
- EVA
- VVR
- EITHER
- GTM
- ONCOCIN
- KB-Reducer
- CHECK
- RCP
- COVER

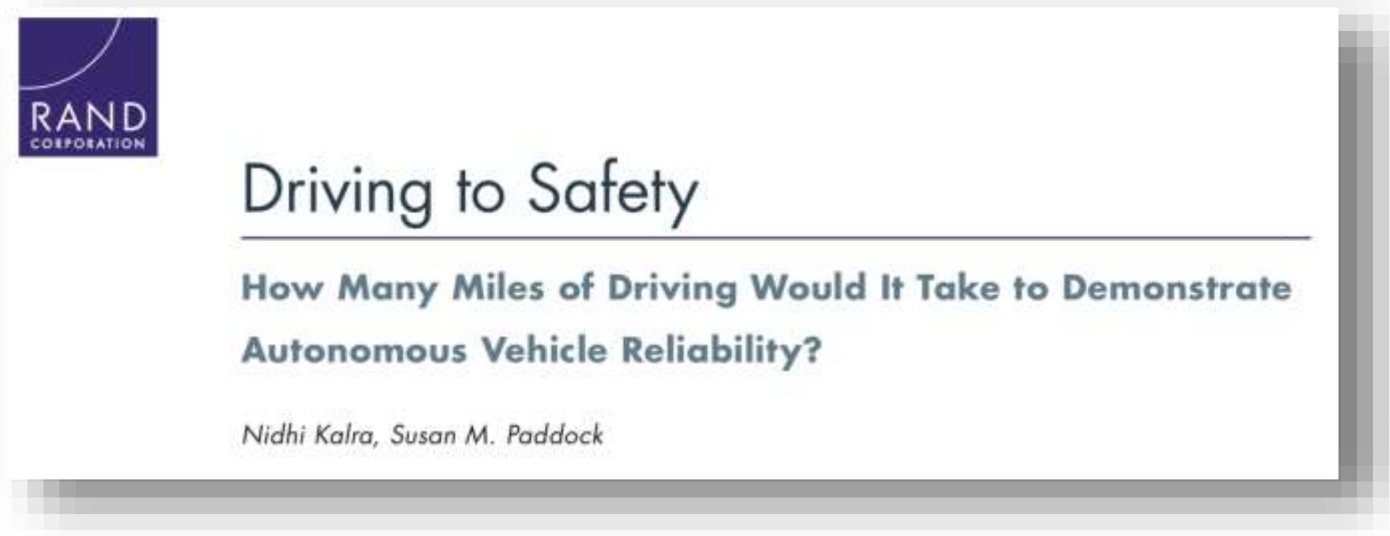
A. D. Preece, R. Shinghal, and A. Batarekh, "Principles and practice in verifying rule-based systems," *Knowl. Eng. Rev.*, vol. 7, no. 2, pp. 115–141, Jun. 1992.
N. Zlatareva and A. Preece, "State of the art in automated validation of knowledge-based systems," *Expert Syst. Appl.*, vol. 7, no. 2, pp. 151–167, Apr. 1994.

AI Everywhere!



deepstack.io
WikiMedia foundation, various images
Tzutalin. LabelImg. Git code (2015). https://github.com/tzutalin/labelImg

Traditional techniques are impractical


RAND
CORPORATION

Driving to Safety

How Many Miles of Driving Would It Take to Demonstrate Autonomous Vehicle Reliability?

Nidhi Kalra, Susan M. Paddock

Common assurance techniques:

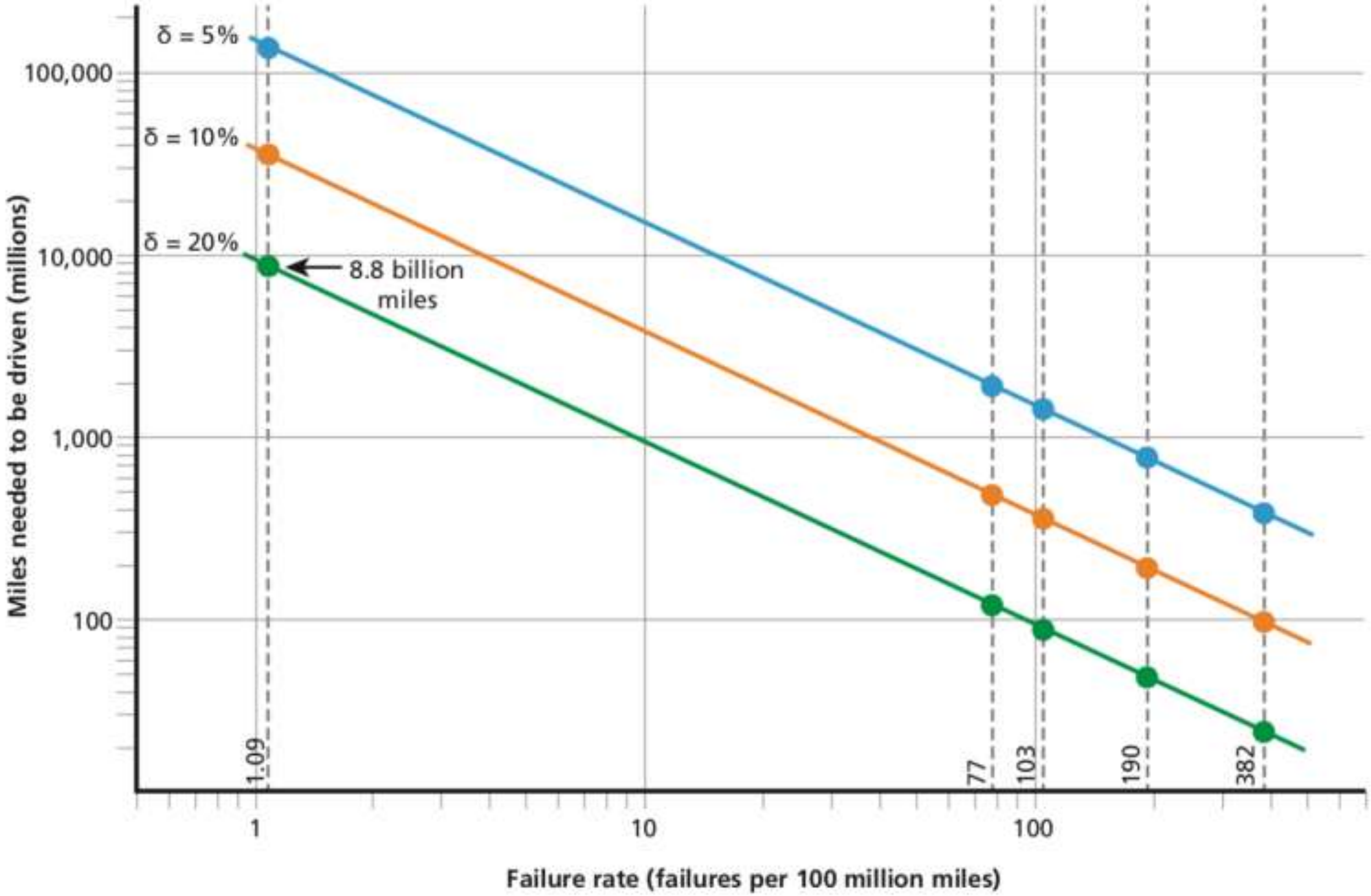
- Power analysis
- 95% confidence interval
- Survival analysis
- Success run statistics

Impractical given:

– Rare anomalies
– Very high cost of failure

N. Kalra, S. M. Paddock, and Rand Corporation, *Driving to safety : how many miles of driving would it take to demonstrate autonomous vehicle reliability?*

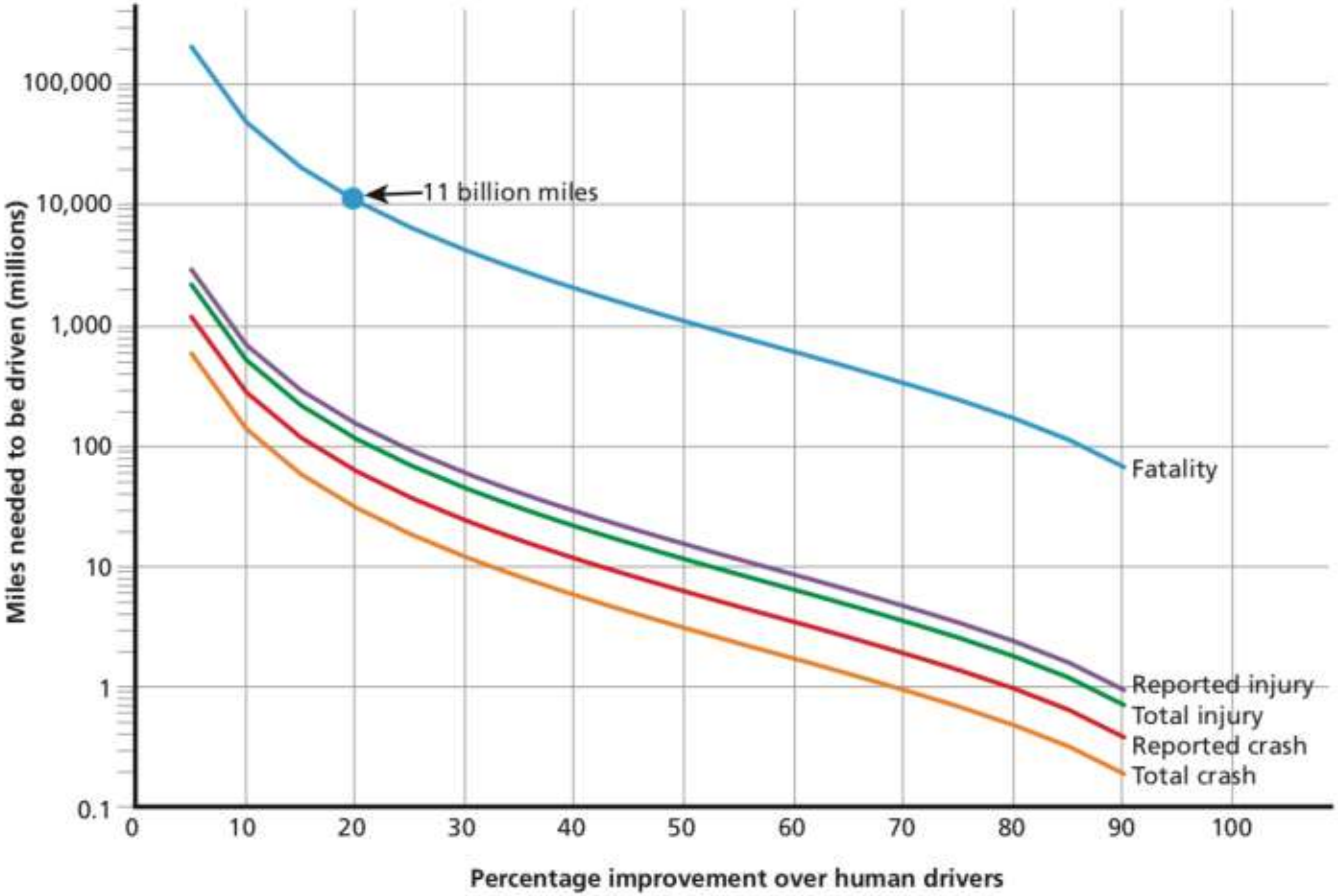
Figure 2. Miles Needed to Demonstrate Failure Rates to a Particular Degree of Precision



SOURCE: Authors' analysis.
NOTE: These results use a 95% CI. The three colored lines show results for different levels of precision δ, defined as the size of the CI as a percent of the failure rate estimate. The five dashed vertical reference lines indicate the failure rates of human drivers in terms of fatalities (1.09), reported injuries (77), estimated total injuries (103), reported crashes (190), and estimated total crashes (382).
RAND *RR1478-2*

N. Kalra, S. M. Paddock, and Rand Corporation, *Driving to safety : how many miles of driving would it take to demonstrate autonomous vehicle reliability?*

Figure 4. Miles Needed to Demonstrate with 95% Confidence and 80% Power that the Autonomous Vehicle Failure Rate Is Lower than the Human Driver Failure Rate



SOURCE: Authors' analysis.
NOTE: The results depend upon the estimated failure rate of autonomous vehicles. This is shown on the horizontal axis and defined as a percent improvement over the human driver failure rate. The comparison can be made to the human driver fatality rate (blue line), reported injury rate (purple line), estimated total injury rate (green line), reported crash rate (red line), and estimated total crash rate (orange line).

RAND *RR1478-4*

N. Kalra, S. M. Paddock, and Rand Corporation, *Driving to safety : how many miles of driving would it take to demonstrate autonomous vehicle reliability?*

Modern approaches: Highly varied

Traditional CS tools

- Static analysis
- Dynamic analysis
- Model checking
- Hybrid systems
- …

Application-specific techniques

- Self-driving cars, aircraft
- Energy
- Medicine
- …

Algorithm-specific techniques:

- Neural nets
- Bayes networks
- MapReduce
- …

Specific recent attention on Adversarial AI has generated a more broad literature

Model checking

Well-established techniques, but difficult to apply to large systems

Entire field of identifying optimal search algorithms

Applicable in a vast number fields:

- Software consistency checks
- SpaceX
- Self-driving cars
- Game playing (chess, Go, StarCraft, etc)
- …

Model checking

Overview of approach:

1. Define a system as $H(\sim)$, where ~ can include location l, arbitrary variables $\mathbf{x}$, functions to transition between states T, etc
2. Define state as the tuple $(l, \mathbf{x})$ describing the system at a given location and with specific values for variables
3. Given a possible set of error states $\boldsymbol{S}_{error}$, want to identify all states $s_\varepsilon \in \boldsymbol{S}_{error}$ that can be reached from s

S. Bogomolov, G. Frehse, R. Grosu, H. Ladan, A. Podelski, and M. Wehrle, Springer, Berlin, Heidelberg, 2012, pp. 479–494.

A Box-Based Distance between Regions for Guiding the Reachability Analysis of SpaceEx

Sergiy Bogomolov[1], Goran Frehse[2], Radu Grosu[3], Hamed Ladan[1], Andreas Podelski[1], and Martin Wehrle[1,4]

Given two states $s = (l, \mathbf{x})$ and $s' = (l', \mathbf{x}')$, can define a trajectory as a set of discrete states connecting s and s' at times $t_{0 \ldots n}$.

Instead of modeling entire trajectory, segment space into boxes and calculate state at a few representative points

- Orders of magnitude faster in identifying error states
- If no reachable error states, similar simulation time as breadth- or depth-first search

S. Bogomolov, G. Frehse, R. Grosu, H. Ladan, A. Podelski, and M. Wehrle, Springer, Berlin, Heidelberg, 2012, pp. 479–494.

Exploiting Learning and Scenario-based Specification Languages for the Verification and Validation of Highly Automated Driving

Werner Damm*
OFFIS
Oldenburg, Germany
werner.damm@offis.de

Roland Galbas
Robert Bosch GmbH, Chassis Systems Control
Abstatt, Germany
Roland.Galbas@de.bosch.com

Hybrid systems (continuous & discrete modeling) useful for modeling system, far too large feature space

Use *pattern database:*

- (1) Define initial state s_t, goal as desired state(s)
- Calculate all possible s_{t+1} states
- Drop states that don't get us significantly closer to the goal
- For the rest, iterate (3) and (4) until find a solution

W. Damm and R. Galbas, *2018 IEEE/ACM 1st International Workshop on Software Engineering for AI in Autonomous Systems (SEFAIAS)*, 2018.

Model checking

Hybrid systems fail if discrete steps too large (e.g., binary)…
pattern database improves search time by orders of magnitude

W. Damm and R. Galbas, *2018 IEEE/ACM 1st International Workshop on Software Engineering for AI in Autonomous Systems (SEFAIAS)*, 2018.

Improved simulations

Decades-old technology, widely used

Advances focused on accurate modeling of subsystems, specific regions of probability space, interactions between systems, merging virtual and real

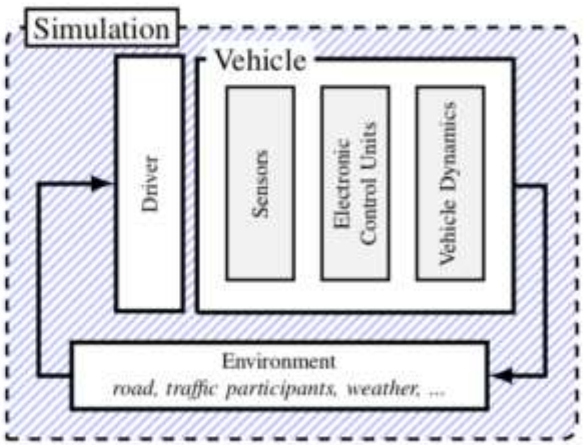
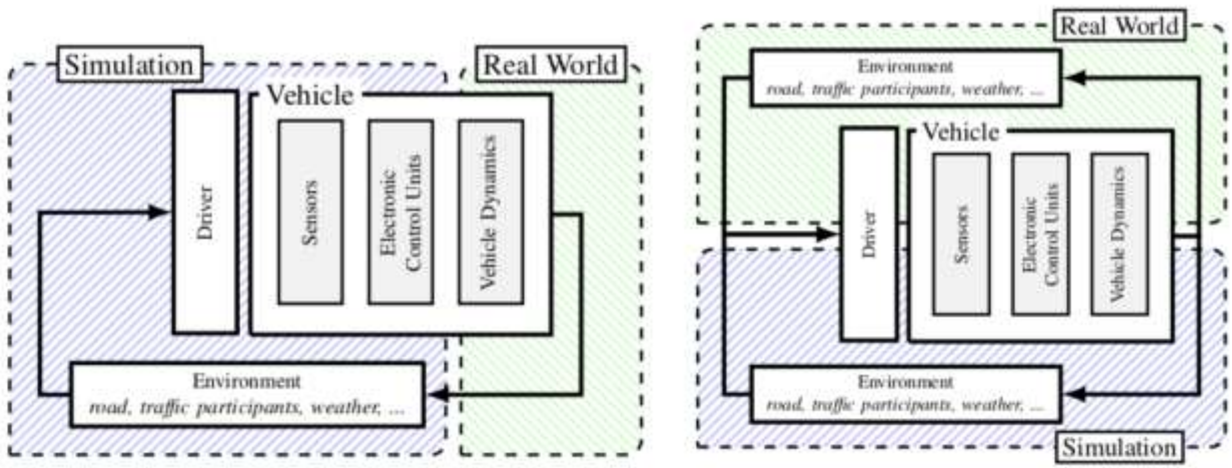


Fig. 6: An ADAS simulation includes models of the environment, the human driver, sensors and the vehicle dynamics.



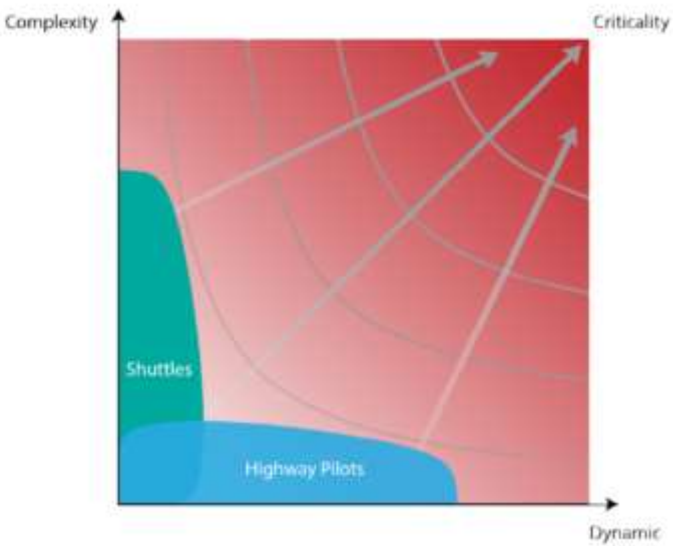
(a) X-in-the-loop: Hardware components are connected to the virtual environment.

(b) Measured and simulated environmental aspects are augmented and aligned in order to test ADAS on both worlds.

Fig. 7: Different concepts of combining measurements and simulations.

J. E. Stellet, M. R. Zofka, J. Schumacher, T. Schamm, F. Niewels, and J. M. Zollner, , *2015 IEEE 18th International Conference on Intelligent Transportation Systems*, 2015, pp. 1455–1462.

Improved simulations



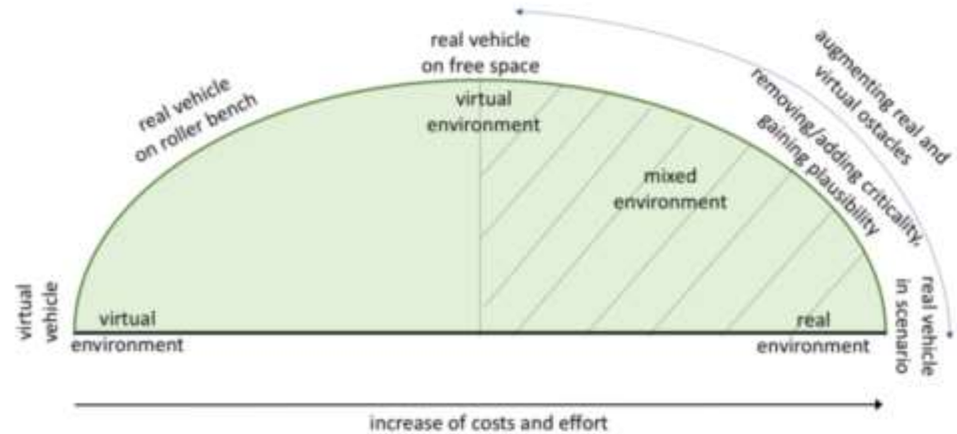


Fig. 2. Typically, a SuT is tested within different instances of the *X-in-the-loop* paradigm (left side). Sleepwalker transfers this idea to static and dynamic aspects of the environment, gradually varying them from reality to virtuality, while the vehicle is tested physically in the loop (right side).

M. R. Zofka, M. Essinger, T. Fleck, R. Kohlhaas, and J. M. Zollner, *2018 IEEE International Conference on Simulation, Modeling, and Programming for Autonomous Robots (SIMPAR)*, 2018, pp. 151–157.
W. Damm and R. Galbas, *2018 IEEE/ACM 1st International Workshop on Software Engineering for AI in Autonomous Systems (SEFAIAS)*, 2018.

Cross-validation

It’s easy

It’s well-understood

Compute power is cheap

Great for non-critical systems

If it ain’t broke…

Note to self: mention conversation with CMU professor

Note to everyone else: sorry about that previous note

Adversarial AI

Nascent field, exploding literature

Attacks ➞ Defense ➞ Broken defense

SoK: Towards the Science of Security and Privacy in Machine Learning

Nicolas Papernot*, Patrick McDaniel*, Arunesh Sinha†, and Michael Wellman†
* Pennsylvania State University
† University of Michigan

SEI Early Efforts, Continued work

Towards a Mathematical Definition of Robustness for Machine Learning Algorithms

Zachary Kurtz, Ritwik Gupta, Eliezer Kanal, Matthew Gaston
Software Engineering Institute
Carnegie Mellon University
Pittsburgh, PA 15213
{ztkurtz, ekanal}@cert.org, {rgupta, megaston}@sei.cmu.edu

Rigorously define “robustness”, abstract of specific techniques or implementations

Different definitions needed when referring to different attacks… “robust against what?”

Contact Us

Carnegie Mellon University

Software Engineering Institute

4500 Fifth Avenue

Pittsburgh, PA 15213-2612

412-268-5800

888-201-4479

info@sei.cmu.edu

www.sei.cmu.edu