T&E of AI and Autonomy: An Assurance Case Framework Version 2.0

David Tate

Institute for Defense Analyses

dtate@ida.org

June 2021

BLUF: This is hard but solution methods already exist

Test and evaluation of AI-enabled and autonomous defense systems (ADS) cannot in general be accomplished using historical approaches and tools.

However, there is a framework for analyzing T&E requirements and planning T&E support that potentially addresses all of the new challenges.

Getting people to adopt this framework and organize their T&E efforts accordingly will not be easy.

The goal is assured effectiveness and dependability

Advanced capabilities don't help if we're not sufficiently confident to field and employ the systems.

There will always be **some** kind of certification or licensure or acceptance testing process.

There may be multiple certification authorities (e.g., Safety, Cybersecurity, Effectiveness, Reliability).

State of the Art: Assurance Cases

An ***assurance case*** is a structured argument that the system is sufficiently dependable to permit fielding in a defined operational context.

Existing standards and regulatory bodies **already require** explicit assurance cases for complex systems:

- Safety cases (oldest, most mature literature)
- Software assurance cases (cybersecurity, reliability)
- Robustness cases

Currently, these efforts are generally stovepiped.

Each dimension of assurance generates attack surface

If your system can become unsafe, that's a problem

If your system can become unreliable, that's a problem

If your system might fail the mission, that's a problem

If adversaries can hack or spoof your system…

Is cyber really different from other assurance for ADS?

From a tester's point of view, are adversarial threats importantly different from other threats to dependability?

Is robustness to potential adversary actions importantly different from robustness to environmental complexity?

Claim:

Cyber assurance is just one dimension of assurance in ADS, and is best treated as part of a holistic assurance case

The capabilities that enable autonomy are…

Perception

Reasoning

Planning / Deciding

Learning

Self-organizing behavior

Human-Machine Teaming (HMT)

The technologies that enable these include…

Machine learning

Computer vision

Sensor fusion

Knowledge representation

Inference engines

Path planning

Optimization

Expert systems

HMT CONOPS

●
●
●

How do these generate ~~attack~~ assurance surfaces?

"Assurance surfaces" arise from the **inputs** to these technologies and capabilities…

Perception : sensors, algorithms, stored data, training
Reasoning : world model, ontology, algorithms
Planning : world model, stored data, algorithms
Learning : ***training data*** , scoring, architecture, updates
Self-organizing : world model, CONOPS, sensors/ comms
HMT: messages sent/ received, world model, CONOPS

Assurance cases require both *evidence* and *arguments*

A pile of evidence is not an argument.

An argument without evidence is unconvincing.

The wrong evidence doesn't help.

The outputs of TEV&V should be the evidence that supports the needed assurance cases.

Where does the evidence come from?

Traditional assurance cases are based on a combination of ***empirical*** , ***formal*** , and ***process*** evidence:

~~Exhaustive testing~~

Formal verification

Design of experiments

Run-time monitors

Human in the loop +training

CMMI level, ISO 9000, etc.

Assurance case development tools already exist

- Assurance Case Editor (**ACedit**)
  https://code.google.com/archive/p/acedit/
- NASA Assurance Case Automation Toolset (**AdvoCATE**)
  https://ti.arc.nasa.gov/tech/rse/research/advocate/
- Evidence Confidence Assessor (**EviCA**)
  https://www.youtube.com/watch?v=Rz_P0YlMPBU
- **Astah GSN** (commercial product, see astah.net)
  https://astah.net/products/astah-gsn/

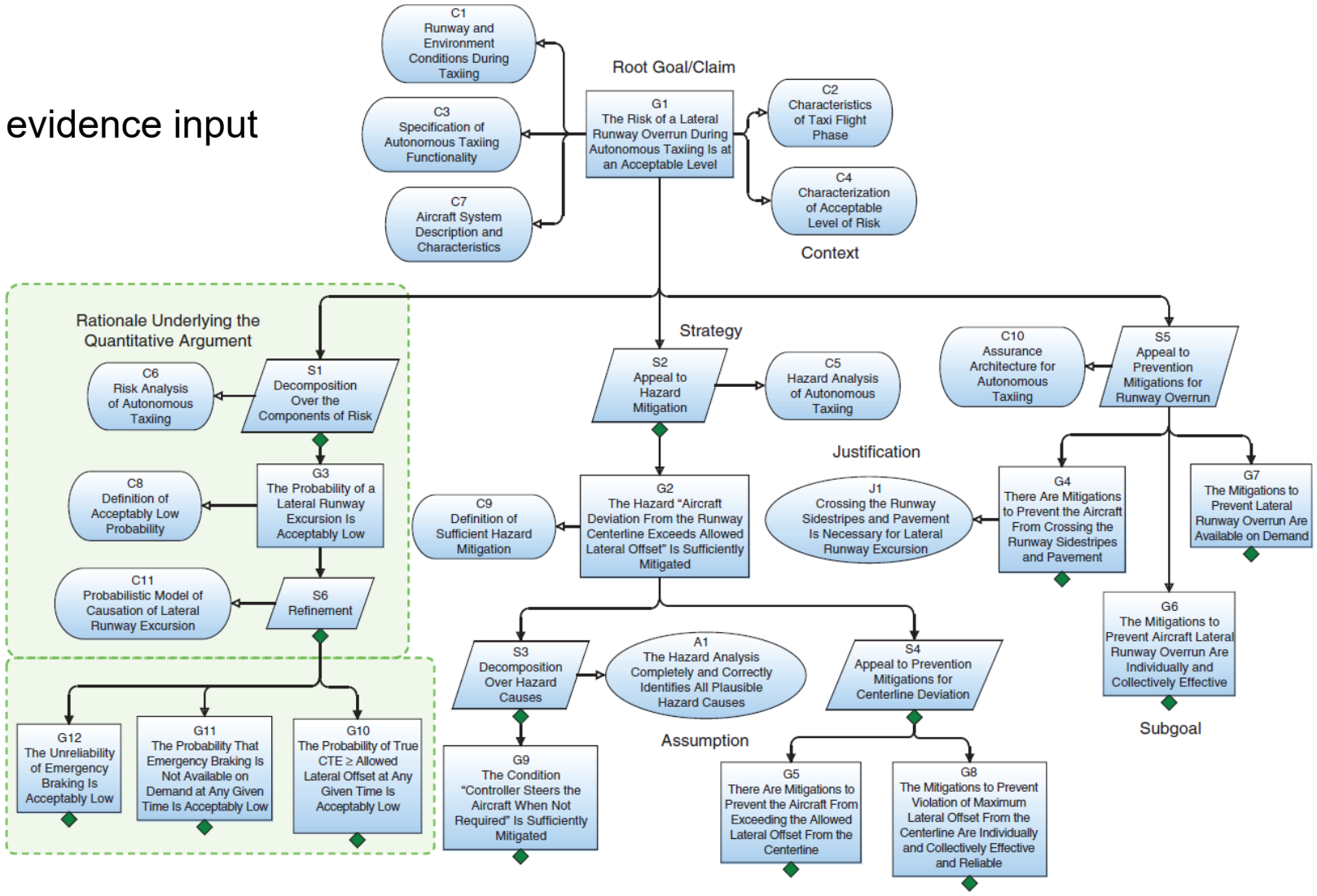
Each tool uses **Goal Structuring Notation** (GSN) as the graphical language for describing and manipulating arguments.

Reference: *Tool Support for Assurance Case Development*, Ewen Denney and Ganesh Pai, NASA Ames Research Laboratory

Example: A Partial Safety Case in GSN

From "Dynamic Assurance Cases: A Pathway to Trusted Autonomy," Asaadi, Denny, Menzies, Pai, Petroff

◆ denotes evidence input



Approved for public release; distribution is unlimited.

What evidence-generating tools do we have?

Designed experiments

Formal methods

Instrumenting cognition/explainable AI

Intelligent adversarial testing

Human-machine interaction testbeds

Examples: Formal Methods

VERIFAI: A Toolkit for the Design and Analysis of Artificial Intelligence-Based Systems

Tommaso Dreossi, Daniel J. Fremont, Shromona Ghosh, Edward Kim, Hadi Ravanbakhsh, Marcell Vazquez-Chanlatte, Sanjit A. Seshia

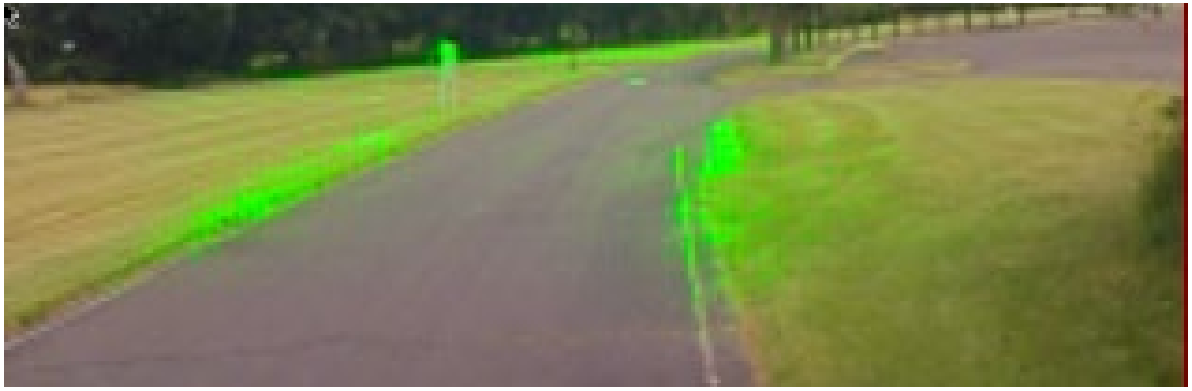
We present VERIFAI, a software toolkit for the formal design and analysis of systems that include artificial intelligence (AI) and machine learning (ML) components. VERIFAI particularly seeks to address challenges with applying formal methods to perception and ML components, including those based on neural networks, and to model and analyze system behavior in the presence of environment uncertainty.

Using Formal Verification to Evaluate Human-Automation Interaction: A Review

Bolton, Bass, and Siminiceanu

IEEE Transactions on Systems, Man, and Cybernetics: Systems 43 #3, May 2013

Example: Empirical Evidence Using XAI



Salient pixel analysis of the NVIDIA PilotNet self-steering system shows that the system all but ignores the road surface itself, focusing instead on features that indicate not -road. This system does not maintain an internal representation of the terrain; the neural net generates steering commands based on the real -time camera inputs.

Image from Bojarski et al., *Explaining How a Deep Neural Network Trained with End-to-End Learning Steers a Car.* arXiv:1704.07911v1 [cs.CV] April 25, 2017

Examples: Assurance Case applications

DARPA Assured Autonomy project

Assurance cases used for dependability of a run-time monitor for automated taxiing of aircraft

Safety case for a helicopter fly-by-wire system

https://www.researchgate.net/publication/248165577_Turning_Up_the_HEAT_on_Safety_Case_Construction

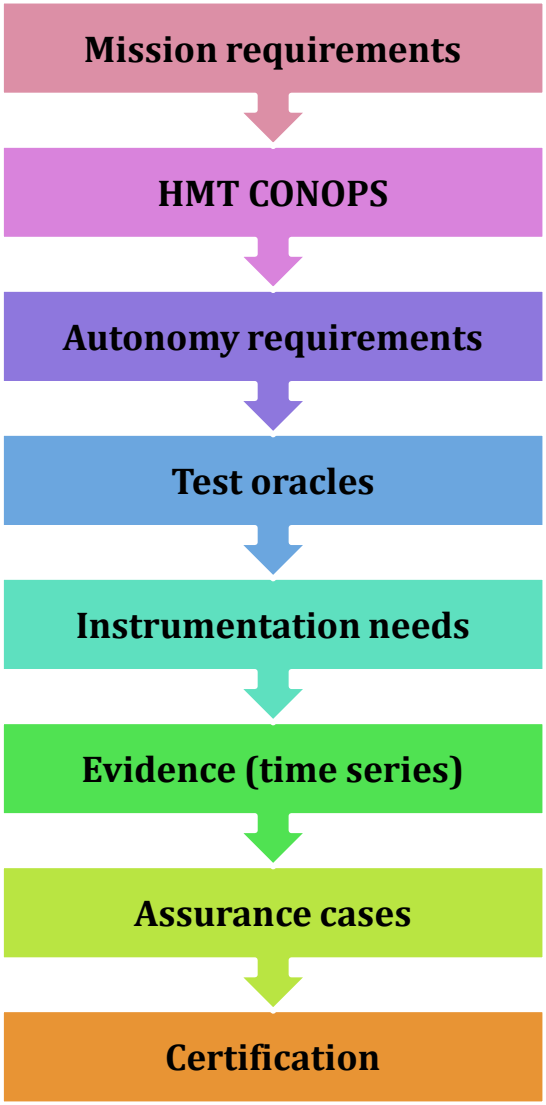
A Survey of Tool-supported Assurance Case Assessment Techniques

Maksimov, Kokaly, and Chechik

University of Toronto

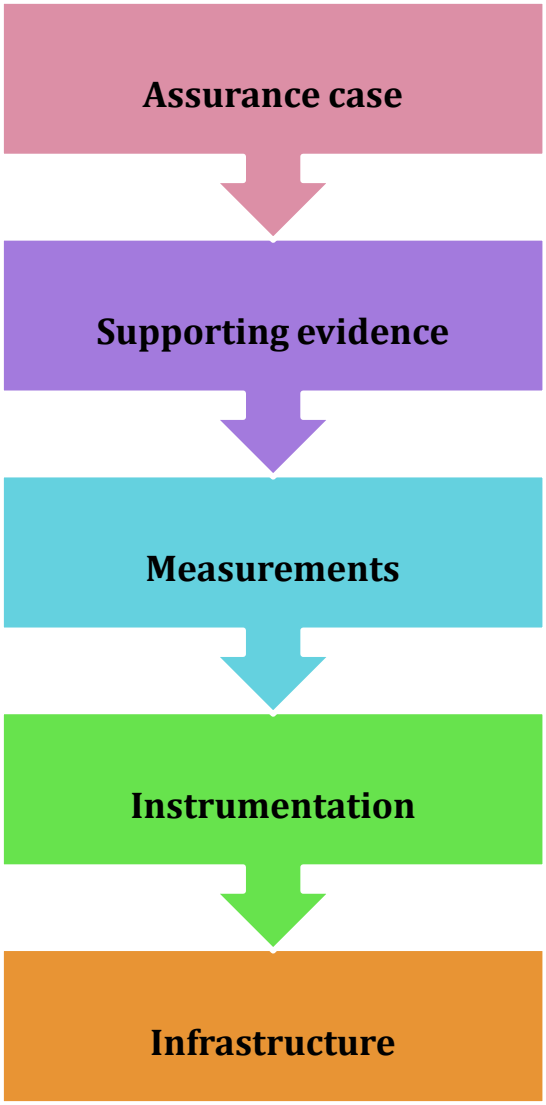
https://dl.acm.org/doi/10.1145/3342481

The assurance process is iterative



1. Make an **initial guess** at how the ADS will operate and team with humans.
2. Codify **test oracles** for acceptable behaviors, including internal behaviors and ethics.
3. Construct **assurance case outlines** – what arguments will convince? What evidence will they require?
4. Formally **verify** compliance where possible.
5. Derive **evidential requirements** – what measurements will be needed to assess performance against the oracles and provide the empirical evidence?
6. Collect evidence and **iterate**.

Work backward to identify test resource needs



1. Given the system assurance case, what evidence will be required?
2. What time series of measurements would produce that evidence?
3. What instrumentation is required to collect those measurements?
4. What infrastructure is needed to support that instrumentation?
    - Simulation testbeds?
    - Telemetry?
    - Training data?
    - Onboard recording?
    - Special mission environment?

Bottom Line at the Bottom

Assurance cases offer the best current approach to integrated test planning for ADS

Assurance cases, including “assurance surface” analysis, will strain human cognitive abilities.

Evidence to support the arguments will require a mix of formal and empirical verification techniques.

Tools exist to support the automated development and management of assurance cases and the incorporation of both formal and empirical evidence.

Backup

Both AI and Autonomy complicate assurance cases

We can't test exhaustively – the state space is too large.

We can't rely solely on DoE – we don't know the factors and can't assume smooth response everywhere.

Acceptable behavior in test events is not enough – must be *for the right reasons* to support the assurance case

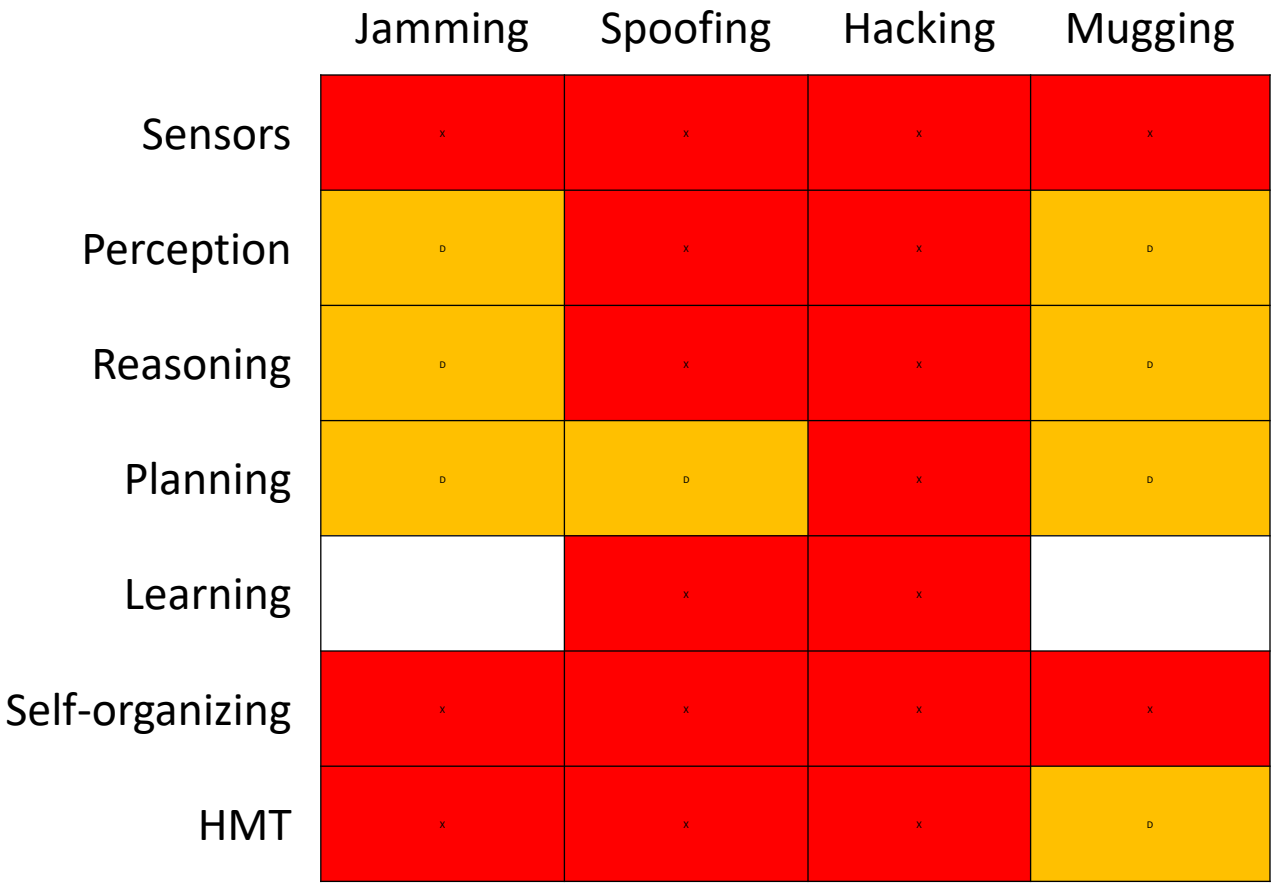
Wrappers and run-time monitors **add** complexity to T&E.

Human-Machine Teaming (HMT) explodes both the state space and the set of potentially relevant factors.

Machine Learning adds issues of training data V&V.

An informal taxonomy of mischief

1. “Jamming” – information denial (to or from)
2. “Spoofing” – input impersonation or confusion
3. “Hacking” – unauthorized access / control
4. “Mugging” – threat of physical harm



Direct vs. Indirect effects

Example: ISO/IEC 15026-2 (2011)

Systems and Software Engineering — Systems and Software Assurance — Part 2: Assurance Case

1 Scope

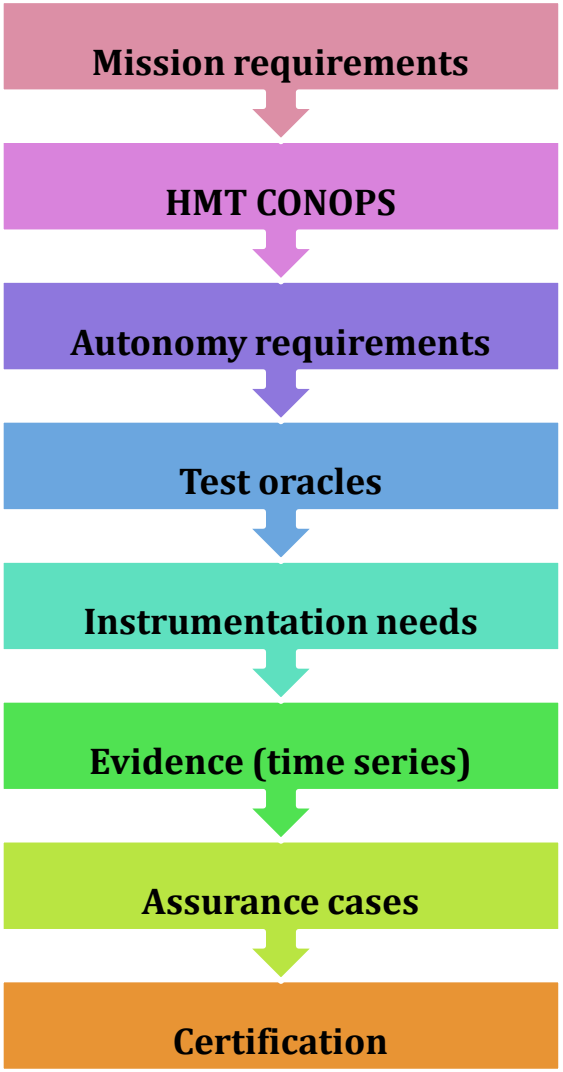
This part of ISO/IEC 15026 specifies minimum requirements for the structure and contents of an assurance case. An assurance case includes a top -level claim (or set of claims) for a property of a system or product, systematic argumentation regarding this claim, and the evidence and explicit assumptions that underlie this argumentation. Arguing through multiple levels of subordinate claims, this structured argumentation connects the top -level claim to the evidence and assumptions.

"Levels of autonomy" is a red herring

"It takes more sophisticated technology to keep the humans in the loop than it does to automate them out … On a commonly used scale of levels of autonomy, level one is fully manual control and level 10 is full autonomy … history and experience show that the most difficult, challenging and worthwhile problem is not full autonomy but the perfect five—a mix of human and machine and the optimal amount of automation to offer trusted, transparent collaboration, situated within human environments."

-- David Mindell, MIT

ADS TEV&V R&D Priorities



Specifying testable cognitive requirements

- Perception, Reasoning, …
- Teaming and self -organizing
- Negative requirements

Formal verification methods

Instrumenting cognitive functions

- Aligned with oracles
- Support both assurance and trust

Virtual test environments

V&V of ML training and training data

Intelligent adversarial testing

Test oracle specification

- Technology (in)dependent
- Human-machine teaming
- Automated generation of oracles

Logic of assurance cases

- Multi-legged arguments
- Combining formal and empirical
- Composability criteria

Regression testing criteria