INSTITUTE FOR DEFENSE ANALYSES

Formal and Empirical Verification of Autonomy for Assurance (Conference Briefing)

David M. Tate

February 2021

Approved for public release;
distribution is unlimited.

IDA Document NS D-21555

Log: H 21-000040

INSTITUTE FOR DEFENSE ANALYSES
4850 Mark Center Drive
Alexandria, Virginia 22311-1882

About this Publication

This work was conducted by the Institute for Defense Analyses (IDA) under contract HQ0034-19-D-0001, Project AX-1-3542, "Policy, Guidance, and Resource Analysis for the Director of Developmental Test and Evaluation," for the Director of Developmental Test & Evaluation (D,DT&E). The views, opinions, and findings should not be construed as representing the official position of either the Department of Defense or the sponsoring organization.

Acknowledgments

Thank you to David A. Sparrow for performing a technical review of this document.

For More Information:

David M. Tate, Lead Author
dtate@ida.org, (703) 575-1409

Steve Warner, Director, System Evaluation Division
swarner@ida.org, (703) 845-2096

Executive Summary

Assurance cases are a standard technique for establishing the trustworthiness of systems. An assurance case is a structured argument establishing that the system is sufficiently dependable to justify fielding it in a specific, defined operational context. Assurance cases require both evidence and arguments. For verification tools to be useful in helping to field autonomous systems, the tools must produce evidence that supports relevant assurance cases.

Many autonomous systems will be too complex to permit comprehensive, formal verification of all their dependability properties. This is especially true of systems that rely on machine learning (ML) to provide important perception and/or decision-making capabilities, or whose operations involve nontrivial interactions with human beings. For such systems, assurance cases will need to incorporate empirical evidence in addition to formal results to support valid arguments that provide confidence in a system's safety, dependability, and performance.

For a given autonomous system, the structure of the appropriate assurance case arguments is driven by the concept of operations and architecture of the system. This structure can be represented formally. Currently, the most common formal model of argument structure is a graph representation called Goal Structuring Notation (GSN). The evidential nodes of the GSN representation of the assurance argument provide a useful interface between the assurance case and test planning, and between the assurance case and formal verification efforts. The outputs of empirical testing and formal verification activities become the inputs to specific evidential nodes of the assurance case model. How the evidence is used in the assurance argument informs what the outputs of the formal and empirical verification activities need to be. This, in turn, informs the selection of specific formal methods and experimental designs. The specific evidence to be generated also has implications for test instrumentation, physical infrastructure, and personnel needed to execute the test design.

Formal and Empirical Verification of Autonomy for Assurance

David Tate

Institute for Defense Analyses

February 2021

The goal is assured effectiveness and dependability

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

There will always be **some** kind of certification, 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 cases are generally stovepiped.

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.

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 verification should be evidence that supports needed assurance cases.

Where does the evidence come from?

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

~~Exhaustive testing~~

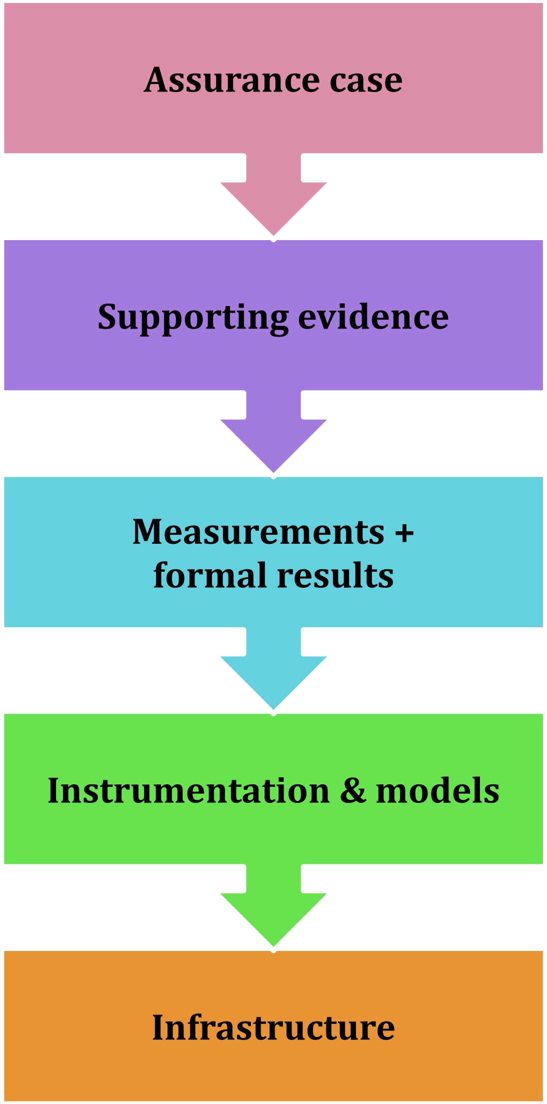
Formal verification

Design of experiments

Run-time monitors

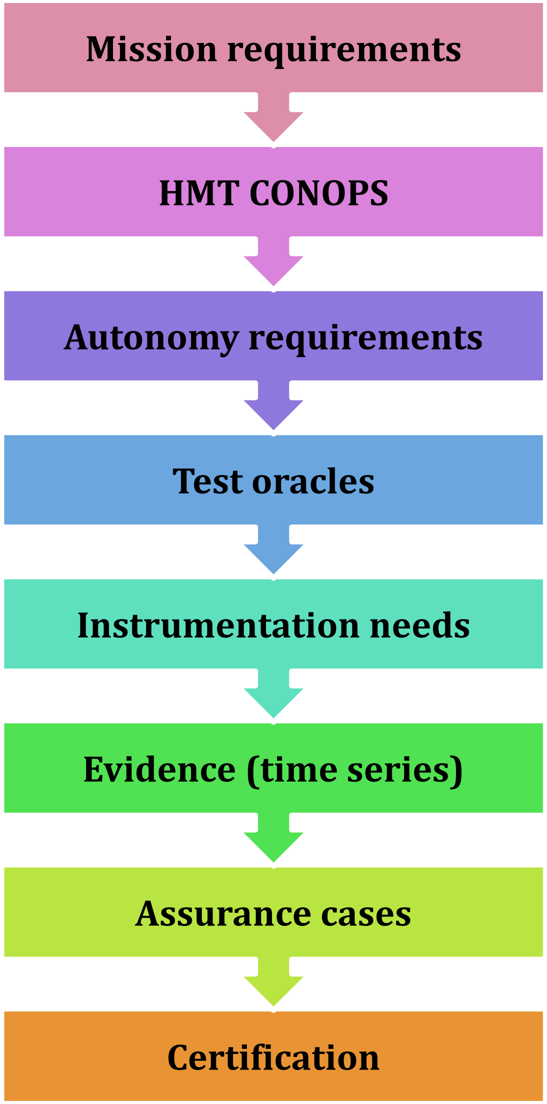
Human in the loop + training

Work backward to identify infrastructure needs



1. Given the system assurance case, what evidence will be required?
2. What time series of measurements would produce (some of) that evidence?
3. What instrumentation is required to collect those measurements?
4. What infrastructure is needed to support that instrumentation?
    - Simulation models
    - Formal models
    - Software tools
    - Human capital

Iterate forward to develop assurance cases



1. Make an **initial guess** at how the autonomous system will team with humans.
2. Codify **test oracles** for acceptable behaviors, including internal behaviors.
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.**

What new (and existing) tools might be useful?

Formal methods

Instrumenting cognition/explainable AI

Intelligent adversarial testing

Assurance case development tools

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

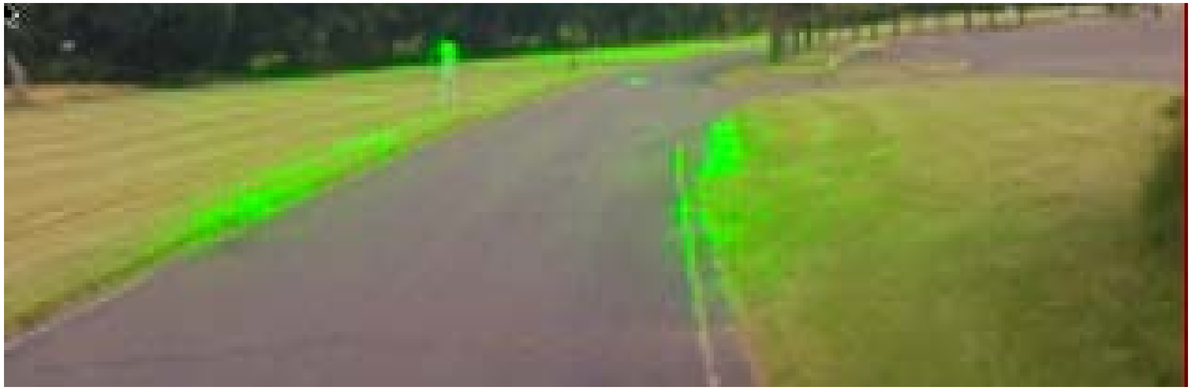
We present VERIFAI, a software toolkit for the formal design and analysis of systems that include artificial intelligence (AI) and machine learning (ML) components. VERIFAI particularly seeks to address challenges with applying formal methods to perception and ML components, including those based on 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 Development Tools

Assurance Case Editor (**ACedit)**

Assurance Case Automation Toolset (**AdvoCATE**)

Evidence Confidence Assessor (**EviCA**)

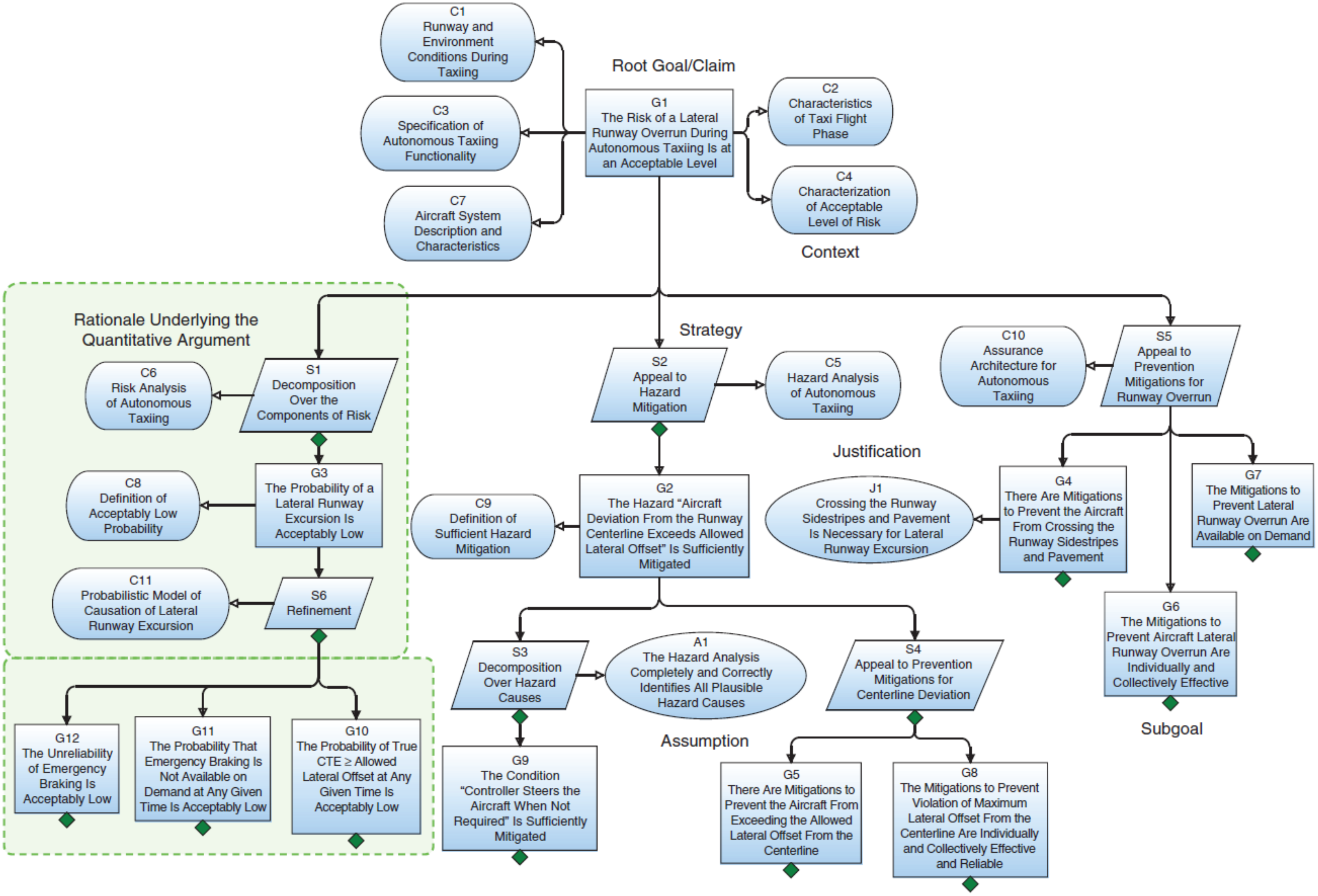
Astah GSN (commercial product, see astah.net)

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



Root Goal/Claim

G1 The Risk of a Lateral Runway Overrun During Autonomous Taxiing Is at an Acceptable Level

C1 Runway and Environment Conditions During Taxiing

C3 Specification of Autonomous Taxiing Functionality

C7 Aircraft System Description and Characteristics

C2 Characteristics of Taxi Flight Phase

C4 Characterization of Acceptable Level of Risk

Context

Rationale Underlying the Quantitative Argument

S1 Decomposition Over the Components of Risk

C6 Risk Analysis of Autonomous Taxiing

G3 The Probability of a Lateral Runway Excursion Is Acceptably Low

C8 Definition of Acceptably Low Probability

S6 Refinement

C11 Probabilistic Model of Causation of Lateral Runway Excursion

G12 The Unreliability of Emergency Braking Is Acceptably Low

G11 The Probability That Emergency Braking Is Not Available on Demand at Any Given Time Is Acceptably Low

G10 The Probability of True CTE ≥ Allowed Lateral Offset at Any Given Time Is Acceptably Low

Strategy

S2 Appeal to Hazard Mitigation

C5 Hazard Analysis of Autonomous Taxiing

G2 The Hazard “Aircraft Deviation From the Runway Centerline Exceeds Allowed Lateral Offset” Is Sufficiently Mitigated

C9 Definition of Sufficient Hazard Mitigation

S3 Decomposition Over Hazard Causes

A1 The Hazard Analysis Completely and Correctly Identifies All Plausible Hazard Causes

Assumption

G9 The Condition “Controller Steers the Aircraft When Not Required” Is Sufficiently Mitigated

S4 Appeal to Prevention Mitigations for Centerline Deviation

G5 There Are Mitigations to Prevent the Aircraft From Exceeding the Allowed Lateral Offset From the Centerline

G8 The Mitigations to Prevent Violation of Maximum Lateral Offset From the Centerline Are Individually and Collectively Effective and Reliable

Justification

J1 Crossing the Runway Sidestripes and Pavement Is Necessary for Lateral Runway Excursion

C10 Assurance Architecture for Autonomous Taxiing

S5 Appeal to Prevention Mitigations for Runway Overrun

G4 There Are Mitigations to Prevent the Aircraft From Crossing the Runway Sidestripes and Pavement

G7 The Mitigations to Prevent Lateral Runway Overrun Are Available on Demand

G6 The Mitigations to Prevent Aircraft Lateral Runway Overrun Are Individually and Collectively Effective

Subgoal

Bottom Line at the Bottom

Assurance cases for autonomous systems 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.

IDA

Backup

Case Study: Assurance Case Development

A Case Study in Assurance Case Development
for Scientific Software
Mojdeh Sayari Nejad
MS (Computer Science) Thesis
McMaster University, 2017

Assurance case for 3dfim+ software for analyzing functional MRI images of the brain

https://macsphere.mcmaster.ca/handle/11375/23075

“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 1 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

Autonomy TEV&V R&D Priorities

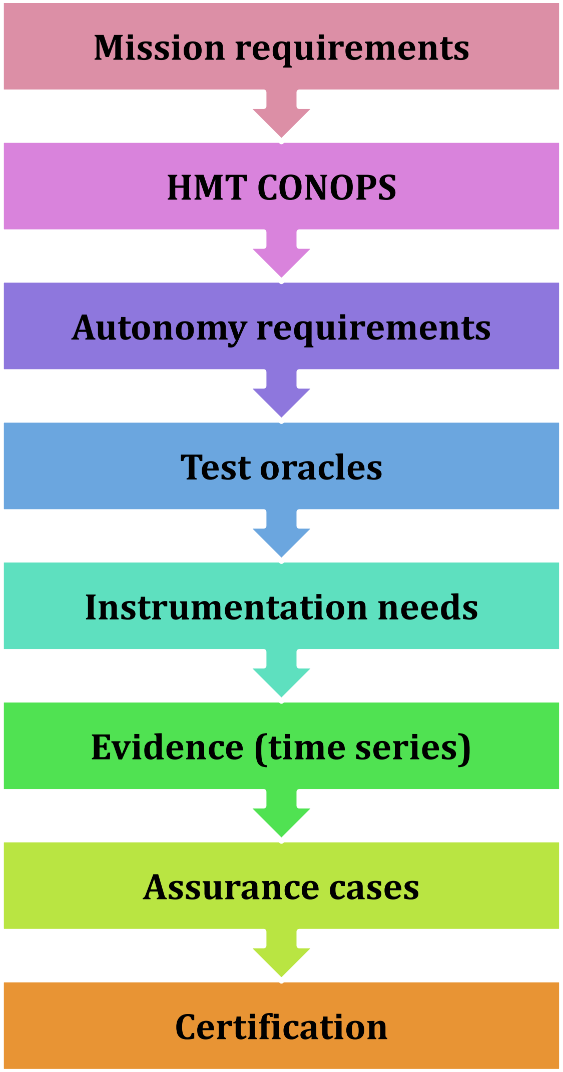
Test oracle specification

- 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



Mission requirements

HMT CONOPS

Autonomy requirements

Test oracles

Instrumentation needs

Evidence (time series)

Assurance cases

Certification

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

Examples: Intelligent Adversarial Testing

The **Range Adversarial Planning Tool** (RAPT) developed at Johns Hopkins Applied Physics Laboratory automates adversarial testing of autonomous systems using simulations of the autonomy software and environment. RAPT builds a machine-learning model of the autonomy performance and then identifies regions of the configuration space with steep response gradients, indicating possible edge cases. RAPT then generates test designs that oversample the identified regions.

Similarly, the IBM **Adversarial Robustness Toolbox** (ART) supports the verification of robustness and hardening for machine learning models.

REPORT DOCUMENTATION PAGE	*Form Approved* *OMB No. 0704-0188*

1. REPORT DATE *(DD-MM-YYYY)* xx-02-2021	**2. REPORT TYPE** Final	**3. DATES COVERED** *(From - To)* February 2021

4. TITLE AND SUBTITLE Formal and Empirical Verification of Autonomy for Assurance (Conference Presentation)	**5a. CONTRACT NUMBER** HQ0034-19-D-0001
	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S) Tate, David, M.	**5d. PROJECT NUMBER** AX-1-3542
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Institute for Defense Analyses 4850 Mark Center Drive Alexandria, VA 22311-1882	**8. PERFORMING ORGANIZATION REPORT NUMBER** IDA Document NS D-21555 H 21-000040
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) Director, Developmental Test, Evaluation and Assessments Engineering Policy and Systems Office of the Under Secretary of Defense Research and Engineering Washington, DC 20301	**10. SPONSOR/MONITOR'S ACRONYM(S)** D(DTE&A)
	11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT

Approved for public release; distribution is unlimited.

13. SUPPLEMENTARY NOTES

The materials in this briefing are from approved public release sources. IDA Document NS D-14269, "T&E of Cognitive EW: An Assurance Case Framework (Conference Presentation)"; IDA Document NS D-10523, "Attack Surfaces of Autonomy"; and IDA Document NS D-10922, "What Counts as Progress in the T&E of Autonomy?".

14. ABSTRACT

Many autonomous systems will be too complex to permit comprehensive formal verification of their properties. For such systems, assurance cases must combine both formal and empirical evidence to form valid arguments supporting confidence in system safety, dependability, and performance. This talk presents a conceptual framework for constructing assurance cases for autonomy, and describes a systematic procedure for identifying evidential requirements. It also discusses implications for instrumentation and test infrastructure, and existing software tools to support management of assurance cases and incorporation of formal results.

15. SUBJECT TERMS

Autonomy; formal methods; test and evaluation; assurance case; test oracle; instrumentation; modeling and simulation

16. SECURITY CLASSIFICATION OF:			**17. LIMITATION OF ABSTRACT**	**18. NUMBER OF PAGES**	**19a. NAME OF RESPONSIBLE PERSON**
a. REPORT	**b. ABSTRACT**	**c. THIS PAGE**			Tate, David, M.
Unclassified	Unclassified	Unclassified	Same as Report	24	**19b. TELEPHONE NUMBER** *(Include area code)* (703) 575-1409

Reset