Developmental Test and Evaluation of Autonomous Systems Guidebook

September 2025

Office of the Director,
Developmental Test, Evaluation, and Assessments

Office of the Under Secretary of Defense
for Research and Engineering

Washington, D.C.

Developmental Test and Evaluation of Autonomous Systems Guidebook

Office of the Director, Developmental Test, Evaluation, and Assessments
Office of the Under Secretary of Defense for Research and Engineering
3030 Defense Pentagon
Washington, DC 20301-3030
osd.r-e.comm@mail.mil
https://www.cto.mil/dtea/

Executive Summary

The Department of Defense (DoD) developed this guidebook to support the test and evaluation (T&E) of autonomous systems. This guidebook addresses the novel challenges of removing or greatly reducing involvement of human operators from DoD systems and empowering future autonomous systems to independently act in operational environments. These challenges demand iterative approaches for evaluating the growing capabilities of autonomous systems to ensure trusted mission capability across complex operational environments. Therefore, this guidebook intends to:

- Identify and explain DoD policy for autonomous systems T&E.
- Identify and explain overarching and specific challenges for autonomy T&E.
- Share guidance on methods and best practices for the full continuum of autonomy T&E.
- Provide information about tools and resources for autonomy T&E from trusted Federal sources.

The information detailed in this guidebook focuses primarily on issues faced by independent government test teams for planning and executing autonomous systems T&E, while also providing insights to stakeholders who support or rely upon T&E processes. Recognizing that autonomy is an emerging technology, this guidebook intends to provide the best information available on current issues and will be updated as technology and methodologies evolve.

A key challenge of testing autonomous systems is that a human operator is absent from continuous control, requiring the autonomous system to perform dynamic observe-orient-decide-act (OODA) loop operations through a diverse range of environmental and mission conditions and scenarios. This overarching challenge produces many complicating difficulties such as the following:

- Requirements and the concept of operations intended for autonomous system behaviors are often too broad or too narrow, incomplete, inconsistent, subjective, untestable, or poorly defined.
- The safety of autonomous systems that take physical actions, independently and without human control, shifts safety responsibilities and risks from the user to the designer, developer, and tester.
- Data problems abound, such as realism, availability, analysis, security, and adequacy.
- Black box software or artificial intelligence components lead to unknown performance in untested scenarios.

- Human-autonomy teaming models and measures for DoD are not yet comprehensive or mature.

To address these and many other emerging challenges, the guidebook includes the following methods and best practices based on lessons learned for the full continuum of autonomy T&E:

- End-to-end autonomy T&E processes, based on mission and system decomposition and iterative testing, for evidence aggregation and ongoing validation of autonomous system trustworthiness.
- Acquisition and test strategy practices such as open system architecture; assurance case arguments; extensive use of modeling and simulation; and live, virtual, and constructive testing.
- Test planning and execution methods including scientific test and analysis techniques, runtime assurance, continuous testing, adversarial testing, and cognitive instrumentation.
- Data analysis supporting model validation, quantified risks, and task-based certifications.

The guidebook leverages emerging best practices in agile and iterative testing to extend success throughout the T&E continuum. By applying these best practices to achieve efficient, effective, and robust developmental T&E, autonomous DoD systems will be primed for successful operational T&E and operational employment.

FLORES.ORLANDO.F.1232194045

Digitally signed by FLORES.ORLANDO.F.1232194045
Date: 2025.09.30 15:43:38 -04'00'

Mr. Orlando F. Flores
Acting Director, Developmental Test, Evaluation,
and Assessments

Contents

Figures

Tables

1 Introduction

1.1 Purpose of This Guidebook

This guidebook provides focused guidance and recommended practices for early and developmental test and evaluation (T&E) of autonomous systems for the purposes of the Department of Defense (DoD), primarily for independent government developmental test and evaluation (DT&E), while informing industry T&E as well. This guidebook addresses the novel challenges of removing or greatly reducing the involvement of human operators from DoD systems and empowering future autonomous systems, especially those that are artificial intelligence (AI)-enabled, to independently act in operational environments. These challenges demand iterative approaches for evaluating the growing capabilities of autonomous systems to ensure trusted mission capability across complex operational environments. Therefore, this guidebook intends to:

- Identify and explain DoD policy for autonomous systems T&E.
- Identify and explain overarching and specific challenges for autonomy T&E.
- Share guidance on methods and best practices for the full continuum of autonomy T&E.
- Provide information about tools and resources for autonomy T&E from trusted Federal sources.

The guidebook leverages emerging best practices in agile and iterative testing to extend success throughout the T&E continuum. By applying these best practices to achieve efficient, effective, and robust DT&E, autonomous DoD systems will be primed for successful operational T&E and operational employment.

1.1.1 Scope

To provide the best value to the intended audiences while limiting the length of this document, the authors have defined this guidebook's scope as follows:

Included in This Guidebook

- T&E policy, background, vision, challenges, methods, best practices, resources, tools, glossary, and references relevant to *autonomous systems* that:
  - Are developed or purposed primarily for military use.
  - Are integrated systems, not limited to just hardware or software.

  - Utilize AI, both from machine learning (ML) and from complex programming rules, to make system decisions and dictate behavior.
  - Include aspects of AI T&E to understand how it drives requirements and informs and synergizes with integrated autonomous systems T&E.
  - Act independently, physically in the real world in some way, and in any domain such as ground, water, air, or space.
  - Interface, interact, and team with other systems and with human teammates.
  - Use and generate data—AI training data, test data, modeling and simulation (M&S) data, operational data and more.
  - Are employed in both peacetime scenarios and high-intensity combat operations.
  - Are at any point across the system life cycle, from M&S to early technology development through contractor testing (CT), developmental testing (DT), operational testing (OT), and post-acceptance testing during operations and sustainment.
- References and links to other documents that can expand upon this information.

Not Included in This Guidebook

- Organizational roles and responsibilities; this guidebook is not a directive.
- Automatic systems without complex rules.
- Automation of tasks for human-operated systems.
- T&E of AI that performs other roles, external to autonomous systems.
- Business systems and human decision aids that do not physically act.
- Commercial systems not intended for military use.

Scope Summary

In summary, if the information is important to the testing or evaluation of military, integrated autonomous systems, or if it is a product of autonomous systems T&E, it is intended for inclusion in this guidebook.

1.1.2 Key Definitions

Autonomy and AI are sometimes used interchangeably; however, they are different concepts, and the distinction is important. A useful starting point to compare AI and autonomy with regard to T&E is the AI Acquisition Guidebook, which states:

The following distinction between autonomy and AI should be recognized – autonomy refers to an agent or machine being delegated to perform a task, while AI is a means to achieve that goal.

This statement makes two important distinctions. First, autonomous systems will need to be evaluated in a system and mission context, emphasizing measures of effectiveness. The AI contribution is evaluated primarily in terms of measures of performance. Second, the T&E of AI techniques or models under development will include broad characterization of the performance measures, without necessarily having an application in mind. This characterization of performance is an important stage in making AI components understood and available for inclusion in systems—and an important element in AI adoption. As an AI-enabled component is joined to a platform, the evaluation needs to include a full system and mission context along with human elements.

- **Autonomy** refers to an agent or machine being delegated to perform a task—capable of independent operation without external control. Autonomous systems in DoD will need to be evaluated in a system and mission context, emphasizing measures of effectiveness to characterize the system's integrated capabilities and limitations including software, hardware, and the synergies or disconnects between them.
- **AI** refers to the ability of machines to perform tasks that normally require human intelligence; AI may be an element in a system pursuing a goal. AI and ML can be used at different stages of autonomy control and operation to aid in determining the best and most efficient solutions for the tasks to which the system has been assigned.
- **ML** refers to the ability of machines to learn from data without being explicitly programmed. ML is a *subset of AI* techniques. AI that does not utilize ML goes by different names, such as expert AI, rule-based AI, symbolic AI, domain ontology and reasoning, and multi-agent planners.

Note: For a discussion of the specific challenges and guidance for the T&E of AI, see the DT&E of AI-Enabled Systems Guidebook.

1.1.3 Audience

This guidebook is intended for T&E practitioners, including program managers, test planners, test engineers, and analysts. It is primarily intended for a government DoD audience, though most guidance may be very useful to industry practitioners as well. The guidebook should support the development of test strategies with applicable methodologies and the tools to improve rigor in addressing the challenges unique to the T&E of autonomous systems. It should also be useful for researchers into advanced T&E methods and tools to inform them about the

most urgent gaps in current capabilities and where innovations are needed. Finally, this guidebook should be informative for all professionals relying on T&E or supporting T&E, such as requirements managers, program executives, contracting officers, systems engineers, M&S managers, system developers, human-system interface engineers, military commanders, and operators, by illuminating how to maximize the benefits of T&E for autonomous systems as well as how to facilitate effective, robust, and efficient T&E best practices. The guidebook is intended to be a living document with contributions by the entire DoD community and will adapt to ensure that it is getting the right information to the right audience.

1.1.4 Benefits

The guidebook is intended to generate the following benefits across the DoD autonomous systems community:

- Establishing effective, efficient, and robust autonomous military systems T&E enabling future mission capabilities.
- Informing program managers on program and contract decision T&E risks and opportunities.
- Informing autonomy systems engineering on model, simulation, and system design T&E risks and opportunities.
- Informing autonomy requirements, the concept of operations (CONOPS), human systems integration (HSI), and mission capabilities based on T&E risks and opportunities.

1.2 Evolution of Guidance from Emerging Information

One of the many challenges in developing this guidebook is that autonomous systems employ new technologies; and new challenges, information, methods, practices, tools, and resources are emerging at a rapid pace. Recognizing this rapidly changing technology environment, the authors of this guidebook specifically note the following:

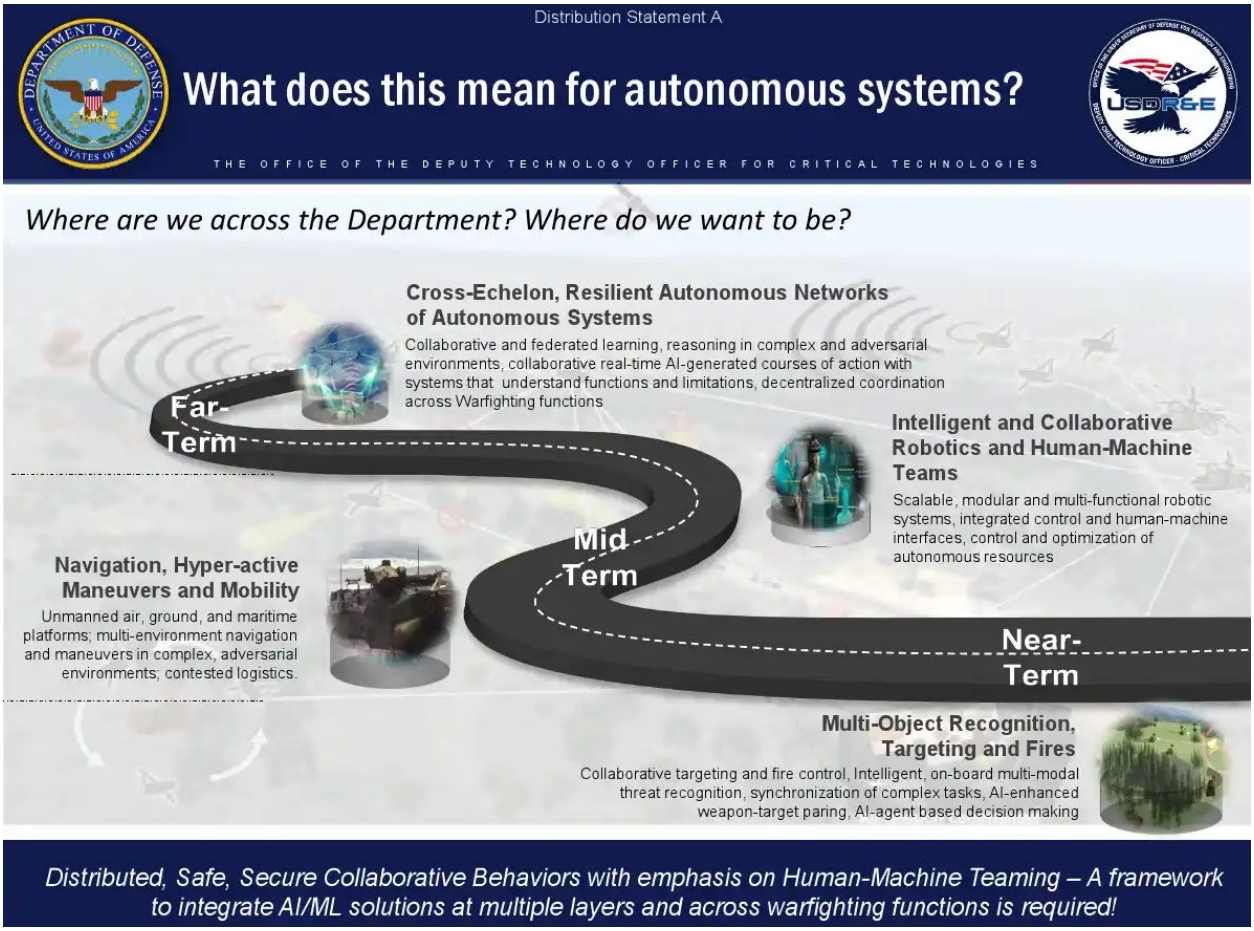
- This guidebook provides a snapshot of the best available information at the time of writing and publication. Some challenges, methods, practices, tools, or resources described herein may become outdated in the future.
- New challenges, methods, practices, tools, and resources not described in this guidebook may have a significant impact on autonomous systems T&E and may become available in the future.

Given this dynamic nature of autonomous systems emerging technologies, the authors and sponsors of this guidebook intend to update and expand it on a relatively frequent basis. Every

effort will be made to continue to provide the most useful information available at the time as future guidebook revisions occur. Every effort will also be made to ensure that revisions are distributed quickly to those audiences who benefit from this guidance.

1.2.1 Future of Autonomy Test and Evaluation

The future of autonomous systems within DoD is uncertain. The promise of many benefits from increased employment of autonomous systems is currently driving great interest and investment into advancing and developing future autonomous capabilities. The Trusted AI and Autonomy Roadmap includes a future DoD autonomous system vision, as depicted in Figure 1-1.



Source: Trusted AI and Autonomy Roadmap 2024

Figure 1-1. Future Autonomous Systems Roadmap Vision

As DoD develops more advanced and capable autonomous systems, so too the T&E community must develop and implement more advanced and capable T&E processes, methods, and infrastructure to effectively, efficiently, and robustly perform T&E of these emerging capabilities. Independent government T&E of autonomous systems must provide complete, timely information and analysis to facilitate the employment of trustworthy, capable future autonomous systems. This T&E information must accurately characterize the systems' risks and benefits to support the DoD mission to defend and protect the United States and its allies.

2 Policy

The policies discussed in this section are U.S. Federal policies or DoD policies that provide important goals, procedures, and other direction that are highly relevant to autonomous systems and their T&E.

2.1 Federal Policies

2.1.1 Executive Order on Artificial Intelligence

Executive Order 14179, "Removing Barriers to American Leadership in Artificial Intelligence," is relevant to autonomous systems T&E because it provides policy for Federal goals for AI, which may include how AI supports autonomous systems.

Overview. This Executive order (which rescinds Executive Order 14110, "Safe, Secure, and Trustworthy Development and Use of Artificial Intelligence") states, "It is the policy of the United States to sustain and enhance America's global AI dominance in order to promote human flourishing, economic competitiveness, and national security" for the purpose of developing "AI systems that are free from ideological bias or engineered social agendas."

Policy Implications for Autonomous Systems T&E. The policy intends to prioritize U.S. dominance in AI and may have future interpretations that affect the sharing of AI tools and resources with foreign allies or organizations. It may also lead to increased focus on bias in AI that could impact the T&E requirements for AI-enabled systems, which may include autonomous systems.

Summary for DoD DT&E. Executive Order 14179 sets a goal for U.S. dominance in AI that informs potential prioritization and use of AI tools and resources.

2.1.2 Responsible Military Use of Artificial Intelligence and Autonomy

The 2023 Department of State Declaration, "Political Declaration on Responsible Military Use of Artificial Intelligence and Autonomy," is relevant to autonomous systems T&E because it provides internationally coordinated guidance specific to the military use of autonomous systems.

Overview. This policy provides a normative framework addressing the use of AI and autonomy capabilities in the military domain. To date, 58 nations have endorsed the Declaration, including the United States, United Kingdom, France, Japan, Australia, and most Western and democratic nations. Notably, nations including Russia, China, Iran, North Korea, India, Pakistan, and most countries in Asia, Africa, and South America have not endorsed it.

Policy Implications for Autonomous Systems T&E. The policy states that military use of AI and autonomy should be ethical and responsible; must comply with international law; and should be accountable within a responsible chain of command and control. Furthermore, their use should carefully consider risks and benefits, which requires T&E to effectively characterize the system across its operational environment, and should minimize unintended bias and accidents, which requires robust T&E of the most dangerous usages in addition to the most common ones.

The policy provides that:

- These capabilities should be developed with methodologies, data sources, design procedures, and documentation that are transparent and auditable.
- These capabilities should have explicit, well-defined uses and be designed and engineered to fulfill those intended functions.
- Personnel who use or approve the use of these capabilities should be trained so they sufficiently understand the capabilities and limitations of those systems in order to make appropriate context-informed judgments on the use of those systems and to mitigate risks.

The policy directs that states should ensure that the safety, security, and effectiveness of military AI capabilities are subject to appropriate and rigorous testing and assurance within their well-defined uses and across their entire life cycles. For self-learning or continuously updating military capabilities, leaders should ensure that critical safety features have not been degraded, through processes such as monitoring. Finally, leaders should implement appropriate safeguards to mitigate risks of failures, such as the ability to detect and avoid unintended consequences and the ability to respond, for example by disengaging or deactivating deployed systems, when such systems demonstrate unintended behavior.

Summary for DoD DT&E. This Declaration sets a foundation of expectations for the T&E of AI-enabled and autonomous systems that have military use. The need for ensuring the understanding of system capabilities and risks and characterizing system safety, security, and effectiveness across usage contexts drives the demand for rigorous testing across the life cycle.

2.2 DoD Policies

2.2.1 DoD Instruction 5000.89

DoD Instruction (DoDI) 5000.89, “Test and Evaluation,” provides a baseline of T&E requirements and best practices for the acquisition of all DoD T&E systems, which aids in scoping autonomous systems T&E.

Overview. This policy directs that the DoD Components will conduct developmental, operational, and live fire T&E as part of an adequate T&E program and will integrate test planning and test execution across stakeholders to facilitate an efficient use of data and resources. DoDI 5000.89 also defines specific roles and responsibilities for the organizations and personnel that manage, execute, and support DoD T&E.

Policy Implications for Autonomous Systems T&E. DoDI 5000.89 states that DT&E activities will start when requirements are being developed to ensure that key technical requirements are measurable, testable, and achievable; as well as provide feedback that the systems engineering process is performing adequately. In practice, the significant inclusion of DT&E into requirements development and systems engineering has often been minimal for traditional DoD systems. For autonomous systems, these early program activities become much more essential for the inclusion of DT&E expertise because the capabilities to robustly and effectively test autonomous systems must be integrated into the design of the system. For more information on this concept, see Section 4 of this guidebook.

DoDI 5000.89 mandates that T&E provides engineers and decision-makers with knowledge to assist in managing risks; to measure technical progress; and to characterize operational effectiveness, operational suitability, interoperability, survivability (including cybersecurity), and lethality. These objectives are met by planning and executing a robust and rigorous T&E program. For autonomous systems, these T&E needs help translate into a baseline of trust through the establishment of the system's trustworthiness. See Section 3 of this guidebook for more information.

Summary for DoD DT&E. This policy sets the foundation of expectations for the T&E of all DoD systems. The guidance set forth in DoDI 5000.89 provides a baseline for understanding many of the challenges and best practices for autonomous systems T&E.

2.2.2 DoD Directive 3000.09

DoD Directive (DoDD) 3000.09, "Autonomy in Weapon Systems," provides guidance and requirements for the design, development, acquisition, testing, fielding, and employment of autonomous and semi-autonomous weapon systems, including guided munitions that are capable of automated target selection, that apply lethal or non-lethal, kinetic or non-kinetic, force.

Overview. DoDD 3000.09 defines an autonomous weapon system as "a weapon system that, once activated, can select and engage targets without further intervention by an operator" and a semi-autonomous weapon system as "a weapon system that, once activated, is intended to only engage individual targets or specific target groups that have been selected by an operator."

DoDD 3000.09 provides short examples and clarification about the applicability of the system to the directive based on the definition.

Before a system enters formal development, the directive requires a review and approval of the system by the Under Secretary of Defense for Policy (USD(P)), the Under Secretary of Defense for Research and Engineering (USD(R&E)), and the Vice Chairman of the Joint Chiefs of Staff (VCJCS). Another review and approval by the USD(P), the Under Secretary of Defense for Acquisition and Sustainment, and the VCJCS is required before fielding. Both reviews are coordinated with and supported by the Autonomous Weapon Systems Working Group to ensure system applicability, completeness, and readiness.

Policy Implications for Autonomous Systems T&E. DoDD 3000.09 provides guidelines designed to minimize the probability and consequences of failures in autonomous and semi-autonomous weapon systems that could lead to unintended engagements. This goal is achieved through demonstration to the review panels of a rigorous systems engineering process that will include hardware and software verification and validation (V&V) and realistic system developmental and operational T&E.

Although this guidebook is a DT document, information related to OT is included for awareness and to facilitate agile and integrated test approaches. The policy outlines specific requirements, as excerpted in Figure 2-1, that pertain to T&E and V&V that should be considered during the development and testing of any system that would fall under the purview of DoDD 3000.09.

Although the rigorous systems engineering process should exist for any systems developed, the additional review and approval helps to ensure the coverage and completeness of the development process.

a. Systems will go through rigorous hardware and software V&V and realistic system developmental and operational T&E, including analysis of unanticipated emergent behavior.

(1) Hardware and software V&V will include iterative cyber T&E in accordance with DoDI 5000.89, to verify that the weapon system is resilient and survivable in contested cyberspace.

(2) Systems incorporating AI capabilities will go through rigorous developmental and operational T&E to verify and validate that the AI is robust according to design requirements.

b. T&E of systems incorporating AI capabilities will include testing to confirm that their autonomy algorithms can be rapidly reprogrammed on new input data.

c. After initial operational T&E, as directed by the Director, Operational Test and Evaluation (DOT&E), system data will be collected and any further changes to the system will undergo appropriate V&V and T&E to ensure that critical safety features have not been degraded.

(1) System software will be tested using best-available DoD means and methods to validate that critical safety features have not been degraded. Automated testing tools, such as modeling and simulation, will be used whenever feasible. The testing will identify any new operating states and other relevant changes in the autonomous or semi-autonomous weapon system.

(2) As directed by the DOT&E:

(a) Each new or revised operating state will undergo appropriate and tailored additional T&E to characterize the system behavior in that new operating state.

(b) Changes to the state transition matrix may require whole system follow-on operational T&E.

d. In coordination with the USD(R&E) and DOT&E, the owning Component will provide for monitoring to identify and address when changes to the system design or operational environment require additional T&E to provide sufficient confidence that the system will continue to avoid unintended engagements and resist interference by unauthorized parties.

Figure 2-1. DoD Directive 3000.09 Autonomy T&E Policy Excerpt

Summary for DoD DT&E. DoDD 3000.09 sets the expectations for the T&E of all DoD autonomous weapon systems. The guidance set forth demands rigorous, realistic autonomous weapon systems T&E, and although it does not explicitly apply to non-weapon autonomy, its

principles should be applied as a best practice to all autonomous systems, not just weapon systems.

2.2.3 DoD Data, Analytics, and Artificial Intelligence Adoption Strategy

The 2023 DoD Data, Analytics, and AI Adoption Strategy is relevant to autonomous systems T&E because it guides DoD leaders and warfighters on how to make rapid, well-informed decisions by expertly leveraging high-quality data, advanced analytics, and AI as part of a continuous, outcome-driven, and user-focused development, deployment, and feedback cycle applicable to autonomous systems.

Overview. This policy calls for an *agile* approach to technology development and deployment that ensures a tight feedback loop between technology developers and users through a continuous cycle of iteration, innovation, and improvement of solutions that enable decision advantage. Creating effective, *iterative* feedback loops among developers, users, subject matter experts, and T&E experts will ensure that capabilities are more stable, secure, ethical, and trustworthy.

Policy Implications for Autonomous Systems T&E. The policy states that sound assurance processes for testing, evaluation, validation, and verification are imperative for providing increased data quality and insightful analytics that are needed for improved, faster, and ethical mission outcomes with responsible AI. The policy advocates open standard architectures, improved data management and cybersecurity, and the design and testing of AI-enabled solutions via robust campaigns of learning to account for different operational environments. The policy's top dimension of data quality is data accuracy and poses these questions: How frequently do data values match ground truth? How is error measured, and is error tolerable for the specified purpose? These questions beget robust, effective, and iterative T&E to underpin data and generate useful, actionable insights.

The strategy approach embraces the need for speed, agility, learning, and responsibility—these goals induce several of the prominent challenges for the T&E of autonomous systems, discussed in Section 4 of this guidebook. The most noticeable challenge that this strategy generates is the need to change the paradigms of DoD T&E from traditional "waterfall" sequential series of test programs—with early science and technology (S&T), vendor testing, DT, OT, and follow-on post-acceptance testing all discrete, sequenced stages—into new paradigms where T&E is a continuum of integrated, synergistic, and overlapping S&T, CT, DT, OT, and beyond to effect agile, iterative capability development and delivery to the warfighters.

Summary for DoD DT&E. This policy sets a strategy for agile, iterative development and deployment of AI-enabled and data-driven systems including autonomous systems. The need for speed and agility in learning generates demands for agile, iterative T&E as a continuum

throughout the life cycle, extending both "left"—or earlier in development—and "right"—or later into post-acceptance and post-fielding product improvement.

2.2.4 DoD Instruction 5000.61

DoDI 5000.61, "DoD Modeling and Simulation Verification, Validation, and Accreditation," provides a baseline of M&S requirements for all DoD systems, which aids in scoping autonomous systems T&E.

Overview. DoDI 5000.61 establishes policy, assigns roles and responsibilities, and prescribes procedures for the verification, validation, and accreditation (VV&A) of models, simulations, distributed simulations, and associated data. DoDI 5000.61 applies to all models, simulations and associated data developed, used, made available, or managed by the DoD Components, including those used by non-DoD organizations to support DoD processes, products, or procedures.

Policy Implications for Autonomous Systems T&E. DoDI 5000.61 states that all models, simulations, and associated data used to support DoD processes, products, and decisions must undergo V&V throughout their life cycles and must be accredited for a specific intended use. DoDI 5000.61 also provides documentation requirements for M&S VV&A and introduces Military Standard MIL-STD-3022, which provides standardized templates to help enable the efficient reuse of M&S data and tools.

The benefits of extensive use of M&S for very complex systems such as autonomous systems are enormous, as discussed in Section 5.1.3 of this guidebook. The effective, robust application of T&E best practices to the strategy, planning, execution, and analysis of autonomous systems M&S allows efficient V&V of the M&S results and aids actual system hardware and software T&E by exploring and understanding the system's simulated performance throughout the operational environment and scenario contexts. Conversely, robust autonomous systems T&E supports efficient M&S through producing justified confidence that the M&S provides credible insights.

Summary for DoD DT&E. This policy mandates the VV&A of all models, simulations, distributed simulations, and associated data used to support DoD processes, products, and decisions. The guidance set forth in DoDI 5000.61 provides a foundation for efficiently and effectively integrating autonomous systems T&E and M&S into a coherent evaluation strategy.

2.2.5 DoD Instruction 5000.90

DoDI 5000.90, "Cybersecurity for Acquisition Decision Authorities and Program Managers," provides a baseline of cybersecurity activities for all DoD systems, which aids in understanding autonomous systems cybersecurity needs for T&E.

Overview. DoDI 5000.90 directs that program managers will assess, mitigate, and monitor cybersecurity risks to the program information and the information system and to the platform information technology. It mandates the identification of risks and consequences of a cybersecurity breach, including situations where a cybersecurity breach or failure would jeopardize military technological advantage or mission-critical functionality, including the cybersecurity of enabling networks, supporting systems, and supply chains.

Policy Implications for Autonomous Systems T&E. DoDI 5000.90 identifies the many risks and management needs for providing cybersecurity to DoD systems. Given that autonomous systems will utilize integrated software to analyze information and make decisions that affect real-world actions, the importance of successful cybersecurity takes on immensely greater importance for these autonomous systems. Autonomous systems will have expanded cybersecurity vulnerabilities based on their design and reliance on networks for mission-essential information.

For T&E processes, two concepts are key:

- The need for T&E itself to use and maintain effective cybersecurity during T&E events to protect the confidence and credibility of the T&E processes as a trusted source of truth about the autonomous system and its capabilities.
- The effective use of T&E as an evaluation process for verifying and validating that cybersecurity features and capabilities are effective within the autonomous system and its larger system of systems (SoS) needed for operational effectiveness.

DoDI 5000.90 provides requirements and procedures that drive cybersecurity T&E needs. The DoD Cyber DT&E Guidebook, Version 3.0, provides extensive details on these cybersecurity requirements and methods. For future planning, the DoD Zero Trust Strategy provides guidance for a shift in DoD culture on network trust that, although not yet implemented in cybersecurity T&E guidance, should be used in planning for the cybersecurity requirements for future systems.

Summary for DoD DT&E. This policy sets the foundation for the cybersecurity requirements of all DoD systems. The guidance set forth in DoDI 5000.90 provides a baseline for understanding the cybersecurity needs for autonomous systems T&E.

3 Background and Vision for Autonomous Systems T&E

This section introduces key concepts that help establish the background and emerging developments behind many of the current challenges for the T&E of emerging technologies. This discussion should provide new practitioners with an understanding of some underlying issues for autonomous systems T&E to understand the reasons behind the many difficult challenges that testers will encounter.

The first background concept, discussed in Section 3.1, emphasizes how AI and autonomy are fundamentally distinct but have great synergistic potential in future autonomous systems. Sections 3.2 and 3.3 examine the implications of cyber-physical systems (CPS) and how they have driven development to an agile framework. Then, Section 3.4 introduces a view on trust because it is a new focus for T&E, and Section 3.5 introduces the three main levels of autonomy in relation to human interaction. Sections 3.6 and 3.7 discuss considerations for future guidance for autonomous systems and how guidance may need to adapt in the future. Section 3.8 summarizes how T&E infrastructure is evolving for autonomous systems, and Section 3.9 outlines a future expansion of this guidebook that will include information about DoD test ranges, tools, and test organizations with specific capabilities focused on the T&E of autonomous systems.

3.1 Relationship Between Autonomous Systems and Artificial Intelligence

Autonomous systems and AI are distinct but closely linked. Understanding the AI role within an autonomous system provides insight into how AI can be effectively utilized and how T&E efforts for AI and autonomous systems can complement each other. An analogy for this relationship is an aircraft's flight control system: Although flight controls are tested as a subsystem before integration, their evaluation continues as part of the fully integrated aircraft's flight test program. Similarly, AI must be tested before integration into an autonomous system, but full validation occurs only when the AI is evaluated within the complete system under realistic conditions. Additionally, just as an aircraft consists of multiple subsystems beyond flight controls, an autonomous system integrates various components—both AI and non-AI—each requiring appropriate testing methodologies.

A graphic example helps to illustrate the relationships between autonomous systems and the AI components that are part of them. In Figure 3-1, a notional, simplistic functional view of an autonomous ground vehicle is shown as an example of AI use within an autonomous system.

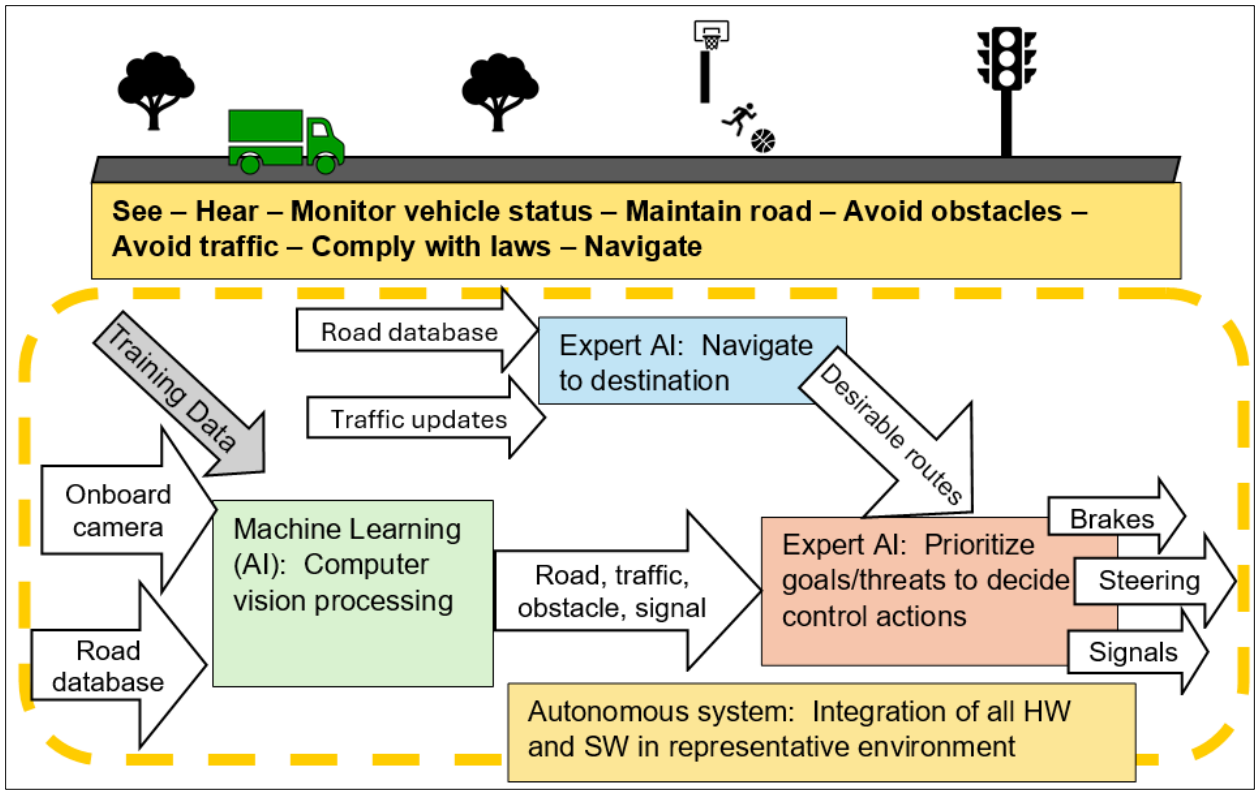


Figure 3-1. Notional Example of an Autonomous System Using AI Components

The autonomous ground vehicle has several tasks that it must autonomously perform (independently self-directed without human control). These tasks include:

- Seeing in front of and around itself.
- Hearing noises such as sirens and possibly police verbal directives.
- Communicating with surrounding bystanders.
- Monitoring its own status for fuel and other internal dynamic parameters.
- Maintaining its position on the roadway or other suitable vehicle paths.
- Avoiding obstacles such as holes, signs, barricades, vegetation, and dropped objects.
- Avoiding traffic such as other vehicles (human or autonomous), pedestrians, and animals.
- Complying with laws such as right-of-way, traffic signals, and school speed zones.
- Adapting for temporary changes such as road closures, construction, flaggers, and outages.
- Navigating to its destination to pick up or deliver its cargo (the mission goal).

These tasks may have a variety of enabling technologies that contribute to accomplishing the vehicle's overall mission safely and efficiently. As an example, an AI component utilizing ML may be used to provide computer vision processing of onboard camera video from around the vehicle. Another AI component might be the navigation software that optimizes the vehicle's routing to its destination based on roadway databases with real-time traffic updates, but this might be an "expert AI" using complex rule-based programming rather than ML. Finally, an AI component might be used to integrate the computer vision AI outputs with the navigation AI outputs to decide vehicle control actions such as braking, steering, and signaling. From this example, several AI components might integrate with other hardware and software to form a complete, autonomous system. From a T&E perspective, each AI component can be tested independently of the system first but then must be tested as it is integrated within the complete autonomous system for a robust evaluation. The scope, methods, and results of the AI components' tests should both inform and focus the system-level tests to validate AI test results as well as explore scenarios and interactions that are untestable below the integrated system level.

Another concept in understanding autonomous systems and AI is the idea of varying levels of autonomy. Autonomous systems may be empowered to execute certain tasks without any human oversight at all, whereas other situations might require human approval or other intervention for safety or other reasons—these are considered differing levels of autonomy and could exist simultaneously within the same system for different tasks or situations. In the example above, the autonomous ground vehicle might be empowered to navigate without oversight to its destination on clear roads, but the vehicle might be designed to pull over and stop if it detects a construction crew on the road ahead, signaling back to a control room for a human remote operator to intervene and navigate it through a temporary, one-lane flagger situation, before returning to full autonomous mode once it is past the construction zone. Autonomous systems T&E practitioners must have a complete understanding of what levels of autonomy and what safeguards are in place to fully test the system's AI-enabled autonomy features as well as to test its safeguards and transitions to enable effective human intervention. More details on HSI are provided in Sections 3.5, 4.2.8, and 5.1.3 of this guidebook.

In the example above, note that although an ML AI is included to illustrate how it supports the integrated autonomous system, there is no need to necessarily utilize ML for any specific components—it is possible that an autonomous system might not use ML but still qualifies as an autonomous system.

3.2 Cyber-Physical Systems

CPS are integrations of computation with physical processes. In CPS, physical and software components are deeply intertwined, are able to operate on different spatial and temporal scales,

exhibit multiple and distinct behavioral modalities, and interact with each other in ways that change with context. As CPS, autonomous systems will require the T&E of both hardware and software development and capabilities. The T&E practitioner must recognize that a hybrid approach may be necessary, where some test events occur on a pace with hardware development, but other test events occur at a higher frequency for software development. This varying development tempo creates a challenge for the project managers and the responsible test organizations to appropriately test features at the right times in development. The complex mix of hardware and software components also creates a challenge for understanding how the software and hardware interact, such that T&E events primarily intended to evaluate hardware also account for and evaluate the software, and vice versa. Finally, the integration of these components creates challenges for cybersecurity and its T&E because of the highly networked nature of CPS. Overall, the autonomous system's capabilities depend on both the hardware and software working together with correct integration and security.

3.3 Agile Development Framework

An agile development framework is a project management approach that involves breaking the project into phases and emphasizes continuous collaboration and improvement. This framework has been adopted widely in DoD systems development and acquisition for software-intensive products, utilizing the "agile" processes of near-continuous development and system integration, with rapid, repeated software updates on a relatively short cycle of time. The InfoWorld article, "What is CI/CD? Continuous integration and continuous delivery explained" (Sacolick 2024), states that "continuous integration (CI) and continuous delivery (CD), also known as CI/CD, embodies a culture and set of operating principles and practices that application development teams use to deliver code changes both more frequently and more reliably." The implementation is also known as the CI/CD pipeline. An agile methodology utilizing CI/CD is a best practice for software development and test teams to implement.

The InfoWorld article defines CI as "a coding philosophy and set of practices that drive development teams to frequently implement small code changes and check them in to a version control repository."

CD picks up where CI ends by automating software delivery to multiple use sites, such as CT, simulators, DT, and possibly OT or operators. CI and CD require continuous testing because the objective is to deliver quality software code to the end users. Continuous testing is often implemented as a set of automated regression, performance, and other tests that are executed in the CI/CD pipeline.

The use of agile development frameworks with CI/CD and continuous testing will be a major paradigm shift for DoD T&E organizations. However, the rationale for autonomous military

systems using agile processes is sound. For example, a battlefield commander could order a change in operational tactics to occur rapidly because of adversary threat changes, friendly asset changes, or simply operational considerations occurring on the battlefield. By employing an agile process, an autonomous system can respond to the commander's directives with flexibility and responsiveness. The development and execution of robust continuous testing processes will be a significant challenge for DoD organizations. These processes, however, are extremely valuable to enable highly flexible and effective autonomous systems that meet national defense objectives. Continuous testing for autonomous systems might involve feedback mechanisms from the operational user directly back to the system vendors and program managers to facilitate quick and effective software solutions to support commanders' needs in the battlespace.

3.4 Perspectives on Trust

One of the chief concerns that DoD leaders and the public have about autonomous systems, especially AI-enabled autonomous systems, is "*How can we trust the system?*" This question becomes very important because of several compounding reasons:

- Autonomous systems may operate without human control and will be empowered to act at times without the ability for humans to intervene or have complete oversight.
- Autonomous systems will need to coordinate as a participant alongside operators, bystanders, and team members in harmony and in support of human missions.
- DoD operational environments can be extremely varied and diverse, including weather, terrain, lighting, temperatures, and electromagnetic interference.
- DoD operational scenarios can be incredibly complex and may involve varied threats, numerous possible friendly forces, and a crucial need to account for noncombatants, with diverse objectives.
- Environmental conditions and operational scenario features can both change rapidly.
- Autonomous systems, especially those utilizing AI, may operate as a "black box" where the direct mapping of inputs to outputs may be impossible to fully understand.
- Adversaries will attempt to disrupt and exploit DoD systems' operations.
- Autonomous systems may be empowered to use lethal force or otherwise endanger lives and property during their operations.

Important concepts to understand for autonomous systems are the ideas of *trust* as a personal, human feature that will vary based on many factors, versus *trustworthiness* as a system feature that can be more objectively measured. Trustworthy systems first and foremost must have effective performance. Trustworthy systems also need safety, security, reliability, availability,

maintainability, the ability to be understood, the ability to self-report problems the system cannot handle, and the ability to be controlled by humans when these features are violated.

T&E plays a crucial role in establishing autonomous system *trustworthiness*, which helps to establish human trust. Effective, robust T&E can characterize the autonomous system's performance and trustworthiness:

- When (under what conditions and scenarios) the system will perform effectively.
- When the system is ineffective, unreliable, unsafe, vulnerable, or otherwise problematic.
- When there is uncertainty about system performance and trustworthiness.

A key feature that T&E must also address is the ability of the autonomous system to be understood. Many humans may interact with the system in various roles: operator, maintainer, engineer, programmer/developer, manager, teammate, ally, mission commander, operational commander, and even the test personnel themselves. The autonomous system should have *human-system interfaces* to provide each of these human perspectives with enough information for them to do their jobs in operating, maintaining, improving, reprogramming, managing, commanding, teaming, and testing the system effectively and efficiently. Figure 3-2 presents three orthogonal axes showing different elements of trust in autonomous systems.

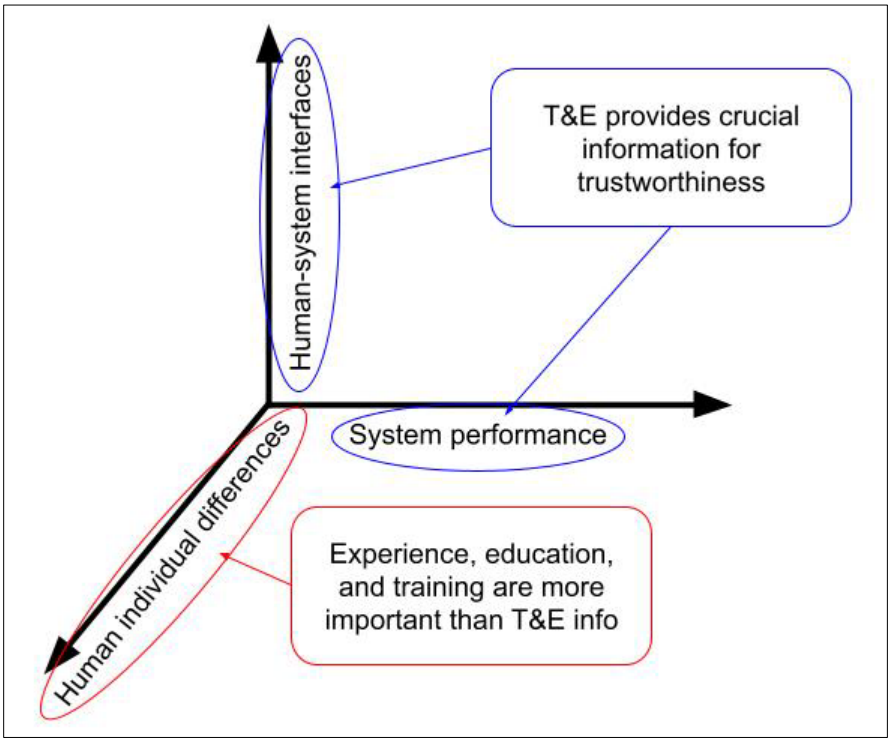


Figure 3-2. Factors of Trust in Autonomous Systems

Human individual differences based on experience, education, training, personality, and other factors also have a significant impact on trust in autonomous systems. These personal factors vary widely and should be managed by military leadership authorities during education, training, and operation. Practices provided in Section 5 of this guidebook address the measurement of human personal trust in some cases, but they do not attempt to address how to improve or optimize human individual differences beyond the features of the autonomous system under test (SUT).

In summary, the effective and robust T&E of autonomous systems provides two crucial categories of information to evaluate the *trustworthiness* of the system:

- Insight into the scenarios and conditions where the system is effective, ineffective, or unpredictable, as well as insight into other trustworthiness factors.
- Insight into the effectiveness of the human-system interfaces needed for the numerous human perspectives to understand the autonomous system.

The challenges discussed in Section 4 and the methods and practices presented in Section 5 address in more depth the details of many of the factors involved in autonomous system trustworthiness and its evaluation.

3.5 Human Systems Integration Aligned to Autonomy

This section discusses the human interaction aspects of autonomous and semi-autonomous systems, specifically detailing the levels of human involvement. Three types of human interactions exist for varying levels of autonomy: human in the loop (HITL), human on the loop (HOTL), and human out of the loop (HOOTL). Figure 3-3 shows a process flow diagram of these three different types of human interaction with an AI tool. Section 5 of this guidebook provides recommendations for the testing of each of the three levels.

HITL systems require a human to be actively involved in the decision-making process, where the autonomous system provides recommendations, but a human must approve or reject them. In contrast, HOTL systems have humans monitoring the autonomous system's performance, but they only intervene when necessary, such as in cases of exceptions or anomalies. Finally, HOOTL systems operate independently, without human oversight or intervention, making decisions fully autonomously.

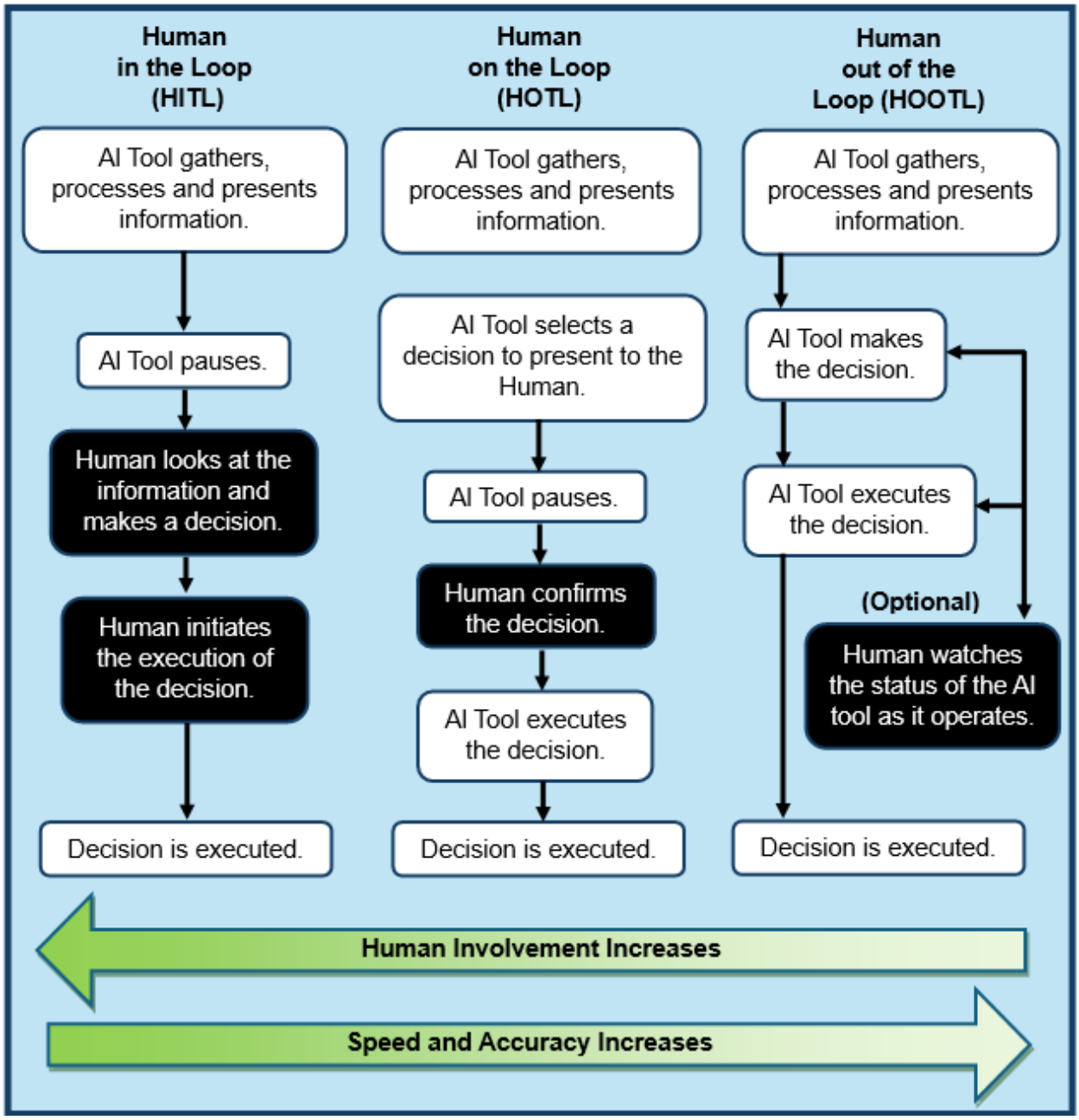


Figure 3-3. Levels of Human Interaction

A single autonomous system may act in any or all three of these interaction levels at different times, depending on the system design and scenario. These varying interaction levels create challenges for T&E, in that the tester must evaluate the system both with and without human interactions, including fail-safe mechanisms where human intervention may be expected but absent, as well as incorporating T&E of the transitions between each of the three interaction levels as a scenario unfolds. Appropriate and smooth transitions from HOOTL to HITL at critical times can be key to the trustworthiness of the autonomous system.

By addressing these interaction differences and including the human element of performance contribution along with the autonomous system performance, the T&E practitioner can develop a more comprehensive understanding of the impacts of human-autonomy teaming (HAT) on mission outcomes. The intended effect enables system developers to design systems that effectively leverage the strengths of both humans and autonomy agents, ultimately leading to improved mission outcomes.

Sections 4 and 5 discuss more details on the challenges of HAT and the best practices to address the challenges.

3.6 Future-Proofing Guidance

Creating policy that stands the test of time is challenging. It requires the intent of the policy to be clear and relevant even after expected—but not always predictable—advancements as well as completely unanticipated changes in technology, the global landscape, and other critical areas. The incredible difficulty of this task leads to a need to plan regular review and update cycles for written policy to incorporate necessary changes to deal with unpredictable and unanticipated changes.

Policy should focus on setting, identifying, and specifying outcomes, decisions, and processes rather than specific metrics or measurements to direct development and drive results. Specific metrics related to advancing technology tend to be overcome, sometimes very rapidly, leading to policy becoming out-of-date. Metrics also face the dilemma of Goodhart's Law: "When a measure becomes a target, it ceases to be a good measure."

Even more challenging than mere technological progress is the tendency for clearly established targets or restrictions, such as a numeric compute threshold, to incentivize solutions that aim at them or avoid them. The Center for Naval Analyses recently published a paper on Goodhart's Law with several examples of this challenge in defense programs (Stumborg et al. 2022).

In contrast, DoDD 3000.09 focuses on identifying key decisions or actions as thresholds—selecting and engaging a target—and establishes a process to ensure senior-level review of certain systems that avoids defined metrics but could readily adapt to changing technology. This

method was successful enough that the original directive, published in 2012, was reviewed and updated in 2023 with minimal changes. Even with the incredible advancement of AI and autonomy in the intervening decade, the directive needed only minor changes such as adding references to offices and requirements that did not exist in 2012, including the Chief Digital and Artificial Intelligence Office (CDAO) and the DoD AI Ethical Principles.

These examples highlight the importance of creating flexible policy that emphasizes desired outcomes and processes as well as regularly reviewing and updating the policy.

3.7 Evaluating Guidance in Response to Novel Capabilities

Novel capabilities and technologies have the potential to enable new ways to achieve mission success. Some may even enable entirely new missions. However, novel capabilities may also create or exacerbate gaps in existing policies, regulations, and guidance (referred to in this section as "guidance") on how to ensure that these technologies and their uses support the values and principles of DoD. At the farthest edges, novel capabilities may even undermine or create perverse incentives regarding existing guidance.

Missile defense provides an example of novel technology highlighting gaps and potentially undermining the intent of existing guidance. The presence of a human for decision-making in autonomous systems may seem to support key values and goals of DoD by having a human review the decisions in real time to ensure the appropriate levels of human judgment. However, the human may not actually have that effect. The short window of time for response to an incoming munition may preclude a human operator from responding effectively. Human operators either may be unable to intercept the munition, leading to potential loss of life or other damage, or may blindly trust (or refuse to trust) the system, preventing the human from applying appropriate judgment (Hawley 2017).

Another example is the current acquisitions approach to T&E. Traditional acquisition pathways assume that a system is static after deployment and will not need additional T&E. Autonomous systems, however, may display emergent behavior in response to changing real-world scenarios. Ensuring that this behavior is beneficial and does not create undesired consequences must necessarily be part of the T&E of these systems. Guidance then needs to be assessed to determine whether programs are properly resourced, incentivized, and structured to conduct T&E across the full life cycle, including after transitioning and fielding.

It is crucial that technological adoption and innovation not outpace written guidance in a way that puts the deployment of novel capabilities ahead of the DoD ability to conduct T&E. However, it is also important that a lag in guidance does not prevent the United States from maintaining parity with or dominance over its adversaries or allow for technologies that

undermine its values as a nation. To achieve effective and timely guidance, technologists and developers, policymakers, and T&E communities need understanding and alignment across stakeholders and the entire system life cycle to facilitate adaptive and responsive policy review rather than reactive and post hoc evaluation. Policy stakeholders need to understand issues as they arise, and developers need to communicate potential concerns to their policy colleagues.

3.8 Test and Evaluation Infrastructure Advancements

Recent and ongoing advances in AI and autonomy T&E are revolutionizing how DoD prepares for future warfare. The Test Resource Management Center (TRMC) is at the forefront of this transformation, investing in pioneering infrastructure to support the T&E of intelligent unmanned systems for the joint warfighter. This research and development (R&D) effort is aligned with four key areas within the USD(R&E) strategy: Trusted AI and Autonomy, Integrated Network Systems-of-Systems, Advanced Computing and Software, and Human-Machine Interfaces. This strategy marks a groundbreaking initiative within DoD, aiming to modernize and accelerate T&E capabilities supporting these key areas. For more information, see the DoD Critical Technology Areas Website (https://www.cto.mil/osc/critical-technologies/).

To proactively modernize DoD test infrastructure to support T&E requirements, TRMC is focused on three core areas: developing modernized test strategies for emerging AI and human-machine interface technologies; understanding the necessary test infrastructure services to support these test strategies; and optimizing the application of these capabilities across the Major Range and Test Facility Bases, mobile experiments and test environments.

Over the past 4 years, critical “test force multipliers” have been identified to accelerate AI and human-machine interface capabilities for the joint warfighter across all warfare domains (from the seafloor to space). These force multipliers include enhancing ML operations at the test edge; updating AI software at mission speed; and improving big data management services to support AI from the tactical edge to the laboratory. Efforts are also directed at achieving over-the-air updates for evolving algorithms; improving research and engineering network speeds and satellite communications; and implementing automated data management services to reduce manual data processes.

Investments in scalable, on-demand digital infrastructure, supported by hybrid-cloud services, are enabling test managers and SUTs to provide the necessary resources for rapid training, validation, and testing of AI and human-machine interfaces. Furthermore, joint developmental and operational test infrastructure is being developed to support realistic experimentation across all warfare domains. Innovative research into autonomy and AI trust aims to create metrics for validating and ensuring trustworthy machine behavior. Through these targeted efforts, DoD is set

to revolutionize its T&E practices, enhancing readiness and resilience in an era of rapid technological change and evolving threats.

3.9 T&E Resources: Test Ranges, Tools, and Organizations

A future expansion of this guidebook will provide information about DoD test ranges, tools, and test organizations with specific capabilities focused on the T&E of autonomous systems. This expansion is planned to include:

- Resources for the T&E of autonomous systems:
  - DoD test labs and test ranges as well as other test facilities.
  - Hardware resources: testbeds, surrogates, platforms, etc.
  - Software resources: assurance cases, requirements analysis, runtime monitoring, etc.
  - Simulation environments.
- Examples and case studies:
  - Land autonomy.
  - Sea autonomy.
  - Air autonomy.
  - Space autonomy.
  - Swarm autonomy.

Every effort will be made to ensure that these resources are complete and up-to-date, and this information is available to a wide audience.

4 Challenges

Autonomous systems involve unique T&E challenges that derive from the new system features and capabilities, such as perception, learning, reasoning, deciding, teaming, and emergent behavior, which may include unpredictability. The autonomy T&E community, including both researchers and practitioners, has collectively identified many specific challenges. This section introduces the most urgent of these challenges.

One overarching challenge that potentially will require the most significant cultural changes within the DoD T&E enterprise is the concept of a paradigm shift from the traditional, more segregated T&E of hardware processes to a new concept of continuous testing for a system centered on software. The second overarching challenge stems from the nature of autonomous decision-making, which moves beyond simple condition-response mechanisms to a dynamic cycle of observe, orient, decide, and act (OODA loop). Originally developed for military decision-making, the OODA loop describes how information is continuously processed and how humans or systems adapt to changing conditions and take action. Autonomous systems operate within this framework throughout a mission, requiring T&E approaches that assess their ability to respond to evolving situations in real time.

Section 4.1 discusses the two overarching challenges for autonomous systems T&E, and Section 4.2 explores specific challenges posed by autonomous system capabilities. The complete list of challenges, with direct links to each corresponding section, is provided below:

To help organize the information, each challenge is discussed in a standardized format that includes the specific challenge, the challenge details and risks, and an introduction of methods and practices that address the challenge. The matrix in Section 4.3 cross-references each challenge with each method or best practice that helps address the challenge.

4.1 Overarching Challenges for Autonomous Systems Test and Evaluation

4.1.1 Adapting to Developmental Test and Evaluation as a Continuum

The 21st century has brought a shift to DoD from traditional, primarily hardware-focused combat systems to systems that are heavily reliant on complex software for their primary mission functions. The acquisition system growth of a software acquisition pathway, as well as the DoD Data, Analytics and AI Adoption Strategy, clarifies that the development, integration, fielding, and sustainment of autonomous systems utilizing complex software and AI will have major, lasting effects on the systems' required T&E. These agile and iterative acquisition processes for AI and data-driven systems, which include autonomous systems, demand critical change in how T&E supports capability delivery and becomes agile and iterative throughout the life cycle. T&E will extend both "left" into early development and "right" into post-acceptance and post-fielding product improvement to ensure more complete and efficient knowledge transfer across the engineering life cycle and more effective communication of data needs across all phases of development. Therefore, an overarching challenge for autonomous systems T&E is adapting to this paradigm change, which moves T&E from a serial set of activities conducted largely independently of systems engineering and mission engineering activities to a new agile and integrative framework focused on a campaign of learning termed *developmental Test and Evaluation as a Continuum* (dTEaaC).

developmental Test and Evaluation as a Continuum

This new dTEaaC paradigm is a change:

- From the DoD T&E traditional "waterfall" process using a sequential series of test programs with early S&T, vendor testing, DT, OT, and follow-on post-acceptance T&E all as discrete, sequenced stages primarily supporting key milestone decisions.
- To a new agile, iterative test continuum approach where T&E provides focused and relevant information supporting decision-making continually throughout capability development from the earliest stage of mission engineering through operations and sustainment:
    - Starting at the earliest phases of S&T, prototyping, and experimentation to develop and mature technology.

- Into traditional program-of-record DT (both contractor and government).
- Beyond fielding for systems that continue to evolve (learning and agile software).

As shown in Figure 4-1, this paradigm change involves T&E across a continuum in several ways:

- The continuum of a system life cycle—when the T&E occurs.
- The continuum of system change frequencies—how often T&E occurs.
- The continuum of decision information needs—what needs the T&E feeds.

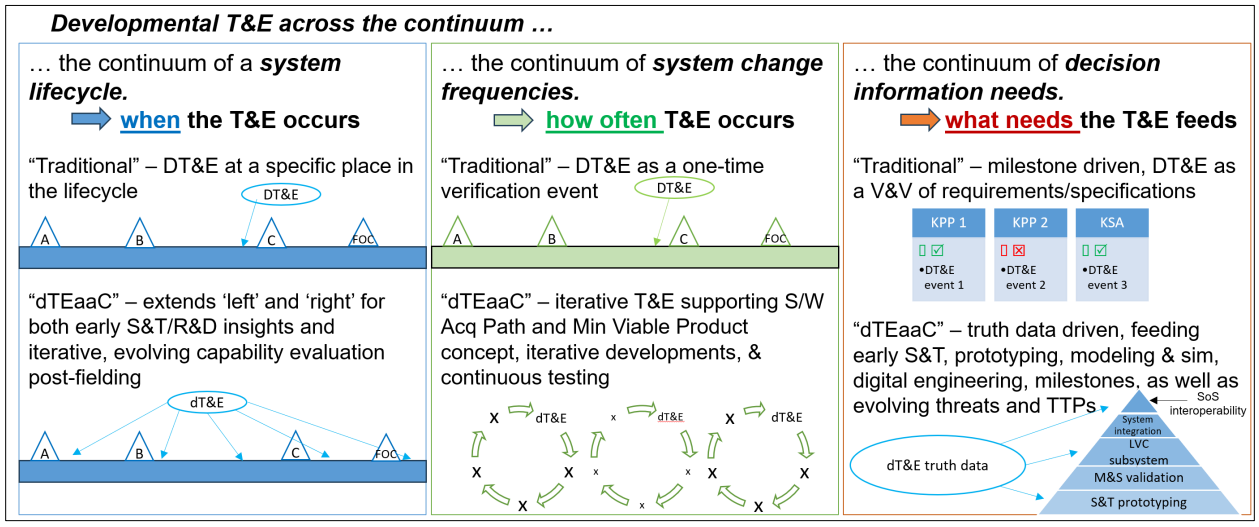


Figure 4-1. Developmental T&E as a Continuum (dTEaaC)

dTEaaC Challenge Details

The paradigm change to T&E as a continuum creates many challenges that cut across all aspects of autonomous systems T&E.

Underlying factors. The DoD effort to continuously develop and deploy software-intensive systems creates this challenge. Additionally, independent testers have a responsibility to provide test information at enough speed to give timely insight into performance and trustworthiness, without compromising the credibility of their conclusions and recommendations. The conflict between T&E speed and T&E quality amplifies this overarching challenge.

Related risks. Failure to adapt to dTEaaC may result in:

- Delayed and more costly deployment of autonomous systems due to the lack of accurate T&E information during development and technology maturation.

- Reduced capabilities due to the lack of effective T&E data and conclusions in time to inform major design, development, integration, or implementation choices.
- Incorrect models and invalid simulations being used for system decisions due to the lack of truth data validating modeling accuracy and precision.
- Delayed and more costly capability deployment of updated, improved autonomous systems due to the lack of T&E information verifying their functionality and validating their effectiveness.
- Reduced capabilities due to the lack of effective T&E data and conclusions before deployment, shifting risk that the updated autonomous system has deficiencies and is ineffective or untrustworthy to the end user.
- Outdated autonomous system capabilities from the failure to update and evolve the system as threats, tactics, and priorities change over time.

Affected individuals. This challenge affects everyone involved in autonomous systems; success or failure in adapting to T&E as a continuum will be dependent on or will affect testers, program managers, researchers, developers, engineers, maintainers, commanders, requirements staff, contracting officers, and operators.

Trade-offs, limitations, or assumptions. T&E as a continuum becomes more challenging and critical when:

- Early R&D success depends greatly on realistic T&E.
- Early T&E results lead to major changes in the requirements, CONOPS, or design.
- The continuous system updates occur more frequently (weekly or even daily).
- The continuous system updates affect safety, security, or other major capabilities.
- The understanding of the system's capabilities, interactions, mission effectiveness, and trustworthiness relies on the accuracy and realism of complex models and simulations.

Nine of the 13 specific challenges discussed in subsequent subsections provide more details about the specific difficulties and risks associated with this overarching challenge.

Methods and Practices

The dTEaaC guidance addresses how data evaluation can occur across all analyses and studies; live, virtual, and constructive (LVC) testing and M&S; and test activities while being rooted in a common learning construct. By aligning all data-driven activities across the life cycle,

engineering and acquisition professionals can gather data more efficiently and evaluate data more holistically.

The dTEaaC methods are built on three core tenets:

- **Deliberately Executing a Campaign of Learning**: Integrating knowledge needs and learning opportunities across the entire engineering life cycle.
  - Deliberately plans knowledge transfer through the integration of data capture and evaluation activities across the solution life cycle.
  - Promotes mission validation via an agile, data-driven, and knowledge-based framework.
  - Defines engineering life cycle data needs (continuous and integrated) independently from acquisition life cycle processes (discrete and sequential).
  - Provides for greater understanding of S&T contextualization in the perspective of final capability delivery.
  - Diminishes legacy distinctions between data types based on capture source and transitions to data categorization based on context and data use (evaluation).
- **Data-Driven Decision-Making**: Embracing the assistance of decision support systems (models, AI, data dashboards, etc.) to inform critical acquisition or make-or-buy decisions.
  - Enables earlier discovery of opportunities and defects by providing more complete and nuanced insights to decision-makers.
  - Facilitates the embedding of mission-driven requirements and contexts across all decisions by better integrating how data are communicated.
  - Enables more effective data evaluation planning (including test, experimentation, and virtual-constructive (VC) efforts) to increase the efficiency of data capture, management, and exploitation investments.
  - Provides ongoing and continuing learning feedback to build better understanding over time.
  - Defines requirements for the implementation of a Decision Support Evaluation Framework and an Integrated Decision Support Key to aid in consistent decision support system composure and use.

- **Leveraging Digital Ecosystems**: Exploiting readily available digital ecosystems instead of one-off or stand-alone tools, data repositories, and workflows.
    - Employs digital engineering to integrate mission engineering, systems engineering, and traditional T&E data into a single digital artifact or integrated knowledge management environment.
    - Integrates with ongoing digital workforce initiatives and curriculum to leverage the skillset of industry and academic partners and the modern government workforce.
    - Supports the establishment of a digital thread continuum that simplifies the transformation for digital adopters by indicating known connections between tools, ecosystems, and data repositories.
    - Facilitates knowledge transfer by implementing multi-stakeholder, multiphase information architectures.

The methods and best practices listed below and described in Section 5 can help address this overarching challenge with specific practices that address dTEaaC.

- Scientific test and analysis techniques (STAT) for autonomous systems.
- Open system architecture.
- Small-scale development.
- Continuous testing.
- Code isolation.
- Assurance cases.
- LVC testing.
- Experimentation T&E.
- Surrogate platforms.
- AI model testing.
- Post-acceptance testing.
- Cognitive instrumentation.
- Runtime assurance.
- Automatic domain randomization.
- HAT performance.
- Task-based certification.

- Quantified risk and performance growth curves.

4.1.2 Test and Evaluation of the Observe-Orient-Decide-Act Loop

The OODA loop is a decision-making model used to understand how intelligent agents think, learn, and make decisions. Understanding and testing an autonomous system's OODA loop is critical for understanding how an autonomous system makes decisions, predicting what to expect when the system encounters various test scenarios, and determining the testing required to validate the system's decision-making quality. Failure to understand this decision-making process and the implications of each step can result in user "surprise" by system actions and contribute to potential user distrust of the system. Testers must evaluate OODA loop processes to characterize them accurately and identify deficiencies to ensure autonomous system trustworthiness for users.

Challenges of T&E of the OODA Loop for Autonomous Systems

The challenges of T&E of the OODA loop for autonomous systems are critically important including these key issues:

- Autonomous system effectiveness depends on the correct and complete performance and trustworthiness of all stages of the OODA loop process; therefore, system testing is required to determine the effectiveness and deficiencies for each stage of the system's OODA loop.
- OODA loop component operations can be highly sensitive to inputs from other components, which causes comprehensiveness concerns for system integration DT&E.
- Latency, noise, miscalibration, or failures in one OODA loop component can escalate into problematic conditions throughout the rest of the process.
- Design and implementation of the autonomous system's initial settings, world models, state spaces, and other internal OODA loop software features can greatly affect performance while being difficult to observe and measure during evaluation.
- Software components with incorrect assumptions about the system hardware capabilities can cause limitations, complexities, and internal conflicts.
- HSI factors including cognitive workload, trust calibration, human-machine communication, and shared situational awareness play a crucial role in the OODA loop process. The CDAO HSI T&E of AI-Enabled Capabilities framework provides mappings of 13 HSI concepts that align with different stages of the OODA loop.

Example. As shown in the OODA loop model of a sample autonomous undersea vehicle depicted in Figure 4-2, the *Observe-Orient-Decide-and-Act* process has many unfolding interactions. The *Observe* and *Act* phases interact directly with the environment and external entities. These activities also translate the physical to digital domain (*Observe*) or digital back to physical domain (*Act*). *Orient* and *Decide* activities reside entirely in the digital domain and can require intensive edge processing and computing capabilities. Understanding this cycle can help shape the validation testing required and can frame the presentation of evidence required to establish and communicate trust in the system to the warfighter. Simulation of the physical domain enables the exploration and testing of hypotheses about the performance of the *Orient-Decide* (world model and autonomy software) activities and capabilities, accelerating the understanding of the system performance in a wide range of potential operating environments.

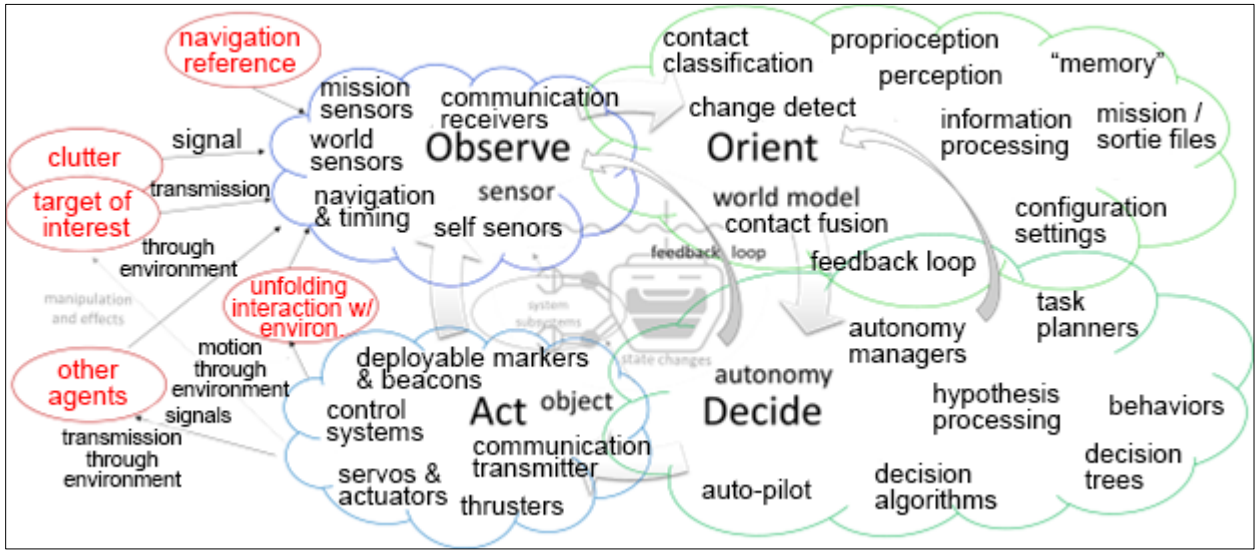


Figure 4-2. OODA Loop Model of a Sample Autonomous Undersea Vehicle

OODA Loop Challenge Details

The challenges of the OODA loop in autonomous systems create several difficulties for T&E. Breaking down the process's steps can help to explain these challenges.

1. Observe

The intelligent agent observes with sensors. For the purposes of this section, sensors include mission sensors, environmental sensors, navigation sensors, and communications receivers. The act of observation is primarily the detection and translation of "signals" into digital information about the environment, potential targets, and the agent itself. This observation includes the signal processing (sampling, beam forming, filtering, etc.) required to digitally process incoming time series data about the unfolding interaction with the world for use by the system. The observe phase can also include human inputs and interactions through sensors or communications.

A sensor's capability to discriminate important signals from background noise and clutter is dependent on the physical design of the sensor; the stability of the sensor; the digital signal processing; the origin and nature of the signal; the transmission path and losses of the signal through the environment; and the background noise of the environment. Sensors provide dynamic inputs to the autonomous system, translating the agent's view of the world into the agent's digital decision-making process. Therefore, the performance of the autonomous system is tightly coupled to the sensor performance. Potential challenges during system testing include:

- Testing sensor noise and biases.
- Testing sensor resolution and calibration.
- Testing sensor performance coupling and sensitivity to environmental conditions.
- Evaluating self-monitoring of sensor performance.
- Evaluating sensor failure modes and system responses.
- Evaluating how necessary or sufficient the sensors are to inform the decisions required to achieve mission success.
- Testing a myriad of human inputs and human-generated communications.

2. Orient

The intelligent agent uses the world model, system configuration, and mission settings to orient observed information. Orientation includes translating sensor information into perception and proprioception, contextualizing and prioritizing information about the environment, targets, and system. Capabilities such as automatic target recognition, change detection, and contact fusion bring together information from the system's "memory" with direct observations from sensors to draw conclusions about the meaning (significance or relevance) of these observations.

For fully autonomous systems, the system's world model, configuration settings, and mission plan or script place a boundary on the system "understanding" of observed information. Even in a dynamic learning system, where feedback loops enable this understanding to evolve over the course of the mission, the system's orientation capability is limited by the available settings. Potential challenges during system testing include:

- Evaluating how observation underpins perception.
- Testing diverse iterations to understand the sensitivity of mission success to initial settings.
- Evaluating agent understanding based on initial settings, updated by system experience.

- Evaluating whether observation without orientation leads to reflexive actions rather than decisions.

For HITL or HOTL systems, the operator's own awareness, training, and remote observation of sensor data provide additional system capability to orient the data collected by sensors. Potential challenges during system testing include:

- Evaluating sensor data display or visualization quality and latency.
- Accounting for operator experience, training, and proficiency.
- Assessing mechanisms for bidirectional human-machine communication for situational awareness.

3. Decide

The intelligent agent uses autonomy software algorithms to decide what the system (and subsystems) will attempt to execute. Feedback loops provide explicit guidance and control over how observation and orientation are conducted by the system. Intelligent systems use algorithms to plan, test, and select proposed sequences of elementary moves or behaviors. Autonomy managers and arbiters assign resources, prioritize goals, and select between competing courses of action. As in orientation, the choices available to the system are defined by the initial autonomy algorithms. In some systems, humans may also have oversight of some decisions. Potential challenges during system testing include:

- Planning for the complexity of decision space.
- Evaluating the management of budgets (time, energy, power, processer cycles, memory, etc.).
- Testing responses to off-nominal conditions.
- Evaluating the adjudication of conflicting behaviors.
- Testing the boundaries of the behavior space.
- Evaluating the resilience of autonomy behaviors.
- Evaluating the self-governing behaviors that shape observation and orientation.
- Efficiently and effectively managing software regression testing.

4. Act

An agent's utility depends on its ability to act effectively on the environment, translating decisions into physical or digital actions that achieve objectives. Potential physical actions could

include changes to the mission profile, intentional manipulation of the environment or targets, target engagement, initiation of communications with another agent or operator, placement of markers, or many other events; human interaction may or may not be part of this phase depending on the system and circumstances. Although many actions involve the system's physical components interacting with the environment, this step of the decision-making cycle also encompasses nonphysical actions, such as electronic warfare, cyber operations, or digital communication, which are equally critical to achieving mission objectives. Potential challenges during system testing include:

- Planning for the physical architecture.
- Evaluating environmental limits and effects on the execution of actions (operational envelope).
- Testing subsystem failure rates and reliability.
- Evaluating system resilience and redundancy requirements.
- Testing control system stability.
- Evaluating the integration with other systems and agents, including humans.

Related risks. Failure to address T&E of the OODA loop challenges may result in:

- Deficient autonomous system performance for unknown or unexpected reasons.
- Surprising system behaviors during operations.
- Integrated system failures and deficiencies despite "fully functioning" components.
- Overestimation of the system capabilities due to testing with only ideal component inputs, lacking the complexity, noise, and latency of actual hardware and environments.
- Delayed and more costly development, testing, and fixes due to inadequate insight into the root causes of deficiencies, based on the misidentification of OODA loop problems.
- Evaluations that cannot accurately characterize trustworthiness due to limited data on the internal communications, states, priorities, and decisions of the autonomous system.

Affected individuals. T&E of the OODA loop challenges will affect all who design, integrate, manage, evaluate, and rely on autonomous systems.

Methods and Practices

The methods and best practices listed below and described in Section 5 can help address the challenges of T&E of the OODA loop for autonomous systems:

- T&E for autonomy M&S.
- Operational modeling.
- Open system architecture.
- Continuous testing.
- Assurance cases.
- LVC testing.
- Surrogate platforms.
- AI model testing.
- Adversarial testing.
- Post-acceptance testing.
- Cognitive instrumentation.
- Test user interface.
- HAT performance.
- Automatic domain randomization.

4.2 Specific Challenges for Autonomous Systems Test and Evaluation

The list of specific challenges described in the following subsections was developed by research and collaboration with many stakeholders and experts from both the autonomy and T&E communities. Other lists of challenges exist that are described differently but overlap. The specific titles are not critical if they sufficiently frame the issues.

4.2.1 Requirements Challenges

Writing effective requirements is not an easy task, as requirements definition is one of the most important and difficult aspects of DoD acquisitions. Furthermore, issues in requirements management are often cited as major causes of project failures. Effective system requirements are specific, verifiable, clear, accurate, feasible, necessary, consistent, and explicit; well-constructed requirements typically lead to well-executed system design, development, and T&E. However, requirements are a particular problem for autonomous systems. In theory, a set of requirements is sufficient for a third party to design and integrate an acceptable system and for a test organization to establish a suite of tests that will evaluate whether the system meets its requirements. In practice, however, although requirements for hardware, power, and

communications subsystems are tractable, the requirements for autonomous behavior are often not tractable.

Challenges of Requirements for Autonomous Systems

The challenges of requirements for autonomous systems involve the following basic issues:

- Requirements intended for autonomous system behavior are often too broad or too narrow, incomplete, inconsistent, subjective, untestable, or poorly defined.
- Constraining the operational environment and operating conditions to simplify the statement of accurate requirements may not allow adequate requirements specification, especially because autonomous system development and testing require a clear CONOPS.
- T&E of autonomous systems (like all T&E) begins with understanding the requirements; therefore, effective, efficient, and robust T&E is nearly impossible when the autonomous system requirements and CONOPS are problematic.

Example. Autonomous system behavior as seemingly simple as avoidance poses significant difficulties in requirements definition, for example, where a team of systems engineers is attempting to define a requirement that a robot arm avoid objects in its workspace while grasping a tool.

Often, the behavior specification process devolves into one of two cases. In one case, illustrated in Figure 4-3, the behavior may be specified in such detail that nothing is left for the autonomy designer to accomplish. In this case, because the systems engineers creating the requirements lack expertise in autonomy, the design is likely to include incompatible requirements (e.g., “the arm shall not contact objects” and “the arm shall grasp the tool”) and loopholes that introduce risk and fragility into the behavior design (e.g., “the arm shall not contact the objects listed in reference X” where reference X cannot include all the objects that may enter the environment that should be avoided). In the other case, all aspects of autonomy may be eliminated as “untestable” or “not specific enough” from the requirements set except at the highest level, without providing enough information for the designer to determine what the actual autonomy needs.

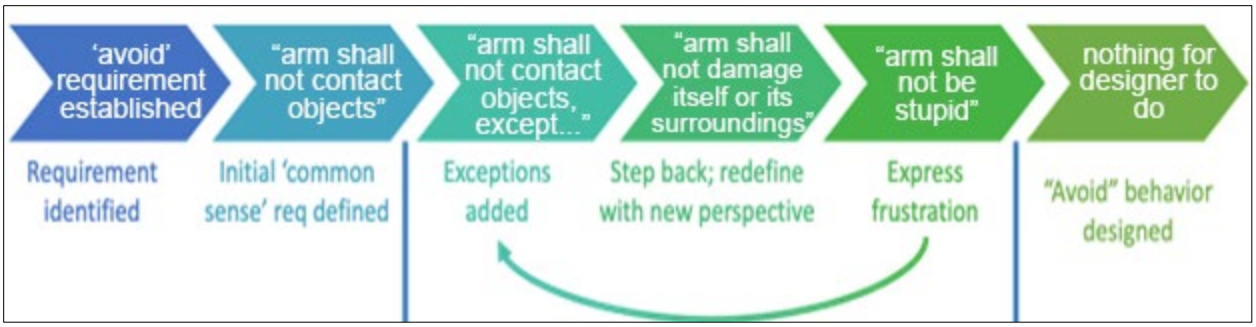


Figure 4-3. Sample Autonomous Behavior Requirement Development Process – Non-Autonomy Specialists

Requirements Challenge Details

The challenges of requirements for autonomous systems create several difficulties for T&E.

Underlying factors. Autonomous systems operate in uncertain environments. Therefore, the details of the scenarios the system may encounter are fundamentally unknown at the time the requirements are being developed—full prior knowledge is effectively impossible.

Related risks. The failure of T&E to address requirements challenges may result in:

- Ambiguity in the required autonomous behaviors. Frequently, a behavior that is desirable under one set of circumstances is undesirable in others. Because the purpose of an autonomous system is to decide what action to take, both the available actions and the circumstances under which specific actions are or are not correct must be specified to create complete and correct requirements.
- Misapplication of the autonomous system concept of employment (CONEMP) and/or its CONOPS. The system CONOPS is defined based on the information that humans need to know, meaning that traditionally, many concepts and assumptions are unstated because humans can be relied on to infer them correctly without explicit reference. CONOPS information must be made explicit for the autonomous system behavior designer and for the autonomous systems T&E personnel.
- Adverse or unanticipated interactions between autonomous systems components—for example, an acoustic communications array may interfere with a sonar sensor; a navigation algorithm may make assumptions about platform mobility or the environmental complexity that can be detected by a perception algorithm; or other hardware specifics may dramatically affect observed behavior.

- Incompatible autonomy component algorithms that make decisions and interact with each other in unpredictable ways because the decision-making elements themselves may be black boxes such as ML controllers or use deeply buried, undocumented optimization criteria.
- Incompatibilities between multiple autonomous systems interacting in the same scenario. Interoperability requires the common understanding of information being communicated and shared goals, but the autonomy community is currently developing standards to describe only simple task structures.

Affected individuals. Requirements challenges affect everyone involved in autonomous systems, including testers, program managers, researchers, developers, engineers, maintainers, commanders, requirements staff, contracting officers, and operators.

Trade-offs, limitations, or assumptions. Requirements issues are more challenging when:

- The autonomous system operating conditions and scenarios are complex, nuanced, or difficult to accurately describe.
- System components and their interactions are numerous and varied, especially including complex AI software with many diverse inputs and outputs.
- System interactions with other systems are complicated and nuanced.
- The autonomous system often interacts and/or coordinates with human or human-controlled partners, managers, and/or customers.

Methods and Practices

The methods and best practices listed below and described in Section 5 can help address the challenge of requirements for the T&E of autonomous systems:

- STAT for autonomous systems.
- Operational modeling.
- Open system architecture.
- Autonomy requirements and specifications.
- LVC testing.
- Formal verification methods.
- System-Theoretic Process Analysis (STPA) for autonomy.
- Human performance standards.

- Task-based certification.
- HAT performance.

4.2.2 Autonomy Infrastructure Challenges

The T&E of autonomous systems involves multifaceted infrastructure challenges across the life cycle of autonomous systems, as well as for their SoS.

Infrastructure Challenges for Autonomous Systems

The challenges of infrastructure for autonomous systems involve these basic issues:

- Test infrastructures, including both hardware and software tools and assets, are outdated, having been designed for traditional or legacy systems.
- Current test ranges lack automated and sophisticated software and testing tools able to evaluate the capabilities, behaviors, and interactions of autonomous systems and their interconnected SoS, as well as the scalable information technology backbones such as computational power, data storage, and analytics capabilities needed to run these advanced tools.
- M&S tools and services to support realistic, operationally representative testing of autonomous systems are not developed or are in their infancy.
- Infrastructure supporting the safety and human-system teaming aspects of autonomous systems T&E are likewise not developed or are in their infancy.
- The DoD community lacks collaboration tools for effectively integrating the people who develop, design, engineer, manage, test, maintain, and use autonomous systems.
- The few autonomy T&E capabilities that exist have been developed primarily by and for a single, specific autonomy project or program, leading to potential test infrastructure duplication and a lack of centralized, modernized institutional test capabilities.

Infrastructure Challenge Details

The challenges of infrastructure for autonomous systems present many issues.

Underlying factors. Autonomous systems introduce many new challenges that require test infrastructure improvements as part of their solutions.

Related risks. The failure of T&E to address infrastructure challenges may result in delayed, deficient, or overly costly autonomous systems acquisitions due to:

- Lack of trustworthy, scalable, and efficient test safety solutions for autonomous systems.
- Lack of effective data management solutions for autonomous systems.
- Lack of effective characterization of systems with black box components.
- Lack of understanding of security vulnerabilities.
- Lack of effective human-system teaming evaluation.
- Delays, added costs in system development, or deficiencies due to miscommunication or poor coordination among stakeholders.

Affected individuals. Infrastructure challenges affect those who develop, manage, evaluate, and rely on autonomous systems.

Trade-offs, limitations, or assumptions. Infrastructure issues become more challenging when:

- The autonomous system is highly interconnected in an SoS.
- The autonomous system requires complicated human-machine teaming (HMT).
- Existing simulation capabilities are inadequate for realistically and comprehensively evaluating the system in a mission scenario.
- The autonomous system involves sensors, datalinks, or other information inputs and outputs at varying classification levels with separate, stove-piped support.
- New capabilities emerge faster than the infrastructure envisioned to support them.

Methods and Practices

The methods and best practices listed below and described in Section 5 can help address the challenges of infrastructure for the T&E of autonomous systems:

- Open system architecture.
- LVC testing.
- STPA for autonomy.
- Runtime assurance.
- HAT performance.

4.2.3 Personnel Challenges

A major challenge in adapting to a new technology will often be the adjustments in personnel education, training, workforce composition, and organizational roles needed to efficiently and effectively develop and implement the new technology capabilities.

Personnel Challenges for T&E of Autonomous Systems

The challenges of personnel for the T&E of autonomous systems generally include four basic issues:

- DoD T&E organization personnel who have training and experience in T&E but do not have training or experience in new autonomy technologies and capabilities.
- Personnel trained and experienced in autonomy, especially involving software and AI, who do not have training or experience in DoD T&E.
- Personnel from other specialties who have experience in neither T&E nor autonomy but need to understand both T&E and autonomy to effectively support or utilize autonomous systems T&E.
- Organizational and personnel roles for the T&E of autonomous systems that are not optimized for the efficiency and effectiveness of the test program.

Personnel Challenge Details

The personnel challenges of autonomous systems T&E are many and varied.

Underlying factors. As an emerging technology field, autonomous systems, especially those using AI components, have few historical predecessors that require similar amounts of T&E, hardware, software, human interface, and operational expertise to integrate seamlessly together for a robust, efficient, and effective evaluation of system performance and trustworthiness.

Related risks. The failure to address personnel challenges may result in:

- Longer and more costly test programs due to test events that do not evaluate all relevant autonomous system features comprehensively and efficiently.
- Reduced capabilities due to the lack of test data evaluating comprehensive integration of hardware, software, cognitive, human interface, and mission qualities resulting in missed design, development, or integration deficiencies.
- Delayed or reduced understanding of autonomous system capabilities due to miscommunications or misunderstandings between key personnel specialties.

- Shifting the risks of autonomous system trustworthiness to the end users due to the lack of comprehensive, robust, and timely system T&E.

Affected individuals. Personnel challenges for autonomous systems T&E will affect all who need to have more than a traditional understanding of autonomous system hardware, software, interface, and operational features to succeed in their roles:

- Testers, who misunderstand the factors, measures, and scenarios of autonomy.
- Software and AI developers, who misinterpret the complex needs of DoD testing.
- Program managers, who misallocate or ineffectively utilize test resources.
- System developers and engineers, who fail to design efficiently testable systems.
- Commanders and operators, who misapply autonomy to operational concepts.

Trade-offs, limitations, or assumptions. Personnel issues are more challenging when:

- Software performance is highly dependent on hardware nuances, and vice versa.
- System effectiveness depends greatly on human interface effectiveness.
- System effectiveness depends on mission CONOPS and employment tactics.
- T&E success depends greatly on the specifics of integration, interfaces, or CONOPS.
- Autonomous system capabilities, interactions, or missions change rapidly.

Methods and Practices

The methods and best practices listed below and described in Section 5 can help address the challenges of personnel for the T&E of autonomous systems:

- Open system architecture.
- Continuous testing.
- Assurance cases.
- LVC testing.
- Surrogate platforms.
- AI model testing.
- Post-acceptance testing.
- Cognitive instrumentation.

- Runtime assurance.
- Task-based certification.
- HAT performance.

4.2.4 Exploitable Vulnerabilities Challenges

The use of CPS has become widespread in commercial and personal applications, and the use of AI in systems is expanding. The growing use of networked and AI-enabled systems by DoD, although offering benefits, introduces new vulnerabilities for adversarial attack or exploitation. Most autonomous DoD systems will have net-enabled or AI-enabled vulnerabilities.

Vulnerability Challenges for T&E of Autonomous Systems

The challenges of exploitable vulnerabilities for the T&E of autonomous systems generally include these basic issues:

- Autonomous systems often rely on datalink information, which can be disrupted, denied, or deceived, causing severe problems and demanding robust T&E to uncover.
- ML components, often used in autonomous systems, can be easily disrupted or deceived by small but clever adversarial changes in their inputs.
- Autonomous systems may often operate with little or no human oversight, which means that a traditional human operator "sanity check" on system information may be absent.
- Self-diagnosis and mitigation for these vulnerabilities add to the responsibilities of T&E processes, and few proven T&E techniques for these challenges currently exist.

Exploitable Vulnerabilities Challenge Details

The exploitable vulnerabilities challenges of autonomous systems T&E are many and varied.

Underlying factors. Autonomous systems rely heavily on data from ML training, sensors, datalinks, and software updates. This data reliance creates data vulnerabilities that adversaries or rogue actors can degrade, deceive, exploit, or otherwise disrupt.

Related risks. The failure to address exploitable vulnerabilities challenges may result in:

- Ineffective autonomous systems in operational or realistic test conditions, despite apparent effectiveness in early and highly controlled subsystem testing.
- Overreliance on the availability, completeness, accuracy, and security of data sources that provide the information needed for autonomous system effectiveness.

- Misconceptions about the level and details of human oversight needed for the autonomous system.
- Loss of national security information or critical technology advantages due to unnoticed or untested flaws in autonomous DoD systems.

Affected individuals. Exploitable vulnerabilities challenges for autonomous systems T&E will affect those who provide and who employ autonomous system data:

- Testers, who mischaracterize system trustworthiness without comprehensive vulnerability evaluations.
- Software and AI developers, who fail to mitigate data insecurity or disruption.
- Program managers, who overlook the risks of exploitable vulnerabilities.
- System developers and engineers, who over-rely on questionable data sources.
- Commanders and operators, whose missions may fail when threats expose unmitigated vulnerabilities.

Trade-offs, limitations, or assumptions. Exploitable vulnerabilities issues are more challenging when:

- System effectiveness is highly dependent on datalink availability and accuracy.
- Systems use ML components trained on only limited or synthetic data.
- System safety and security redundancies and mitigation processes are few.
- Autonomous system operations lack human monitoring and oversight.

Methods and Practices

The methods and best practices listed below and described in Section 5 can help address the challenges of vulnerabilities for the T&E of autonomous systems:

- Continuous testing.
- Code isolation.
- Assurance cases.
- LVC testing.
- Formal verification methods.
- AI model testing.

- STPA for autonomy.
- Adversarial testing.
- Post-acceptance testing.
- Cognitive instrumentation.
- Runtime assurance.
- Operational and mission-based testing.
- Quantified risk and performance growth curves.

4.2.5 Safety Challenges

Proving that something exists is far easier than proving that something does not exist. Safety is often an extremely difficult challenge because proving safety means proving that hazards do not exist. This challenge becomes even greater with complex autonomous systems where an independent layer of safety mitigation, namely a human operator, is removed.

Safety Challenges for Autonomous Systems

The challenges of safety for autonomous systems involve these basic issues:

- Autonomous systems are empowered to take real physical action, independently and without human control, and even without human user understanding, which shifts tremendous safety responsibility from the user to the designer, developer, and tester.
- Human operators spend many years gaining training and experience to avoid and mitigate risks in systems they employ. Transferring all of that safety culture and knowledge into autonomous system design and development is a massive, complicated endeavor.
- Autonomous systems T&E may often need to occur before all system safety issues and risks are fully understood, which adds new burdens to ensuring test safety.

Safety Challenge Details

The challenge of safety for autonomous systems was identified in the 2022 Advancements in T&E of Autonomous Systems Workshop Report as the most critical challenge, based on workshop surveys.

Underlying factors. Even with highly trained operators, DoD loses millions of dollars of assets and kills multiple people every year in accidents of DoD systems. Ensuring the safety of

autonomous systems is the first and most important step in accepting the systems' trustworthiness.

Related risks. The failure of T&E to address safety challenges may result in:

- Autonomous systems that kill or injure DoD Service members, allies, or bystanders.
- Autonomous systems that destroy or degrade DoD, national, or allied assets or otherwise cause unintended damage to property.
- Lack of acceptance of needed autonomous capabilities due to prominent unsafe system operations or unsafe tests causing a loss of confidence in autonomous systems' trustworthiness.

Affected individuals. Safety challenges affect those who have contact with autonomous systems, including testers, maintainers, and operators.

Trade-offs, limitations, or assumptions. Safety issues are more challenging when:

- Decisions underlying routine safety are taken out of the hands and minds of operators, who may not have a comprehensive understanding of autonomous systems' behavior or how to control the systems.
- Autonomous system operations or tests are in the vicinity of personnel, bystanders, or valued assets and property.
- Human oversight of the autonomous system is remote, nontechnical, aggregated, inconsistent, or time-lagged.
- Complex software, such as that used in autonomous systems, is designed and coded without transparency, making robust software evaluations very difficult.
- Open-source software or commercial off-the-shelf software is used to save time and costs, despite limited evidence of its safety-significant functions and robustness.
- A "fly-fix-fly" approach is used in development and testing, where safety failures, rather than being proactively addressed, are fixed only after they occur.
- A "fail fast" culture of impatience in development results in autonomous systems with fragile solutions that fail in safety-critical ways.
- Safety monitors and solutions are added to the autonomous system after initial development rather than being integrated from the start.

Methods and Practices

The methods and best practices listed below and described in Section 5 can help address the challenge of safety for the T&E of autonomous systems:

- STAT for autonomous systems.
- Operational modeling.
- Open system architecture.
- Autonomy requirements and specifications.
- Code isolation.
- Assurance cases.
- LVC testing.
- Surrogate platforms.
- Formal verification methods.
- STPA for autonomy.
- Runtime assurance.
- HAT performance.
- Automatic domain randomization.
- Automated outlier search and boundary testing.
- Failure path testing.
- Task-based certification.
- Operational and mission-based testing.
- Quantified risk and performance growth curves.

4.2.6 Ethics Challenges

Autonomous DoD systems will generally have some form of AI empowered to perform tasks traditionally carried out by human operators. One concern is the idea that human operators are accountable for their actions and are taught and trained to act responsibly in accordance with human laws and moral codes. To address this concern, DoD published five AI Ethical Principles, as outlined in the May 26, 2021, Deputy Secretary of Defense Memorandum, which apply to autonomous systems using AI components.

Ethics Challenges for Autonomous Systems

The challenges of ethics for autonomous systems involve these basic issues:

- Ethical actions for autonomous systems are not explicitly detailed in requirements and specifications in ways that can be empirically tested to evaluate ethical system behavior.
- Ethical principles for autonomous systems can be vague and subject to different interpretations by various agencies and stakeholders with differing perspectives.
- The five DoD AI Ethical Principles—Responsible, Equitable, Traceable, Reliable, Governable—have no clear and established standards for implementation, testing, or evaluation in autonomous systems. Additionally, these principles were not intended to address all ethical challenges posed by these systems and require further supplementation.

Example. An autonomous system is in high demand for an upcoming combat operation, but some expected operational conditions remain untested, and several instances of unintended system behavior have already occurred. However, the immediate deployment of the system has the potential to save lives. Can the system be ethically employed? It may be unclear how reliable and governable the system is and what the responsible decision should be.

Ethics Challenge Details

The challenges of ethics for autonomous systems create several difficulties for T&E.

Underlying factors. Autonomous systems in DoD can have the potential to act in ways that risk damage or loss of lives, assets, and trust. Their emerging acquisition and use introduce uncertainties regarding who holds accountability for their ethical behavior. Additionally, ethics is a widely studied field with many different theories and branches, which can lead to ethical dilemmas and conflicting objectives.

Related risks. The failure of T&E to address ethics challenges may result in:

- Unreliable systems with uncertain or inadequate safety, security, effectiveness, or other trustworthiness causing ethical violations of reliability expectations.
- Systems with unclear methodologies, data sources, design, or documentation causing the misunderstanding of appropriate technology use, whether due to vendor secrecy or excessive proprietary protections.
- Inequitable systems with biased actions that unjustly discriminate based on characteristics of people such as race, color, religion, age, and sex.

- Ungovernable systems that cannot detect or avoid unintended consequences or, upon demonstrating unintended behavior, cannot be disengaged or deactivated.
- Irresponsible use of autonomous systems due to inadequate or misinformed evaluation or reporting, causing misunderstanding of the systems' trustworthiness.

Affected individuals. Ethics challenges will affect all who design, integrate, manage, evaluate, and use autonomous systems, especially:

- Testers who fail to evaluate, analyze, and report the full span of operational conditions and scenarios where system effectiveness and trustworthiness are adequate or inadequate.
- Program managers, who misunderstand risks and trade-offs with ethical impacts based on mistaken trustworthiness or misplaced accountability.
- Commanders and operators, who employ autonomous systems in unethical ways because of deficient design, implementation, employment, or evaluation of systems.

Trade-offs, limitations, or assumptions. Ethics issues are more challenging when:

- The autonomous system may encounter scenarios where a single best ethical action is not clear because of conflicting priorities and objectives.
- Unintended behaviors may be encountered in complex situations with many complicating factors of the environment, scenario, or other circumstances.

Methods and Practices

The methods and best practices listed below and described in Section 5 can help address the challenges of ethics for the T&E of autonomous systems:

- STAT for autonomous systems.
- Assurance cases.
- LVC testing.
- STPA for autonomy.
- Post-acceptance testing.
- Cognitive instrumentation.
- Runtime assurance.
- HAT performance.

- Task-based certification.

4.2.7 Data Challenges

The use of data in the T&E of autonomous military systems, especially those incorporating AI and ML, presents significant challenges. These challenges include managing vast amounts of test, simulation, and operational data; ensuring data quality and accuracy; integrating diverse data types from multiple subsystems; and managing data without established data standards.

Data Challenges for T&E of Autonomous Systems

Data challenges for the T&E of autonomous systems generally include many issues and difficult tasks:

- Managing and storing vast amounts of data required for robust training, testing, and validation of ML component models.
- Ensuring data coverage, quality, and accuracy.
- Providing accurate annotation and labeling of data for autonomous systems, which requires domain-specific expertise.
- Using manual data labeling, which is often inadequate and time-consuming.
- Ensuring data security and privacy due to the sensitive nature of military operations.
- Integrating diverse data types from multiple subsystems.
- Handling interoperability issues arising from varying data formats and standards.
- Using real-time data processing for immediate decision-making during live tests and experiments.
- Utilizing historical data and establishing robust data governance policies.
- Addressing data access problems, such as balancing effective data sharing and collaboration among stakeholders with competing security or classification concerns.
- Recognizing a mismatch between the data available and the actual task.
- Realizing that data on the realistic behavior of U.S. adversaries in future conflicts is not only unavailable but that adversary nations actively conceal data and mislead U.S. observers on their plans, capabilities, tactics, etc.; intelligence data are inherently uncertain.
- Analyzing test data from complex operational scenarios without an obvious optimal solution or without established correct solutions.

- Establishing, characterizing, or validating the credibility of synthetically generated data.
- Anticipating that industry vendors may claim that key data are proprietary and refuse to share important data with independent government evaluators.
- Understanding that industry vendors may use unique data labeling, data formats, data organization, and data management processes due to a lack of contracted or mandated government data standards.

Data Challenge Details

The data challenges of autonomous systems T&E are relatively new to DoD T&E processes.

Underlying factors. Autonomous systems require large volumes of data to train, validate, and test ML models. Realistic and diverse datasets are essential to ensure that these systems perform well in real-world scenarios. Simulated environments, however, often fail to capture the complexity of real-world warfare scenarios, leading to a lack of realistic data. Additionally, data on adversary systems may be limited and may not reflect the full range of input factors across all operating conditions. As new operational situations arise, datasets need to be updated. The absence of data standards also creates friction in test data management.

Related risks. The failure to address data challenges may result in:

- Autonomous systems that are deficient or fail because of weak or brittle ML models, resulting from defective training datasets.
- Systems that provide skewed or faulty results derived from ML trained on biased or non-evaluated data.
- Poor performance in diverse situations, possibly in dramatic or unexpected ways, due to overtraining or overtuning to one set of data.
- Ineffective autonomous systems due to misconceptions about adversary capabilities derived from reliance on accurate adversary data, which is unavailable.
- Unknown ML component performance and robustness due to limitations in evaluating inaccessible data.
- Inaccurate or uncertain test data characterization and conclusions due to missing, unavailable, unused, mislabeled, or confusing data and management processes.

Affected individuals. Data challenges will affect all who design, integrate, manage, evaluate, and rely on autonomous systems:

- Testers, who must ensure that the datasets used for training and validation are representative of the entire operating environment.
- System developers and integrators, who need to manage, process, and store large volumes of data while ensuring data quality and security.
- Operators and commanders, who rely on accurate and reliable autonomous systems that have been thoroughly tested and validated to perform in diverse operating environments.

Trade-offs, limitations, or assumptions. Data issues are more challenging when:

- Data inputs become more complex and diverse because it is harder to ensure data quality and its accurate representation of the operating environment.
- Advanced computing, storage, and analytics capabilities are needed for real-time data processing during live tests and experiments.
- Data security and privacy, especially for classified data, are paramount because of the sensitive nature of military operations.
- The system relies on interoperability with joint force data using varying data formats and standards.
- Correct annotation and labeling of data requires domain-specific expertise.
- Independent government testers cannot access, or duplicate key data held by vendors.
- Vendors invent their own data labels, processes, or tools and are unclear with independent testers about how they work.

Methods and Practices

The methods and best practices listed below and described in Section 5 can help address the many challenges of data for autonomous systems T&E:

- STAT for autonomous systems.
- Open system architecture.
- Continuous testing.
- Assurance cases.
- Experimentation T&E.
- Cognitive instrumentation.
- Test user interface.

- Automatic domain randomization.

4.2.8 Human-Autonomy Teaming Challenges

HAT presents a critical challenge for the development, evaluation, and fielding of autonomous systems. T&E of HAT involves more than verifying and validating that a system is safe, reliable, and usable for human operators. It requires a more comprehensive evaluation than questionnaire-based surveys of subjective user feedback. Evaluating teaming between humans and autonomous systems requires an understanding of team structures, processes, and interdependencies in diverse social, mission, and environmental contexts. The integration of human and autonomy agents in a system and SoS environment poses significant T&E challenges, particularly when it comes to assessing the human element of performance contribution to the mission.

HAT Challenges for Autonomous Systems T&E

The challenges of HAT for autonomous systems T&E involve these basic issues:

- Autonomy capabilities are often developed without robust internal logic for sharing authority, responsibility, and accountability with human teammates.
- The CONOPS for autonomous systems mission execution, including robust HAT, is difficult to envision and mature without an autonomy legacy to learn from (e.g., existing autonomy operators, historic autonomy battles).
- The complexity of human-autonomy interactions is daunting because of the need to consider multiple variables, such as human decision-making, autonomy agent performance, and the dynamic exchange of information between them.
- Autonomy agents can behave differently in various scenarios, making it difficult to predict and evaluate their performance. This variability can lead to inconsistent or unforeseen human-autonomy interactions, further complicating the evaluation process.
- SoS interdependencies cause problems that are difficult to uncover because individual component performance can affect both human and system behaviors. Evaluating the impact of HAT on mission outcomes requires considering these interdependencies and the potential cascading effects of errors or failures.
- Human factors solutions for HAT, such as oversight, teammate interfaces, and even documentation, are immature for DoD autonomy applications and difficult to leverage from other commercial domains.
- Evaluating human factors and their impact on mission outcomes is essential but challenging because a human operator's cognitive load with autonomous systems can

dramatically change at the system and SoS levels, potentially leading to changes in decision-making, situational awareness, and overall performance.

- Autonomous systems involve many levels and types of risk that affect humans from many different perspectives and with different roles.
- Current metrics and assessment strategies rely on single-use test cases that do not account for combinations of humans and machines to perform as partners or determine how to measure team effectiveness.
- Reliable human-performance measures are necessary for evaluating and contrasting potential autonomy solutions, but there are often limited autonomy-development resources for measuring and designing for human performance. Traditional measures, such as reaction time or accuracy, may not be sufficient to capture the nuances of human performance in a complex, dynamic SoS environment. Scaling these measures for SoS evaluation is achievable but time-consuming and resource extensive.
- T&E interfaces and instrumentation with autonomous systems are often lacking comprehensive clarity into system status, priorities, actions, and control, which constrains the evaluation of how competing designs trade off human, autonomy, and HAT performance.

HAT Challenge Details

The challenges of HAT for autonomous systems present many issues.

Underlying factors. Traditionally, users of DoD systems have strategized, planned, executed, and created lessons learned for future improvement in operations by coordinating and collaborating as human operators with expert knowledge of their systems. This paradigm is destroyed by the introduction of autonomous systems without traditional operators—that is, systems that cannot coordinate and collaborate for strategy, tactics, and improvements on a mission-by-mission basis; these autonomous systems are unable to communicate with, understand, or explain detailed concepts to other operators. Until these advanced capabilities emerge, current and near-future human operators will be teaming and accumulating expertise with less complex, but still highly novel, autonomous teammates. These first-generation HATs may eventually remake operator communities, lessons learned, and improvement strategies that will drive future autonomy development. For the near term, however, many risks related to HAT may jeopardize autonomous systems' mission success.

Related risks. The failure of T&E to address HAT challenges may result in:

- Misalignment between human expectations for the autonomous system and its actual capabilities, degrading trust calibration and potentially leading to deficient behaviors and downstream system failures.
- Ineffective communication, coordination, or cooperation between autonomous systems and the humans or systems that interact with them.
- Improperly prioritized or miscommunicated system data that focuses only on the autonomous system's data needs, neglecting the human user's or teammate's needs, which can increase operator stress and frustration and potentially overwhelm a person in critical situations.
- Unoptimized control over the autonomous system, including excessive or insufficient autonomy or human involvement, which can lead to issues with safety; security; efficiency; or misaligned responsibility, authority, and accountability.
- Lack of situation perception by the system regarding its own current or future actions; a failure by the autonomous system to sense or anticipate its human teammates' actions and intentions; or a lack of awareness by the humans regarding their own actions in the context of the HAT.
- Dangers to bystanders or other non-teammates of the system.
- Failure to understand the system's degradation and failure modes, as well as the human's failure modes, leading to incorrect task assignments; deficient system behaviors; or overwhelming the human with monitoring, control, or other tasks that exceed their workload capacity.
- T&E events that may fail to effectively or efficiently characterize the autonomous system's trustworthiness and performance in mission-based scenarios.

Affected individuals. HAT challenges affect those involved in autonomous systems performance and trustworthiness, including testers, program managers, researchers, developers, engineers, maintainers, commanders, and operators.

Trade-offs, limitations, or assumptions. HAT issues are more challenging when:

- The autonomous system needs to coordinate or cooperate with humans to achieve mission objectives.
- Autonomy needs to team with more than one human simultaneously.
- Autonomy needs to team with humans to take effective proactive or responsive actions, where human response or engagement timing is important.

- Autonomy needs to explain behaviors or reasoning to humans in real time.
- Autonomy needs to fuse hardware inputs with real-time human inputs for success.
- Autonomous systems' projections of likely future actions must be communicated or coordinated with human teammates.
- HAT needs change dynamically during operations.
- The CONOPS is not clear in early development, or human teaming involvement in early development is insufficient.
- Legacy user interfaces are insufficient for HAT.
- The autonomous system will interact with untrained individuals who are unaware of the system's features or capabilities.

Methods and Practices

The methods and best practices listed below and described in Section 5 can help address the challenges of HAT for the T&E of autonomous systems:

- STAT for autonomous systems.
- Operational modeling.
- Open system architecture.
- Autonomy requirements and specifications.
- Assurance cases.
- LVC testing.
- Experimentation T&E.
- Surrogate platforms.
- STPA for autonomy.
- Cognitive instrumentation.
- Runtime assurance.
- Test user interface.
- HAT performance.
- Human performance standards.
- Task-based certification.

- Operational and mission-based testing.
- Quantified risk and performance growth curves.

4.2.9 Black Box Components Challenges

ML introduces new challenges for autonomous systems, particularly when ML components function as black boxes, obscuring the reasoning behind their outputs.

Black Box Challenges for T&E of Autonomous Systems

The challenges of black box components for the T&E of autonomous systems generally include several issues:

- Direct evaluation of the ML internal algorithm can be impossible because the exact patterns that the ML model uses depend on the training data, not just the design objectives and software code.
- ML performance is not guaranteed because ML training datasets are finite and the future is uncertain, especially in adversarial applications.
- Large, operationally representative data samples, needed to train ML effectively, are difficult to obtain, especially for foreign adversary combat applications.
- Large, complex software algorithms can also be nearly impossible to analyze in a comprehensive and robust way, even if they do not utilize ML.

Example. In an autonomous vehicle, a computer vision component with black box ML might have unclear reasons why it can or cannot recognize a "stop sign," potentially causing the vehicle to trigger a "stop" command anytime it perceives a certain shade of red, regardless of shape or orientation, or some other common but insufficient features.

Black Box Challenge Details

The black box challenges of autonomous systems T&E are largely new to DoD T&E processes.

Underlying factors. ML components are not explicitly programmed by the designers; instead, they rely on well-designed algorithmic models and data model selection and tuning by the ML engineers, along with large, highly representative training datasets for the application of interest. Consequently, uncertainties arise regarding the data's representativeness and how well the model was selected, both of which can be measured statistically only by extensive testing. The understanding of other complex software may pose different but equally intractable problems.

Related risks. The failure to address black box challenges may result in:

- Unknown autonomous system performance in untested situations.
- Poor characterization of the operational envelope and edge cases.
- Reduced ability to appropriately generate test cases from test sampling methods.
- Surprising failures or deficiencies based on bugs in untested software.
- Overconfidence in performance due to the overfitting of ML components
- Unreliable performance when training data are limited, inadequate, or missing.
- A mismatch of ML model capability with the actual system task.
- Outputs from ML models that cannot be easily interpreted.
- Susceptibility of ML models to manipulation through adversarial data inputs.
- Emergent, unpredictable system behavior when new situations occur.

Affected individuals. Black box challenges will affect all who design, integrate, manage, evaluate, and rely on autonomous systems:

- Testers, who may mischaracterize system performance because of software test limitations.
- Software and AI teams, who inadequately design and integrate software components.
- Program managers, who misunderstand component risks and limitations.
- System developers, who apply inadequate software integration and safeguards.
- Commanders and operators, who over- or under-rely on software trustworthiness.

Trade-offs, limitations, or assumptions. Black box issues are more challenging when:

- The task to which the software component applies is highly complex and dynamic.
- The software task success depends greatly on the effectiveness of human interfaces.
- The ML data inputs are highly variable and complicated.
- The ML training dataset is not large enough or fully operationally representative across all input factors, scenarios, and other characteristics.
- The autonomous system is highly reliant on the software component for effectiveness.
- Software component inputs, scenarios, tasks, or desired outputs change rapidly.

Methods and Practices

The methods and best practices listed below and described in Section 5 can help address the black box challenges for the T&E of autonomous systems:

- Open system architecture.
- Continuous testing.
- Assurance cases.
- LVC testing.
- AI model testing.
- Adversarial testing.
- Post-acceptance testing.
- Cognitive instrumentation.
- Runtime assurance.
- Automated outlier search and boundary testing.

4.2.10 Mission Evolution Challenges

At times, the operational capability needs for a DoD system change during the development and acquisition process, making T&E based on original requirements and specifications somewhat obsolete. Additionally, human operators of DoD systems can often adapt to employ their systems in new, more efficient, and more effective ways than originally planned by using new CONOPS or tactics. For example, the GBU-12 laser-guided bomb was originally designed and built as a precision munition for destroying static targets such as buildings. However, as operational situations evolved and operators needed ways to target ground-moving vehicles, they innovated by employing the GBU-12 against these moving targets and created new techniques to do so. So too, autonomous DoD systems will likely need future CONOPS and tactics adaptations that change the ways they are employed. The challenge of proving the potential and the success of autonomous systems' adaptations, however, will likely fall on the T&E practitioners because these systems will not continuously have human operators.

Mission Evolution Challenges for Autonomous Systems

The challenges of mission evolution for autonomous systems involve these basic issues:

- Mission capability needs evolve over time based on changes in adversary threat systems, adversary tactics, friendly systems or tactics, and other operational priorities.

- Autonomous systems may be the only viable solutions to meet changing challenges, which may cause the objectives, conditions, and scenarios of autonomous systems T&E to evolve as well.

Example. An autonomous aircraft that relies on the Global Positioning System (GPS) for navigation, originally designed for surveillance in a low-intensity conflict, might need to provide surveillance in a high-intensity, GPS-denied conflict. Similarly, an autonomous ground vehicle designed for use on U.S. roadways with full pavement markings may be needed for missions in the United Kingdom, where vehicles drive on the opposite side of the road, or might be needed to drive on unmarked roadways.

Mission Evolution Challenge Details

The challenges of mission evolution for autonomous systems create several difficulties for T&E.

Underlying factors. Warfighters have often adapted their systems and tactics to meet unexpected challenges on the battlefield, making the best use of the assets they are given, regardless of their original intended design purpose.

Related risks. The failure of T&E to address mission evolution challenges may result in:

- Deficient or possibly exploitable autonomous systems in operational scenarios if autonomous systems are employed in mission-evolved ways incompatible with trustworthy effectiveness.
- Added costs and delays for effective mission-evolved capabilities if overly burdensome or outdated T&E processes are applied.

Affected individuals. Mission evolution challenges will affect all who design, integrate, manage, evaluate, and use autonomous systems, especially:

- Testers, who misstate the full span of operational conditions and scenarios where system effectiveness and trustworthiness are adequate and inadequate, or who must retest an autonomous system efficiently to characterize trustworthiness and performance in mission-evolved situations.
- Program managers, who misunderstand risks and trade-offs in improving, testing, and fielding autonomous systems with necessary mission-evolution changes.
- Commanders and operators, who employ autonomous systems in mission-evolved scenarios when the systems are not viable, or who fail to employ autonomous systems in evolved situations where the system would be effective.

Trade-offs, limitations, or assumptions. Mission evolution issues are more challenging when:

- The mission's operational envelope is not clearly defined, leading to a mismatch between what developers and users at various levels (component, subsystem, system, and SoS) expect from the system.
- Details about the conditions, configurations, scenarios, and other information for system testing are incomplete, causing misunderstandings about system suitability across various missions.

Methods and Practices

The methods and best practices listed below and described in Section 5 can help address the challenge of mission evolution for the T&E of autonomous systems:

- Assurance cases.
- LVC testing.
- Adversarial testing.
- Post-acceptance testing.
- Cognitive instrumentation.
- Runtime assurance.

4.2.11 Dynamic Learning Challenges

A long-term goal of systems using AI is the implementation of recursive self-improvement, referred to in this guidebook as "dynamic learning" (also known as online ML). Systems with this capability are sometimes called cognitive systems.

Dynamic Learning Challenges for Autonomous Systems

The challenges of dynamic learning for autonomous systems, though not yet widespread, may arise in some systems and involve key issues:

- Systems using dynamic learning can modify their algorithm while deployed, so system testing must evaluate their ability to adapt effectively during operation.
- System capabilities will change as the ML algorithm adapts, meaning that T&E results provide only a snapshot of system performance and trustworthiness.

Example. In an autonomous vehicle, a decision component with dynamic learning in ML might adapt itself to avoid brown surfaces after getting stuck on downed tree branches. This adaptation, which may be barely perceptible, would warrant evaluation by testers to determine whether the adaptation was appropriate, as well as to evaluate its mission impact on other brown-colored terrain.

Dynamic Learning Challenge Details

The use of dynamic learning in autonomous systems creates several difficulties for T&E.

Underlying factors. Emerging technology exists that allows certain types of ML to continuously "learn" or adapt to improve performance without human involvement or oversight. Trustworthy autonomous systems with dynamic learning capabilities could have significant advantages in a rapidly changing adversarial battlespace compared with static systems that behave predictably in response to enemy actions.

Related risks. The failure of T&E to address dynamic learning challenges may result in:

- Deficient or untrustworthy dynamic learning components being deployed in autonomous DoD systems.
- Deficient or possibly exploitable autonomous systems in operational scenarios if dynamic learning is not employed but could have been.
- Added costs and delays for effective dynamic learning capabilities if overly burdensome or outdated T&E processes are applied.
- Autonomous systems with dynamic learning capabilities that "drift" away from needed performance over time in some conditions.
- "Catastrophic forgetting" by dynamic learning capabilities that eventually replace nearly all their initial ML training because of online learning adaptations.

Affected individuals. Dynamic learning challenges will affect all who design, integrate, manage, evaluate, and use autonomous systems with dynamic learning components, including testers, software and AI teams, program managers, system developers, commanders, and operators.

Trade-offs, limitations, or assumptions. Dynamic learning issues are more challenging when:

- The task to which the dynamic learning component applies is complex and nuanced.
- The dynamic learning component data inputs are highly variable and complicated.

- The common or frequent employment situations for the autonomous system do not fully span the range of potential operational situations, causing the dynamic learning component to "forget" its training for rare but highly important situations.
- The autonomous system is highly reliant on the dynamic learning component for effectiveness.
- The autonomous system cannot easily be regression tested to determine the impact of "drift" in the dynamic learning component performance.

Methods and Practices

The methods and best practices listed below and described in Section 5 can help address the challenge of dynamic learning for the T&E of autonomous systems:

- Continuous testing.
- Assurance cases.
- LVC testing.
- AI model testing.
- Adversarial testing.
- Post-acceptance testing.
- Cognitive instrumentation.
- Runtime assurance.
- HAT performance.

4.2.12 Test Adequacy and Coverage Challenges

A critical question in the T&E of autonomous systems is "How will the system be tested to characterize its performance and trustworthiness?" This question implies two issues. The first is how to ensure *test quality*—adequate realism and fidelity at any individual test point. This issue is addressed differently depending on the autonomous system's design, CONOPS, and implementation details.

The second issue is how to optimize *test quantity*—adequate numbers of test points to provide a robust characterization of the system. Repeating the exact same test conditions over and over, while providing some data on the variability in the system, would not provide adequate characterization across all conditions and scenarios in which the system could operate. A more

useful characterization is gained by designing a test series that provides robust *coverage* of all potential conditions and scenarios the system may encounter.

Coverage refers to the spectrum of input conditions under which the system is understood. It is the sum of the input conditions and combinations that define the expected envelope of operation where the system is expected to perform. In M&S, these conditions are referred to as the state space. In real-world applications, it is known as the performance envelope. Test adequacy implies that the sampling strategy applied to the testing is comprehensive and thorough across these input conditions or, in other words, has adequate "test coverage."

The challenges of test adequacy and coverage are not unique to autonomous systems. The features of autonomous systems, however, cause test adequacy and coverage to be far more difficult compared with traditional systems with human operators.

Test Adequacy and Coverage Challenges for T&E of Autonomous Systems

Adequacy and coverage challenges for the T&E of autonomous systems generally include several issues:

- Black box components used in autonomous systems, such as ML components, may have no underlying physics model able to consistently predict outputs, thus requiring comprehensive system-level testing to characterize their performance.
- The quantity of factors (dimensionality of the inputs) for autonomous systems can be extremely large, for example:
    - Weather-related factors are numerous, including temperature, pressure, humidity, cloud coverage, precipitation, illumination, winds, currents, sun angles, and visibility.
    - Terrain-related factors can include vegetation, structures, elevation, obstacles, surface composition and hardness, snow and ice, colors, and absorption or cooling.
    - Background and clutter-related factors can include vehicles, pedestrians, wildlife, bystanders, intruders, nonparticipating friendlies, and spectrum-related traffic.
    - Threats and adversary potential actions may include a myriad of additional factors.
    - Teammates and friendly entities may vary from none to many, with diverse roles, communication, coordination, and cooperation needs or options.
- All the above may be combined in diverse scenarios with different quantities, distances, frequencies and timing that, in effect, are different system inputs.
- In total, the possible conditions and scenarios the autonomous system may encounter could easily be in the range of billions to trillions of combinations.

Note that "coverage" as used in this document is different from the software term "code coverage," which refers to a measure of the degree to which the source code of a software program is executed during a test sequence. Code coverage is an important metric for software components, but it does not necessarily give any indication of test adequacy and completeness for the entire autonomous system.

Test Adequacy and Coverage Challenge Details

The adequacy and coverage challenges of autonomous systems T&E include many important details.

Underlying factors. In traditional systems with a human operator, the human observes the conditions and scenarios, and the human adjusts the system to achieve objectives safely. In autonomous systems, the system must perform these tasks on its own, so T&E must evaluate these capabilities across varying conditions and scenarios.

Related risks. The failure to address adequacy and coverage challenges may result in:

- Poor autonomous system performance in diverse situations, possibly in dramatic or unexpected ways, due to evaluation on a limited set of conditions and scenarios.
- Overestimation of the system capabilities due to testing only in ideal conditions or baseline scenarios without complex realism.
- Delayed and more costly development, testing, and fixes due to inadequate insight into the root causes of deficiencies, based on piecemeal performance testing rather than on comprehensive evaluation that accurately characterizes trustworthiness.
- Inefficient use of resources due to poorly designed tests.

Affected individuals. Test adequacy and coverage challenges will affect all who design, integrate, manage, evaluate, and rely on autonomous systems:

- Testers, who must evaluate the autonomous system across the complete operating environment in a myriad of scenarios.
- System developers and integrators, who need to understand the many factors involved in system operations and evaluation and ensure that system development accounts for these various situations.
- Operators and commanders, who rely on accurate and reliable autonomous systems that have been thoroughly tested and validated to perform in diverse operating environments.

Trade-offs, limitations, or assumptions. Test adequacy and coverage issues are more challenging when:

- Testing multiple conditions and scenarios is costly and time-consuming.
- The autonomous system's performance is sensitive to small changes in conditions or inputs.
- The CONOPS was not developed early on, leading to system development without a realistic understanding of the conditions and scenarios of operations.
- All potentially relevant conditions and scenarios affecting system trustworthiness were not identified and planned for early in the life cycle.
- Input variables are identified at too high or too low of a degree or level.
- M&S of the full set of operational conditions and scenarios is unavailable or unrealistic in some respects or otherwise limited.
- STAT is not used in test planning and test design to optimize test coverage efficiently.

Methods and Practices

The methods and best practices listed below and described in Section 5 can help address the many challenges of test adequacy and coverage for autonomous systems T&E:

- STAT for autonomous systems.
- T&E for autonomy M&S.
- Continuous testing.
- Assurance cases.
- LVC testing.
- Experimentation T&E.
- Formal verification methods.
- STPA for autonomy.
- Post-acceptance testing.
- Runtime assurance.
- Automatic domain randomization.
- Automated outlier search and boundary testing.

- Failure path testing.
- Operational and mission-based testing.
- Quantified risk and performance growth curves.

4.2.13 Autonomy Integration and Interoperability Challenges

The integration of autonomous systems, including those powered by AI/ML, into larger systems presents significant challenges. This section explores these challenges, specifically focusing on the complexities of integrating AI/ML autonomy subsystems along with ensuring interoperability within the joint force. Interoperability is more than just information exchange; it is the ability to act together coherently, effectively, and efficiently to achieve tactical, operational, and strategic objectives. This challenge includes the integration of systems, processes, procedures, organizations, and missions in appropriately stressed operational environments over the system's life cycle.

Integration and Interoperability Challenges for Autonomous Systems

The challenges of integration and interoperability for autonomous systems involve these issues:

- Autonomous systems extend the scope of integration T&E by adding functions, which were traditionally performed by a human operator, that must be evaluated.
- Autonomous systems likewise extend the scope of interoperability T&E by requiring that compatibility and teaming be tested between independently developed autonomous and traditional systems.

Example. An autonomous vehicle may need to effectively integrate pre-mission intelligence on adversary threats with real-time sensor inputs from radar, infrared, visual, electronic, and datalink threat information to prioritize its mission objectives and manage mission risk. The same vehicle may need to seamlessly synchronize its actions with friendly teammates to divide and conquer team objectives or to simultaneously overwhelm adversary defenses.

Integration and Interoperability Challenge Details

The challenges of integration and interoperability for autonomous systems present many issues.

Underlying factors. Autonomous systems are a complex combination of multiple key subsystems that must interact correctly to operate effectively, and some of these subsystems involve very new technology, such as AI and ML. Autonomous systems also operate within an increasingly complex DoD operational battlespace where systems' capabilities must combine to form coherent SoS solutions.

Related risks. The failure of T&E to address integration and interoperability challenges may result in:

- Autonomous systems with operational deficiencies due to subsystems that appear to perform well separately but are ineffective or inconsistent when integrated together in a deployed system environment.
- Autonomous systems with mission limitations or drawbacks due to deficient interoperability or poor synchronization with other autonomous or traditional systems.
- Subsystems of autonomous systems that require time-consuming and costly redevelopment or major improvements when they fail to integrate as envisioned.
- Inadequate HAT (discussed further in Section 4.2.8).

Affected individuals. Integration and interoperability challenges affect those who develop, integrate, evaluate, and rely on autonomous systems:

- Testers and evaluators, who must ensure that autonomous systems function correctly within the larger system and joint operating environment.
- System integrators and developers, who need to design and implement subsystems that can integrate seamlessly and operate reliably within broader mission packages.
- Operators and commanders, who depend on the autonomous systems to perform as expected in various operational scenarios.

Trade-offs, limitations, or assumptions. Integration and interoperability issues are more challenging when:

- The autonomous system relies upon highly variable and complex data and sensor inputs.
- The autonomous system needs complicated HMT.
- The system uses multiple indirect inputs from datalinks or other sources that can have significant latency or other update and timing issues.
- The autonomous system frequently changes tasks, inputs, or outputs, dynamically affecting how its subsystems and teammates must respond.

Methods and Practices

The methods and best practices listed below and described in Section 5 can help address the challenge of integration and interoperability for the T&E of autonomous systems:

- STAT for autonomous systems.
- Operational modeling.
- Open system architecture.
- Autonomy requirements and specifications.
- Assurance cases.
- LVC testing.
- Experimentation T&E.
- STPA for autonomy.
- Runtime assurance.
- Test user interface.
- HAT performance.
- Operational and mission-based testing.
- Quantified risk and performance growth curves.

4.3 Mapping of Challenges to Methods and Best Practices

Table 4-1 cross-references each autonomous systems T&E challenge with each method or best practice that helps address the challenge.

Table 4-1. Matrix of Challenges vs. Methods

		Challenges														
		T&E as a Continuum	T&E of the OODA Loop	Requirements	Infrastructure	Personnel	Exploitable Vulnerabilities	Safety	Ethics	Data	Human-Autonomy Teaming	Black Box Components	Mission Evolution	Dynamic Learning	Test Adequacy and Coverage	Autonomy Integration and Interoperability
Methods	End-to-End Autonomy T&E Processes	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
	STAT for Autonomous Systems	x		x				x	x	x	x				x	x
	M&S for Autonomy T&E		x												x	
	Operational Modeling		x	x				x			x					x
	Small-Scale Development	x		x				x		x						
	Open System Architecture	x	x	x	x	x		x		x	x	x				x
	Autonomy Requirements and Specifications			x				x			x					x
	Continuous Testing	x	x			x	x			x		x		x	x	
	Code Isolation	x					x	x								
	Assurance Cases	x	x			x	x	x	x	x	x	x	x	x	x	x
	LVC Testing	x	x	x	x	x	x	x	x		x	x	x	x	x	x
	Experimentation T&E	x								x	x				x	x
	Surrogate Platforms	x	x			x		x			x					
	Formal Verification Methods			x			x	x							x	
	AI Model Testing	x	x			x	x					x		x		
	STPA for Autonomy			x	x		x	x	x		x				x	x
	Adversarial Testing		x				x					x	x	x		
	Post-Acceptance Testing	x	x			x	x		x			x	x	x	x	
	Cognitive Instrumentation	x	x			x	x		x	x	x	x	x	x		
	Runtime Assurance	x			x	x	x	x	x		x	x	x	x	x	x
	Test User Interface		x							x	x					x
	Human-Autonomy Team Performance	x	x	x	x	x		x	x		x			x		x
	Automatic Domain Randomization	x	x					x		x		x			x	
	Automated Outlier Search / Boundary Testing							x				x			x	
	Failure Path Testing							x							x	
	Human Performance Standards	x		x		x		x		x	x		x			x
	Task-Based Certification	x		x		x		x	x		x					
	Operational and Mission-Based Testing						x	x			x				x	x
	Quantified Risk / Performance Growth Curves	x					x	x			x				x	x

5 Methods and Best Practices

The previous sections of this guidebook discussed the policies, background, vision, and challenges for the T&E of autonomous systems. This section provides a collection of methods and best practices to address these issues and challenges. The methods presented are divided into two general groups: overarching methods and specific methods.

The overarching methods are three complementary methodologies that apply to all autonomous systems:

- End-to-end autonomy T&E process.
- STAT for autonomous systems.
- M&S for autonomy T&E.

Understanding these overarching methods is essential for achieving autonomous systems T&E that is effective, efficient, and robust. The information provided in the overarching methods sections should provide a foundational understanding for all stakeholders and offer practical frameworks for utilizing the specific methods discussed in subsequent sections.

The specific methods are more focused practices that address the challenges in autonomy T&E. Although not all methods apply universally, they have been categorized based on the phase of the T&E life cycle where they provide the greatest benefit. Many methods, however, provide benefits across multiple life cycle stages. The specific methods discussed in this section are organized according to the following T&E life cycle phases:

- Acquisition and development strategy.
- Test strategy.
- Test planning.
- Test execution.
- Data analysis and evaluation.

Many of the methods complement each other, and their combined application can enhance the overall effectiveness of autonomy T&E.

Each method is described to provide practitioners with a basic understanding, allowing them to determine its applicability to their project. If a method is deemed relevant, practitioners can explore further details for implementation using the tools and references provided.

The discussion of each method is organized to include the following information regarding the method's intended use, benefits, limitations, and other considerations.

- Method description.
- Details and best practices.
- Primary outcomes and additional benefits.
- Costs, limitations, and assumptions.
- Challenges addressed by the method.

As described earlier in this document, all lessons, methods, and tools discussed in the guidebook are intended as a best snapshot at the present time and act as a "living" documentation of currently used and available methods and tools for autonomous systems T&E. No claim is made that the methods provided in the guidebook are sufficient to guarantee success nor to assert that these represent all useful methods for autonomy T&E. Because autonomy is an emerging technology discipline, some methods and tools have limited information available; the intent is to expand and improve upon useful methods and tools as the capabilities for the T&E of autonomous systems mature.

5.1 Overarching Methods for Test and Evaluation of Autonomous Systems

The overarching methods are three complementary methodologies that apply to all autonomous systems:

- End-to-end autonomy T&E process.
- STAT for autonomous systems.
- M&S for autonomy T&E.

Understanding these overarching methods is essential for meeting the many challenges of T&E of autonomous systems.

5.1.1 End-to-End Autonomy Test and Evaluation Process

The first overarching method for autonomous systems T&E is a framework for applying many of the other methods in this section. This end-to-end autonomy T&E process provides a structured approach to assess the performance and trustworthiness of autonomous systems throughout their life cycle. Concise, basic information about the best practices, limitations, and challenges of the end-to-end autonomy T&E process are presented first, followed by a more detailed discussion of these concepts.

Description of the End-to-End Autonomy T&E Process

The end-to-end autonomy T&E process:

- Is a comprehensive and iterative approach that integrates evidence from diverse sources to build a unified assurance argument for the safe and effective operation of autonomous systems.
- Emphasizes a holistic approach, recognizing that no single T&E method can provide sufficient evidence to support a comprehensive assurance argument. The process integrates various methods, including model-based analysis, simulation, formal verification, and live testing, to assess the system's performance across different operational contexts.
- Is an iterative process, allowing for continuous refinement and improvement based on the evidence gathered from T&E activities, model-based analysis, simulation, and operational feedback.
- Is an adaptable approach, recognizing that the specific T&E methods and their level of emphasis may vary depending on the autonomous system's complexity and intended use.

Details and Best Practices

Key features of the end-to-end autonomy T&E process include:

- Multiple T&E methods. The process incorporates various methods, including model-based requirements analysis, simulation-based functional testing, context-independent testing, processor-in-the-loop (PIL) testing, formal analysis, VC operator-in-the-loop testing, LVC testing, live DT, OT, runtime assurance, and assurance aggregation.
- Iterative testing. The process emphasizes an iterative approach, where testing starts with small, focused experiments; uses sequential test designs; and gradually builds toward OT.
- Evidence aggregation. The process integrates evidence from different T&E methods into a unified assurance argument that supports certification and accreditation.
- Test scope expansion. The scope of testing expands throughout the process, starting with individual AI components and progressing to integrated assemblies and complete autonomous platforms.
- Early test team involvement. The test team is involved early in the acquisition process, including during requirements development, to ensure that requirements are complete, consistent, and testable.

Primary Outcomes and Additional Benefits

The primary outcomes and additional benefits of the end-to-end autonomy T&E process include the following:

- Primary outcomes:
  - Justified confidence in the system's safety and effectiveness. The comprehensive and iterative nature of the process provides a strong assurance argument for the system's performance.
  - Reduced risk of system failures. The early identification and mitigation of potential issues through testing help to reduce the risk of system failures in operation.
- Additional benefits:
  - Improved system performance.
  - Reduced development costs.
  - Enhanced communication and collaboration between stakeholders.
  - Increased user acceptance.
  - Faster time to fielding.

Costs, Limitations, and Assumptions

The use of the end-to-end autonomy T&E process may have the following negative impacts:

- Complexity. The process can be complex and requires significant expertise to implement effectively.
- Resource intensiveness. The process can be resource intensive, requiring significant time, personnel, and equipment, which may differ from traditional resources.
- Data management challenges. The process generates large amounts of data that must be effectively managed and analyzed.

Note: The benefits of this end-to-end framework are likely to greatly offset these costs by reducing later fixes, reworks, or failures.

Tools and Resources

For more information about end-to-end autonomous systems T&E processes, see the conference presentation, "A Holistic Look at Testing Autonomous Systems" (Scheidt 2016).

Challenges Addressed by This Method

The end-to-end autonomy T&E process helps to address several challenges for the T&E of autonomous systems including:

- **Adapting to T&E as a Continuum**. The process provides a structured approach to testing and evaluating complex systems from early experimentation and prototyping through employment and ongoing improvements.
- **T&E of the OODA Loop**. The process applies to dynamic autonomous systems and AI technology components to provide evaluation of all phases of the OODA loop.
- **Requirements**. The process links autonomy requirements through all phases of testing.
- **Infrastructure**. The process provides a framework to maximize the efficient use of infrastructure throughout the T&E process.
- **Safety**. The process uses an iterative, build-up approach to reduce test safety risk and mitigate system risk as early as possible and helps to reduce performance risks by providing a strong assurance argument for the system's safe and trustworthy performance.
- **Data**. The process emphasizes the collection and analysis of data from multiple sources.

Considerations and Details for the End-to-End Autonomy T&E Process

A variety of T&E methods can produce useful evidence of autonomy performance; however, no single autonomy T&E method can, by itself, produce sufficient evidence to support a comprehensive assurance argument. Test engineers should follow a test process that integrates evidence from different sources to create a unified assurance argument. Detailed recommendations for specific test methods are provided elsewhere in this guidebook. An overview of the T&E methods that could be included in an autonomy T&E process is summarized as follows:

- Model-based requirements analysis: Incorporates a parametric model of the autonomy's expected characteristics into a system model, which is used to evaluate the expected benefits of the yet undeveloped AI.
- Simulation-based functional testing: Enables automated testing of the autonomy in a constructive world. Using high-performance computing faster than real-time (FTRT) simulation in the loop can examine functional performance of the autonomy under test over trillions of scenarios.

- Context-independent testing: Involves automated source code examination to identify autonomy inputs that cause autonomy decisions that produce unacceptable outputs. Robustness Inside-Out Testing is one such tool that provides this capability.
- PIL testing: Uses a real-time simulation to test the autonomy running on the target computing infrastructure (CPU, RAM, network, etc.). PIL testing is used to validate the timeliness of autonomy decisions.
- Formal Analysis: Uses rigorous mathematical analysis of formal models of AI algorithms and/or autonomy design to define performance guarantees of the algorithm or design.
- VC operator-in-the-loop testing: Connects the real-time PIL test apparatus to operator simulators to examine human-autonomy interactions including operator confidence calibration.
- LVC testing: Examines an autonomous system's ability to operate in highly dynamic and challenging conditions by exposing the live platform to simulated targets or threats, providing a realistic test environment for system performance and operator decision-making.
- Live DT: Provides fully live testing on a closed track in which the autonomy stimuli are planned and controlled by the test team.
- OT: Evaluates an autonomous system's performance by deploying it within realistic wargame scenarios or training exercises, where it operates as intended in accordance with the CONEMP/CONOPS while interacting with human operators and other systems.
- Runtime assurance: Involves an onboard monitoring system responsible for monitoring autonomy performance during a mission and, if necessary, preventing the autonomous system from executing an unsafe action.
- Assurance aggregation: Combines evidence from other T&E methods into a unified assurance argument that can support certification and accreditation.

The scope of AI used in an autonomous system may vary greatly, from simple systems that utilize a single AI component, to platforms containing dozens of complex, interdependent AI subcomponents. Therefore, test engineers should expect to tailor the autonomy T&E process, selecting the test methods that are best suited to their specific autonomous SUT.

Each of the listed methods exhibits unique advantages and disadvantages that should be considered by the test team when developing a Test and Evaluation Master Plan (TEMP). Test tools and methods should be selected to manage the trade-off between completeness and accuracy. Those evaluation methods that are effective at examining large numbers of diverse circumstances can do so only because they make simplifying assumptions that sacrifice test accuracy, whereas the most accurate tests, such as OT, are expensive, time-consuming, and incapable of examining large numbers of circumstances.

The autonomy test process should not only include a variety of evaluation methods but also vary the scope of the autonomy under test throughout the process. It is recommended that T&E processes start small by focusing on the performance characterization of isolated AI components (e.g., target detection and the camera that feeds it data) and then move on to assemblies of integrated AI components (e.g., all perception modules required to produce platform situational awareness), before examining complete autonomous platforms or SoS that include teams of autonomous systems. With autonomous systems, however, integration testing is necessary but not sufficient to determine how components or systems will work together in an operational context or to detect all potential emergent behaviors. Comprehensive T&E must assess interactions across varying levels of integration to uncover unintended effects that may arise only in complex, real-world environments.

Autonomy test engineers should utilize a tailored T&E process that defines the test methods, scope, and scale of each stage to suit the unique needs of the autonomy under test.

The reason the test team should “start small” is because the number of possible autonomy–world interactions is unmanageably large, which effectively prohibits comprehensive system testing. It is recommended that the test team build an assurance argument by combining evidence from an integrated suite of test events that start with small, focused experiments and gradually build toward OT. As shown in Figure 5-1, test engineers should start by conducting large-batch experiments of key AI components contained within the autonomous system. Analysis of these focused results will identify AI performance limitations and the autonomous system’s performance envelope. Data from tests in which it is already known that the AI will fail, or easily succeed, produces little in the way of useful evidence. The test engineer should utilize large-batch analysis to cull the state space for successive tests, focusing on “maximum value” test conditions for more accurate, time-consuming, and expensive high-fidelity tests. Ideally, high-

fidelity system-level tests will confirm previous large-sample, low-fidelity results, and the test team can begin a new cycle of testing by increasing the testing scope by adding AI components and/or expanding the operational scope.

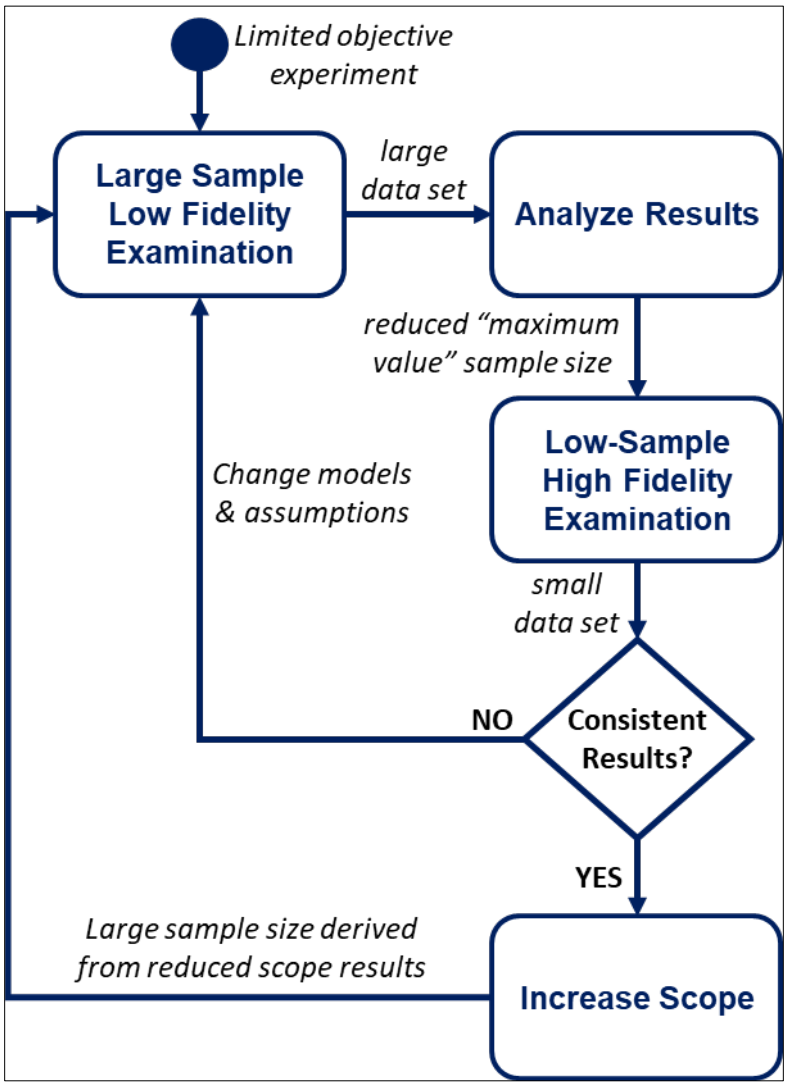


Figure 5-1. Iterative Test Process Concept

High-fidelity results not matching large-batch results usually indicate a flaw in the assumptions and/or models used in the large-batch testing. When high-fidelity results conflict, model flaws should be identified and repaired, and regression testing should be used to correct errors in the large-batch experiments. The test cycle shown in Figure 5-1 should be iteratively repeated, gradually adding scope and complexity of the autonomy being tested until the entire autonomous system is examined under sufficiently robust operational conditions and scenarios. The test conditions examined in each stage should be derived from prior cycles, continually optimizing

the evidence gained per test by focusing on conditions that maximize valuable information and minimize unknowns during each subsequent test (see STAT for Autonomous Systems (Section 5.1.2) and other sections below).

When compared with traditional system testing, the recommended process may appear to be overly complex and cumbersome; however, done correctly, by iteratively refining models and optimizing test conditions in earlier cycles, this process reduces unnecessary system-level tests by ensuring that only the most valuable, high-fidelity scenarios are selected for final evaluation. This targeted approach minimizes costs and maximizes insights while avoiding redundant full-system tests.

Figure 5-2 provides an overview of the complete end-to-end process.

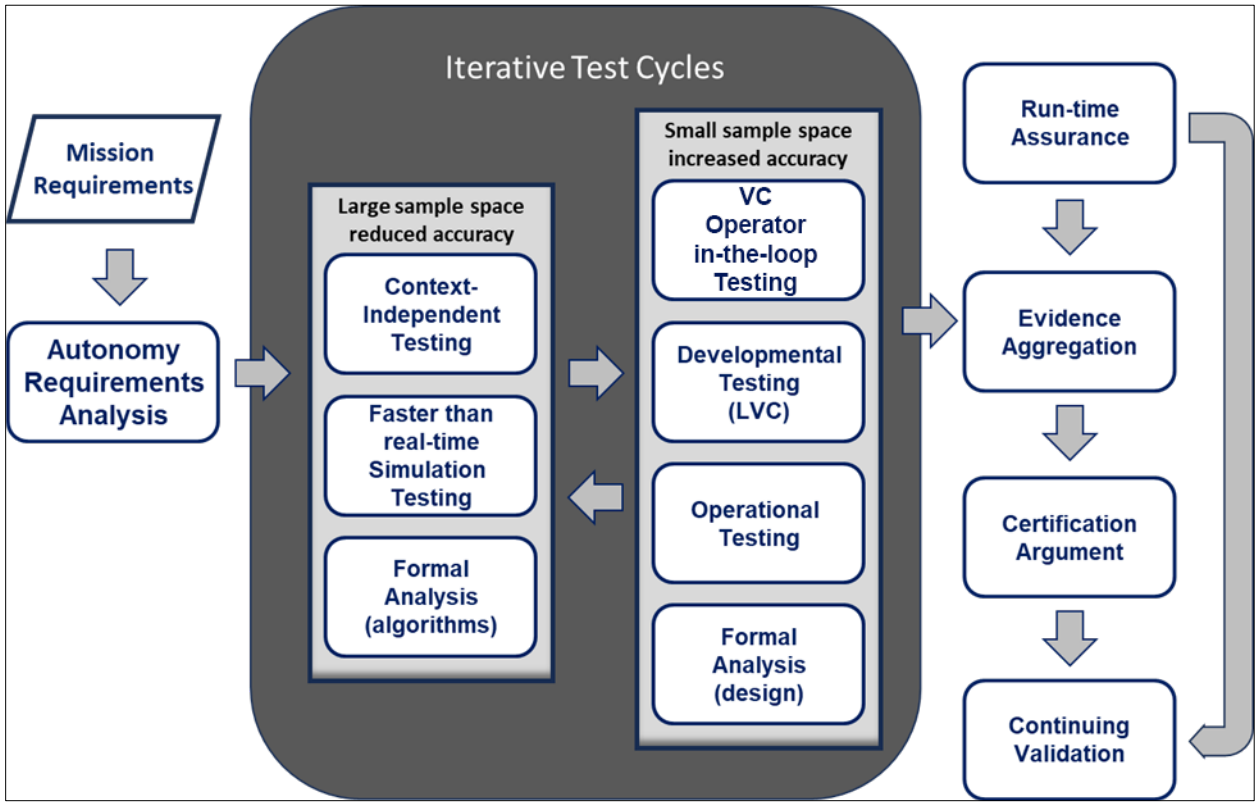


Figure 5-2. Overview of Complete End-to-End Process

One of the key findings of the DoD Autonomy Community of Interest T&E V&V Working Group in its Technology Investment Strategy 2015–2018 is the recognition that autonomy testing requires involvement from the autonomy test engineer throughout the engineering life cycle, starting with the requirements process and continuing through deployment and sustainment. The iterative test process described in Figure 5-1 should be contained within a larger, more linear process as shown in Figure 5-2.

Formal autonomy requirements should be derived from mission requirements early in the acquisition cycle. Autonomy requirements must define what decisions must be made by the platform to satisfy mission requirements; what knowledge is required to make those decisions; and the fitness criteria that can be used to evaluate the effectiveness of a decision and the scope of the operating conditions under which those decisions are to be made. Formal autonomy requirements will be used to define test parameters throughout the test process, so it is vital that the test team be available at inception to ensure that requirements are complete, consistent, and testable.

Figure 5-3 shows the Joint Digital Autonomy Range process.

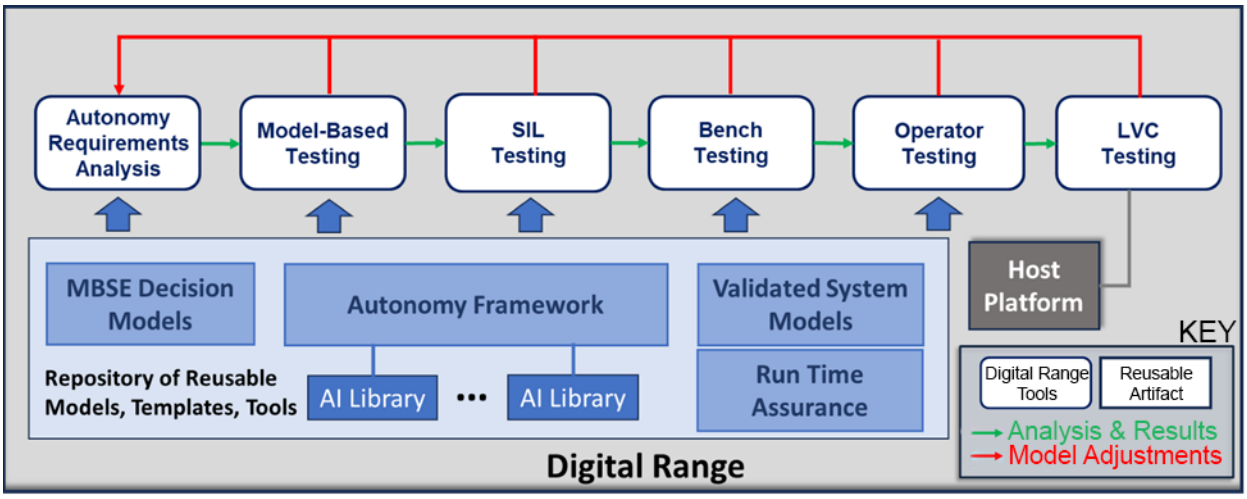


Figure 5-3. Joint Digital Autonomy Range Process

Iterative testing alternates between methods that provide evidence for large portions of the sample space and methods that provide evidence from accurate, small sample size tests. Large sample space methods should be used to develop evidence that is comprehensive, providing confidence that the autonomy under test will satisfy mission requirements under sufficiently robust circumstances. Large sample space methods include context-independent testing, FTRT simulation-based testing, and formal analysis of the underlying algorithms when possible.

Small sample space methods should be used to validate large sample space results, demonstrating that equivalent results are achieved when identical tests are run under conditions accurately representing the real-world. Small sample space methods include VC human-in-the-loop testing, LVC DT, formal *correct-by-construction* design analysis, and OT.

Another use of iterative testing is to implement adversarial tests. Adversarial testing can iteratively and systematically probe the state space for weaknesses and failures. These diverse iterative tests will produce an ensemble of evidence that varies in scope and form. For the evidence to be of use, the test team must aggregate it into a unified assurance argument that can

be used by the *technical warrant holder,* responsible party, or responsible organization to make a certification decision.

The evidence aggregation process can be greatly simplified if a runtime assurance manager is being deployed onboard the target platform. Runtime assurance managers provide safeguards against unacceptable decisions that violate safety protocols or rules of engagement. When a *validated* runtime assurance manager is deployed, the strength of evidence required for certification is relaxed, as the focus shifts to demonstrating that the autonomy operates within defined risk tolerances. The runtime assurance manager ensures that when the system encounters conditions outside of these tolerances, it mitigates risk by intervening, ensuring that system performance degrades gracefully rather than leading to catastrophic failure.

Because the AI being used to make autonomous decisions is based upon some form of model and/or training set, sustained performance requires continuing validation of the autonomy to affirm that the underlying models remain valid with respect to the operating environment. Continuing validation should be performed by a combination of onboard runtime assurance and/or post-mission maintenance activities under which consistency tests are performed to confirm that changing operational conditions remain within the model performance parameters.

When a common framework is being used to develop a family of unmanned platforms, the T&E of these platforms can be greatly accelerated by utilizing a *digital test range*. The digital range consists of autonomy test tools tailored to support autonomous systems using a predefined autonomy framework. An autonomy reference implementation of AI libraries, validated systems models, model-based decision models, and runtime assurance engines allows the test team to rapidly establish limited objective experiments for requirements analysis and component and assembly experimentation. A joint digital range as shown in Figure 5-3 can provide reusable infrastructure to support these processes.

More information about resources for digital ranges will be added in future expansions to this guidebook.

5.1.2 Scientific Test and Analysis Techniques for Autonomous Systems

The second overarching method is STAT for autonomous systems. STAT for autonomous systems, or STAT for autonomy, leverages rigorous scientific principles to ensure the reliable, safe, and effective operation of these complex systems. STAT, a collection of deliberate and methodical processes and procedures, offers a robust framework for addressing the complexities of autonomous systems T&E. By integrating the scientific method into all phases of testing, STAT enables the development of efficient, defensible, and decision-enabling test strategies, test plans, test designs, and test data analysis.

Description of STAT for Autonomous Systems

STAT for autonomy employs a structured, hypothesis-driven approach to T&E, mirroring the scientific method, to ensure that testing is objective, data driven, and focused on drawing meaningful conclusions about system capabilities and limitations.

STAT for autonomous systems:

- Emphasizes the formulation of testable hypotheses about system behavior, derived from system requirements and operational scenarios, to focus T&E activities and ensure that the collected data are relevant to assessing system performance.
- Prioritizes the collection of objective data through well-designed experiments and simulations to characterize autonomy performance and trustworthiness with maximum effectiveness and efficiency.
- Applies rigorous statistical analysis to interpret test data and draw conclusions about autonomous system performance, which ensures that conclusions are supported by evidence and that uncertainty is properly quantified.

The use of STAT for the T&E of DoD systems is required by DoD regulations.

Details and Best Practices

Key features of STAT for autonomous systems T&E include:

- Addressing complexity. Autonomous systems operate within intricate environments influenced by numerous factors, including environmental conditions, sensor inputs, human interactions, and internal algorithms. These factors can interact in complex ways, making it challenging to predict system behavior. STAT provides a structured approach to decompose this complexity and identify the most critical factors and interactions.
- Optimizing test efficiency. STAT employs techniques such as design of experiments to maximize the information gained from each test while minimizing the number of tests required. This scientific approach to test design helps to efficiently explore the vast operational space of the autonomous system and identify potential issues with fewer resources.
- Conducting sequential testing. STAT utilizes sequential testing, where testing continues to build upon prior test results to better characterize the system's performance. This adaptive approach allows for efficient use of testing resources over vast autonomy domains.

- Utilizing Bayesian statistics. Bayesian statistics are employed to incorporate prior knowledge and to update beliefs about system performance as new data become available, leading to more informed decision-making.
- Fostering collaboration and expertise. Effective implementation of STAT requires collaboration between test engineers, statisticians, and subject matter experts. The STAT Center of Excellence (COE) can provide valuable resources and expertise.
- Facilitating early integration. Integrating STAT into the T&E process early in the development cycle allows for iterative learning and informed decision-making throughout the system's life cycle.
- Enhancing flexibility and adaptability. STAT helps accommodate a T&E process that is flexible and adaptable to accommodate changes in requirements, unexpected test results, and evolving technologies.
- Improving documentation. Clear and comprehensive documentation of the T&E process, including test plans, data analysis, and results, is essential.

Primary Outcomes and Additional Benefits

The primary outcomes and additional benefits of STAT for the T&E of autonomous systems include the following:

- Primary outcomes:
  - Accurate characterization of performance and risk. By applying rigorous scientific methods, STAT provides strong evidence to support claims about system performance and trustworthiness, as well as to identify and quantify potential risks.
  - Improved system performance. The iterative nature of STAT allows for continuous improvement of the system based on data-driven feedback.
- Additional benefits:
  - Enhanced understanding of system behavior. STAT provides insights into how the system performs under different conditions and in response to various stimuli.
  - Reduced development time and costs. Early identification of potential issues through rigorous testing can prevent costly rework later in the development cycle.
  - Improved communication and collaboration among stakeholders. The structured approach of STAT facilitates clear communication and collaboration among developers, testers, and end users.

- Increased transparency and accountability. The formalized documentation practices of STAT promote transparency and accountability in the T&E process.
- Better informed decision-making. STAT provides decision-makers with the objective data and analysis they need to make informed decisions about system development and deployment.
- Facilitation of certification and accreditation. The rigorous testing and documentation practices of STAT can support the certification and accreditation of autonomous systems for operational use.

Requirements and Mission Decomposition

The foundation of STAT for DoD systems lies in a thorough examination of the system requirements. For autonomous systems, this examination can be particularly challenging because of the systems' complexity and the potential for unpredictable behavior, which may mean that system requirements must be derived from mission capability needs. A clear understanding of requirements is crucial for effective and efficient T&E.

- **Performing Comprehensive Requirements Analysis**. This step includes decomposition of mission objectives; human-machine interaction and teaming aspects; the CONOPS; supporting assets; and the relationships between the system, threats, and other entities within a mission scenario. By decomposing the mission capabilities along with the conditions and scenarios where they will be employed, the test team achieves understanding of what, where, when, with whom, and how the system must perform.
- **Understanding Expected Autonomy Behaviors**. Autonomous systems pose new challenges in capability requirements because executing a mission task effectively may mean more than just reaching some end state—"how" the system does a task may be important to mission effectiveness based on interactions, such as deconfliction, with allies, teammates, bystanders, and mission command authorities. The expected means of accomplishing a task may result in additional derived requirements for the autonomous system.
- **Defining Test Objectives**. Requirements that are well-defined, specific, measurable, achievable, relevant, and traceable will then allow effective test objectives to be established, guiding the subsequent steps in the STAT process and test planning.

STAT for Autonomous Systems Test Planning

Effective test planning is crucial for the successful T&E of autonomous systems. Because of autonomous systems' complexity and their potential for unpredictable behavior, careful consideration must be given to various factors throughout the planning process.

Key steps in STAT for autonomous systems test planning:

1. **Assess Stakeholder Concerns**: Understand stakeholder expectations and priorities through clear communication channels and establish a prioritized list of test events.
2. **Perform Decomposition**: Break down the autonomous system into manageable components or behaviors (e.g., sensors, perception, planning, control) to focus testing efforts. Decomposition can occur at both the mission and system level.
3. **Establish a Test Iteration Cycle**: Adopt an iterative approach to testing, allowing for adaptation and learning as knowledge is gained.
4. **Generate a Comprehensive Input and Output List**: Identify and document all relevant input and output variables, defining the system's performance envelope and measurement metrics.
5. **Consider Test Execution Constraints and Limitations**: Account for limitations in randomization, factor control, and resource availability when designing test procedures.
6. **Establish a Data Pipeline**: Plan for data reduction, analysis methods, and reporting to ensure efficient and timely processing of test data.
7. **Build a Test Matrix**: Develop a test matrix that aligns with the chosen T&E method (e.g., designed experiment, observational study) and respects resource constraints.
8. **Perform Data Analysis**: Employ appropriate statistical techniques to analyze test data, validate models, and draw meaningful conclusions about system performance.

STAT for Test Design of Autonomy M&S Systems

STAT has a crucial role in the design and analysis of simulations for autonomous systems. It involves creating simplified representations, sometimes called meta-models (surrogate models), of complex simulations to facilitate the efficient exploration and analysis of the system's operational space.

Key considerations of STAT for M&S:

- **Comprehensive Sampling**: Ensure thorough exploration of the simulation's state space to capture critical system behaviors and avoid missing key areas.

- **Surrogate Model Development**: Divide the simulation into manageable components and develop surrogate models for each, linking them together to form a meta-model.
- **Stakeholder Concerns**: Clearly understand stakeholder expectations and priorities for the simulation to guide meta-model development and analysis.
- **Process Decomposition**: Analyze the simulation architecture and identify key functions and potential bottlenecks to inform the meta-modeling strategy.
- **Input and Output List**: Document all relevant input and output variables for each subcomponent of the meta-model, considering potential differences from the real-world system.
- **Data Pipeline**: Establish an efficient data pipeline for extracting and processing simulation data for meta-model development.
- **Test Matrix**: Design test matrices for each surrogate model, considering the importance of factor significance and model prediction.
- **Test Iteration Cycle**: Adopt an iterative approach to meta-modeling, especially for simulations under continuous development, to adapt to evolving requirements and priorities.
- **Test Execution**: Consider the deterministic nature of the simulation and potential benefits for test execution, while still acknowledging the importance of randomization for risk mitigation.
- **Curse of Dimensionality**: Address the challenges posed by high-dimensional data through appropriate techniques such as state space partitioning, principal component analysis, or increased sample size.

STAT for Analysis of Autonomy M&S Systems

Once a meta-model is developed, appropriate analysis techniques are employed to gain insights into the behavior of the autonomous system within the simulation environment. Validating the overall simulation against real-world data is crucial to ensure its accuracy and reliability in representing the behavior of the autonomous system.

- **Comparison with Real-World Data**: Compare simulation predictions with real-world observations, focusing on key factors identified through meta-model analysis.
- **Goodness of Fit**: Evaluate the goodness of fit of the surrogate models, paying close attention to potential overfitting and employing validation techniques to ensure model accuracy.

- **Residual Analysis**: Analyze residuals to identify potential biases in the surrogate models and assess their overall performance.
- **Use of Statistical Criteria**: Apply published scientific statistical criteria to quantify how well the live test data matches simulation and model results, not just for the data mean but for variability such as standard deviation as well.
- **Model Use**: Utilize the meta-model to inform real-world testing, generate rapid predictions, and answer specific stakeholder questions.
- **Combined Analysis**: Combine real-world and simulation data in a single analysis to identify discrepancies and areas where the simulation deviates from reality.

Costs, Limitations, and Assumptions

The use of STAT for autonomous systems may have the following negative impacts:

- Involves initial costs and time in planning. Implementing STAT may require investment in new tools, training, and personnel. In most cases, however, the initial overhead results in more streamlined, efficient, and effective test design, execution, and analysis.
- Requires expertise in statistics and data analysis. Effective implementation of STAT requires personnel with expertise in statistics, data analysis, and experimental design.

Tools and Resources

For more information and tools that support STAT and its benefits for the DT&E of autonomous systems, see:

- STAT COE Website (https://www.afit.edu/stat/index.cfm) to get access to experts for help.
- Ask-a-STAT resource (https://www.afit.edu/STAT/page.cfm?page=498) for quick assistance.
- STAT COE Test Planning Guide for detailed guidance on initiating the requirements analysis process and developing comprehensive test plans.
- Model validation guidance in the Institute for Defense Analyses Handbook on Statistical Design and Analysis Techniques for M&S Validation (Wojton et al. 2019).

Challenges Addressed by This Method

STAT for autonomous systems helps to address several challenges including:

- **T&E as a Continuum**. STAT for autonomy provides a rigorous T&E methodology that accounts for iterative testing processes.
- **Requirements**. The STAT process helps identify derived requirements and leads to well-defined, measurable test objectives.
- **Safety**. Rigorous testing and analysis help to identify and mitigate potential safety hazards.
- **Ethics**. STAT promotes transparency and accountability, helping to address ethical concerns related to the development and deployment of autonomous systems.
- **Data**. STAT emphasizes the importance of data management and analysis for drawing scientifically defensible conclusions to understand system behavior.
- **HAT**. STAT provides a structured approach to assessing the interaction between humans and autonomous systems.
- **Test Adequacy and Coverage**. STAT supports the ability to measure and assess the breadth and depth of testing. Ensuring test adequacy and coverage is the most important challenge that STAT can address.
- **Autonomy Integration and Interoperability**. STAT provides a process for structuring evaluation plans to understand how autonomous systems interact with each other, the environment, and human operators.

5.1.3 Modeling and Simulation for Autonomy Test and Evaluation

The final overarching method for autonomy T&E is M&S, which provides a powerful framework for analyzing, testing, and refining autonomous systems in controlled environments. By replicating operational scenarios, M&S enables early evaluation of system performance, identifies potential issues, and informs design and development processes, ultimately reducing the risks and costs associated with real-world testing. M&S is most effective as a complement to live testing and should not be viewed as a replacement for it.

This section provides an overview of the basics of M&S for autonomous systems and how it supports and is supported by live T&E. Concise, basic information about the best practices, limitations, and challenges of M&S for autonomy T&E are presented first, followed by a more detailed discussion of these concepts.

Description of M&S for Autonomy T&E

M&S for autonomy T&E:

- Involves virtual or physical representations of autonomous systems, scenarios, and interactions to evaluate performance, trustworthiness, and even design choices.
- Simulates operational environments to test autonomous system performance under a wide range of conditions, including potentially difficult or unsafe test scenarios.

Details and Best Practices

Key features of M&S for the T&E of autonomous systems include:

- Clearly defining the goals and objectives of the M&S effort by identifying the specific questions to be answered and the metrics to measure success.
- Using independent, government-owned simulation capabilities to provide separation from developer goals to ensure unbiased and useful evaluations.
- Using standardized and accredited M&S test environments, scenarios, and methodologies to provide credible results to stakeholders.
- Using high-fidelity simulations to replicate complex operational scenarios for the accurate and reliable evaluation of system capabilities.
- Modeling uncertainty and variability and integrating real-world data, when possible, into models to enhance realism and ensure that scenarios are representative of the expected or even unexpected mission conditions.
- Supporting iterative development by allowing rapid experimentation, testing, and refinement of the system designs and algorithms.
- Incorporating multi-agent simulations to assess the interactions between autonomous systems, human operators, and other teammates or assets.
- Providing a scalable framework to evaluate system performance under widely varying operational, environmental, and adversarial conditions.
- When practical, using actual autonomous system software (the exact software used in the actual system) within the simulation environment to enable key insights and evaluation of algorithms and processes.
- Rigorously validating and verifying models used in the simulation to ensure the accurate representation of the real-world system and environment by comparing simulation results with data from physical tests.
- Regularly updating and improving the models used in the simulation as the system and its operating environment evolve to help ensure that the M&S effort remains relevant and effective.

Primary Outcomes and Additional Benefits

The primary outcomes and additional benefits of M&S for the T&E of autonomous systems include the following:

- Primary outcomes:
  - Early identification of performance issues, design flaws, and operational risks through controlled, repeatable testing scenarios.
  - Validation of system capabilities in a wide range of operational conditions, reducing the cost, schedule, safety, and security risks of live testing.
- Additional benefits:
  - Enhanced stakeholder confidence through the transparent and detailed evaluation of system performance.
  - Support for requirements validation by aligning the system design with operational needs before deployment.
  - Facilitation of multi-domain testing by integrating land, air, sea, and space environments within a single simulation framework.
  - Ability to model and test complex interactions, including HAT and multi-agent operations.
  - Cost savings by minimizing the need for physical prototypes and extensive real-world hardware testing.
  - Mitigation of safety and security risks by using simulated test and support assets without endangering or compromising high-value hardware assets.
  - Acceleration of development timelines by allowing the rapid iteration and evaluation of new system designs.

Costs, Limitations, and Assumptions

The use of M&S for autonomy T&E may have the following negative impacts or trade-offs:

- High upfront costs for developing detailed models, simulations, and the necessary infrastructure and personnel to execute them.
- Limited access to intellectual property, such as software emulation or interface control documents, which may hinder the proper integration of autonomous systems in M&S environments. Contracts may need to be negotiated to ensure that cognitive and other software is available for M&S purposes.

- Fidelity of assumptions made in simulation environments not fully representing real-world complexities, leading to gaps in testing accuracy.
- Limited availability of high-fidelity data to populate models, which can reduce the validity of simulation outcomes.
- Potential for overreliance on simulation results, which may lead to unexpected performance issues in real-world applications.
- Lack of awareness among programs, testers, and developers regarding potential M&S frameworks and resources for autonomous systems.

Tools and Resources

For more information and tools that support M&S for autonomy T&E and its benefits for the DT&E of autonomous systems, see:

- DoDI 5000.61, "DoD Modeling and Simulation Verification, Validation, and Accreditation."
- M&S for T&E Guidance in the Director, Operational Test and Evaluation (DOT&E) TEMP Guidebook.
- Advanced Framework for Simulation, Integration, and Modeling (AFSIM) (https://dsiac.dtic.mil/models/afsim/).

Challenges Addressed by This Method

M&S for autonomy T&E helps to address several challenges for the T&E of autonomous systems including:

- **T&E of the OODA Loop**. M&S supports iterative testing of the observe, orient, decide, and act processes by incorporating real-time feedback into system behavior evaluations.
- **Test Adequacy and Coverage**. M&S ensures comprehensive testing by offering a solution to address all operational conditions and edge cases.

Considerations and Applications for Autonomy M&S

M&S provides a versatile framework for addressing challenges in the development and testing of autonomous systems. Beyond standard evaluations, M&S enables the exploration of edge cases, adversarial conditions, and emerging technologies, ensuring that systems are robust, reliable, and adaptable to operational demands. The following discussion provides additional details about these and other benefits and challenges.

Types and Uses of M&S for Autonomy T&E

A variety of models and simulations can be used in the autonomy T&E process, from R&D to acquisition. Models that the test engineer will find useful include:

- Cognitive models: Abstract representations of the autonomy under test.
- System models: Abstract representations of the host platform, its constituent parts (including sensors), and the relationships between those parts.
- Environmental models: Abstract representations of the world in which the system exists including causal relationships between the system and the world.
- Cognitive simulations: Simulacrums of the "thought process" of the autonomy under test, which include the "orient" and "decide" phases of the OODA loop process.
- FTRT simulations: Often constructive simulations that simulate the performance of the autonomy under test in a synthetic world at high rates of speed.
- Real-time simulations: Simulations that simulate the performance of the autonomy under test in a synthetic world that runs on "wall clock" time.
- Virtual simulation: A simulation that involves real people operating simulated systems (often at real time).
- Constructive simulation: A simulation that involves simulated people operating simulated systems (often at FTRT).

As shown in Table 5-1, these models and simulations can be used throughout the acquisition process, for uses that include:

- Requirements analysis. Mission engineers can utilize cognitive models to examine the mission impact of hypothetical autonomous systems producing autonomy requirements that maximize mission performance.
- Formal analysis. Methods can use cognitive and system models and cognitive simulations to define theoretical performance limits of the autonomy under test.
- Simulation-based testing. This approach examines autonomy performance in very large numbers of circumstances, providing evaluation with broad scenario and condition coverage.
- Bench testing. This testing examines the timeliness of autonomy decisions.
- HITL testing. This method examines human-autonomy interactions by allowing the human to interact with the autonomy operating in a virtual world.

- LVC testing. This technique allows the safe, cost-effective testing of autonomy performance in large, complex, or high-risk encounters with a combination of live, virtual simulation and/or constructive simulation assets.

Table 5-1. Application of M&S Across the System Life Cycle

		Types of Methods					
		Cognitive Models	System Models	Environmental Models	Cognitive Simulation	Faster Than Real-Time Simulation	Real-Time Simulation
Types of Testing	Requirements Analysis	✓	✓	✓	✓	✓	
	Formal Analysis	✓	✓		✓		
	Simulation-Based Testing		✓	✓		✓	
	Bench Testing		✓	✓			✓
	Human-in-the-Loop Testing		✓	✓			✓
	LVC Testing		✓	✓			✓

Interoperability Testing with M&S

One of the key strengths of M&S is its scalability. Simulations can model the behavior of multiple autonomous systems working in coordination, offering valuable insights into collaborative tasks such as swarm dynamics, multi-agent interactions, and HAT. Modern weapon systems are increasingly required to operate within a larger, cross-Service, multi-domain warfighting architecture, where horizontal and vertical interoperability is critical. M&S provides a cost-effective and efficient means to evaluate system interoperability, addressing resource constraints and competing test priorities while enabling informed assessments of operational effectiveness in dynamic mission scenarios.

M&S Support to HSI Testing

M&S plays a pivotal role in HSI testing, encompassing domains such as training, human performance, workload, and usability. By leveraging tools such as the Improved Performance Research Integration Tool (IMPRINT) (Mitchell 2003) and Infantry Warrior Simulation (IWARS) (El Samaloty et al. 2007), M&S allows developers to evaluate how human operators

interact with autonomy-enabled systems in mission-critical scenarios. These simulations refine system usability, ensure appropriate workload distribution, and assess human performance metrics under realistic conditions, supporting the integration of HSI considerations into system design.

To advance HSI testing further, M&S must incorporate real-time, mission-level assessments that evaluate the interaction between humans and autonomous systems in operational contexts. This approach includes using validated M&S resources to predict system performance, identify risks, and support evaluations of system effectiveness and suitability. Additionally, HSI-focused simulations provide a safe and cost-effective means to evaluate operator training and situational awareness, reducing reliance on live testing while enhancing mission readiness and operational efficiency.

Test Optimization Using M&S

Another advantage of M&S is its ability to optimize testing strategies throughout the system life cycle. From initial concept development and requirements validation to deployment and sustainment, simulations provide a consistent platform for refining test approaches. This approach ensures that autonomous systems remain effective as operational needs evolve and new challenges emerge.

Using M&S, the value of decisions about a system can be quantified in terms of the system's true parameters. This value is derived directly from the TEMP for the SUT, the associated key performance parameters, and the decision set. By quantifying the operational impact of post–live test recommendations, M&S assigns high value to greenlighting systems with strong performance parameters and low uncertainty, while flagging systems with high uncertainty or poor performance for additional testing or revisions. This framework ensures that stakeholder priorities—such as mission risk, operational effectiveness, and schedule—are evaluated alongside testing costs in a unified model.

M&S regression further supports test optimization by predicting how future live-test options can improve the knowledge of system performance parameters. By simulating various scenarios, M&S identifies the most resource-efficient live-test program, reducing unnecessary testing while ensuring that critical performance data are captured effectively.

Integrated Test and Training with M&S

The integration of M&S with T&E and training creates a unified approach that enhances both system development and operator performance. Traditionally, training and system acquisition pipelines operate separately until merging at OT. With M&S, these pipelines can converge earlier, even before DT or the System Requirements Review, leveraging model-based systems

engineering tools such as the Systems Modeling Language (SysML) and Unified Modeling Language (UML) Testing Profile (UTP). This integration enables the creation of constructive and virtual simulations where operators begin training early, refining system behaviors and improving operational readiness simultaneously.

Early merging of T&E and training delivers significant benefits, particularly for systems incorporating ML. Expert operational personnel involved in training the ML model provide essential expertise for development and testing while simultaneously gaining familiarity and trust in the system. This approach ensures that operators understand how the system was built, the data on which the system relies, and how to use the system effectively, fostering a collaborative feedback loop for improvements. By aligning M&S, T&E, and training from the outset, this approach enhances system effectiveness, calibrates appropriate trust, and maximizes learning outcomes.

Technical Challenges of M&S for Autonomy T&E

M&S faces several technical challenges critical to evaluating autonomous systems. Accurate rotational transformations are essential for simulating movement through space; however, differing approaches between simulation platforms, such as Unity and Unreal Engine, complicate system integration in live-virtual settings. Similarly, kinematics and dynamics modeling—key for collision avoidance and interaction—require precise representations of motion and forces. These computations often involve trade-offs between simulation fidelity and processing speed, particularly when handling complex terrains, environmental interactions, or multi-agent scenarios.

Another technical challenge for autonomous system simulation is the modeling and execution of time. Autonomy decision engines may be optimized to run only at “real time”; thus, attempts to run FTRT simulations may lose realism and value. The ability to run many scenarios and test conditions FTRT is one of the key advantages of M&S, but computational hardware or software limitations of the autonomy under test may not allow realistic FTRT simulations.

Simulation-to-real challenges further complicate testing because aligning simulated outputs with real-world sensor data requires balancing cost, computational demands, and fidelity. Issues such as the “reality gap” emerge when simulated environments fail to account for real-world uncertainties, interactions, or collaborative agent behaviors. Effective integration also depends on accurate environment synchronization to bridge the gap between virtual and physical testing, particularly for reinforcement learning and training applications.

Dynamic environments and fidelity requirements also demand significant consideration. Simulating evolving conditions such as weather, time of day, or non-player character behaviors

adds complexity to virtual testing. Determining the appropriate fidelity involves identifying key questions and behaviors to evaluate, starting with simplified models, and iteratively refining them to balance resource constraints with critical knowledge gaps. Tailored approaches help ensure that simulations achieve the necessary accuracy while remaining practical and resource efficient.

5.2 Acquisition and Development Strategy

Effective T&E of autonomous systems must be strategized, planned, and designed into the program and system from the beginning to achieve justified evidence of performance and trustworthiness. This section discusses several practices related to an autonomous system's acquisition strategy or development strategy, which help to enable the effective, efficient, and robust T&E of autonomous systems:

- Operational modeling.
- Small-scale development.
- Open system architecture.
- Autonomy requirements and specifications.
- Continuous testing.
- Code isolation.
- Assurance cases.

These practices may not apply to every autonomy program, but where implemented, they help enable successful T&E of autonomous systems with reduced costs and time.

5.2.1 Operational Modeling

Operational modeling provides a conceptual framework for understanding the specific roles, tasks, and behaviors that an autonomous system will need to perform. This approach supports requirements development by detailing use cases and procedures, helping ensure that developers build the system with the intended functionality and enabling T&E personnel to objectively evaluate system performance of those tasks and behaviors.

Description of Operational Modeling

Operational modeling:

- Describes specific tasks, use cases, and workflows that the autonomous system must accomplish in its operational environment.
- Details the interactions and procedures between autonomous systems and other assets, ensuring alignment with mission requirements.
- Translates complex operational needs into structured concepts, helping to avoid misinterpretations and unnecessary design assumptions.

Details and Best Practices

Key features of operational modeling for the T&E of autonomous systems include:

- Defining use cases and visually representing key tasks through storyboards to capture requirements, such as practices used in software development, ensuring that developers understand the system's intended functions and interactions.
- Enabling storyboards, operational assumptions, and workflows to be validated with operators and end users to ensure accuracy and relevance.
- Breaking down complex operations into specific behaviors and actions, allowing developers to accurately translate requirements into system capabilities.
- Incorporating detailed procedural steps between autonomous systems and other mission assets, supporting effective requirements development and reducing misinterpretations.

Primary Outcomes and Additional Benefits

The primary outcomes and additional benefits of operational modeling for the T&E of autonomous systems include the following:

- Primary outcomes:
  - Provides a well-defined set of operational roles, tasks, and behaviors that guides developers in creating systems aligned with mission objectives and operational expectations.
  - Produces clearer, more actionable requirements that reduce ambiguity and enable effective, robust T&E by aligning meaningful test objectives and evaluation criteria.
- Additional benefits:
  - Minimizes costly redesign by ensuring early alignment on system functionality.
  - Enhances communication between stakeholders by visualizing complex requirements and workflows.

- Reduces the risk of misinterpretation, helping vendors and developers avoid building unnecessary or incorrect functionalities.
- Supports program managers by providing a structured roadmap of required capabilities and tasks.

Costs, Limitations, and Assumptions

The use of operational modeling may have the following negative impacts or trade-offs:

- Initial time and resource investment to create detailed operational models.
- Necessary modeling updates when requirements change.
- Reliance on subject matter expertise, which may limit modeling accuracy if expert input is unavailable, incomplete, or wrong.

Tools and Resources

For more information and tools that support operational modeling and its benefits for the DT&E of autonomous systems, see the Digital Engineering, Modeling and Simulation (DEM&S) Community of Practice Website (https://www.cto.mil/sea/dems_cop/).

Challenges Addressed by This Method

Operational modeling helps to address several challenges for the T&E of autonomous systems including:

- **T&E of the OODA Loop**. Operational modeling ensures that T&E can provide mission-relevant insights into decision-making under various use-case scenarios and conditions.
- **Requirements**. Operational modeling facilitates the validation of system requirements.
- **Personnel**. Operational modeling enables testers without expert domain knowledge to effectively evaluate autonomy behaviors with efficient, repeatable tests.
- **Safety**. Operational modeling ensures that the testing appropriately matches the complexity and conditions that the system will encounter in operational scenarios.
- **HAT**. Operational modeling provides use cases to realistically test and evaluate integrated human and autonomous system interactions and teaming.
- **Autonomy Integration and Interoperability**. Operational modeling offers scenarios and behaviors to test and evaluate autonomous systems' interactions with other platforms and protocols.

5.2.2 Small-Scale Development

Small-scale development uses inexpensive, simple platforms and assets to accelerate testing and iteration, enabling rapid prototyping and scaling for larger systems. This approach supports efficient development cycles while minimizing initial costs and resources.

Description of Small-Scale Development

Small-scale development encompasses the following:

- Cost-effective prototyping uses inexpensive, scalable platforms to test and refine capabilities before full-scale implementation.
- Rapid development using surrogate platforms that are less complex reduces costs and expedites development, allowing for iterative testing and accelerated refinement.
- Risk mitigation allows the identification and resolution of potential issues on surrogate platforms before deployment on costly systems.

Details and Best Practices

Key features of small-scale development for the T&E of autonomous systems include:

- Use of surrogate assets to replicate primary system functions, enabling early-stage testing and iterative refinement before committing to more expensive systems.
- Documentation of incremental progress, which tracks development and performance improvements over time using small-scale assets, facilitating acceptance and integration of larger, more costly systems.
- Reduced risk through phased testing that identifies and mitigates potential issues at a smaller scale, reducing the risks and costs associated with large-scale testing.
- Rapid adaptation to feedback that allows for quick modifications based on test results, expediting the refinement process and improving readiness for full-scale testing.

Primary Outcome and Additional Benefits

The primary outcome and additional benefits of small-scale development for the T&E of autonomous systems include the following:

- Primary outcome:
    - Enhanced efficiency and cost savings: Uses small, scalable assets to make system development and testing safer, quicker, and more affordable.

- Additional benefits:
  - Lower risk exposure: Conducts initial testing on surrogate platforms, reducing the risks associated with testing on costly, full-scale systems.
  - Reduced testing timelines: Speeds up development cycles by allowing rapid iteration and early validation, minimizing the time needed to move from prototype to full-scale implementation.
  - Flexible prototyping: Provides a modular and adaptive approach to test varying configurations, enabling teams to tailor designs based on evolving requirements.
  - Encouragement of innovation: Uses low-cost testing environments to allow for more experimentation, fostering innovative approaches to system development.

Costs, Limitations, and Assumptions

The use of small-scale development may have the following negative impacts or trade-offs:

- Representational inaccuracy. The assumption that small-scale assets can effectively mimic the performance and behavior of full-scale systems may not always hold true in complex scenarios.
- Risk of overlooking full-system interactions. Testing with simplified assets may miss critical interactions and dependencies present in the complete system, leading to gaps in T&E outcomes.
- Resource allocation for surrogate platforms. Although less expensive than full systems, small-scale assets still require investment in development, management, and maintenance.
- Computational limitations. Size constraints of smaller platforms may restrict onboard processing capacity.

Tools and Resources

Future updates to this guidebook will provide additional information and tools that support small-scale development and its benefits for the DT&E of autonomous systems.

Challenges Addressed by This Method

Small-scale development helps to address several challenges for the T&E of autonomous systems including:

- **T&E as a Continuum**. Small-scale development enables continuous, scalable testing throughout the development cycle.
- **Requirements**. Small-scale development validates requirements on a smaller scale for efficient adjustments.
- **Safety**. Small-scale development lowers risk by testing on small, low-cost assets first.
- **Data**. Small-scale development provides early performance data to guide larger-scale testing.

5.2.3 Open System Architecture

A major development in the acquisition strategy and systems engineering areas is the use of open system architecture, as part of a Modular Open Systems Approach (MOSA). The implementation of MOSA in autonomous systems provides great potential benefits to the systems' DT&E.

Basic Description of Open System Architecture

Open system architecture:

- Employs a modular design that uses modular system interfaces between major systems, major system components, and modular systems.
- Allows major system components to be incrementally added, removed, or replaced throughout the life cycle.
- Uses nonproprietary, open architecture standards for integrating subsystems and services into the mission package with government-owned interfaces.

MOSA is the DoD preferred method for the implementation of open systems and is required in accordance with Section 4401 of Title 10, United States Code.

Figure 5-4 shows the features and processes of MOSA.

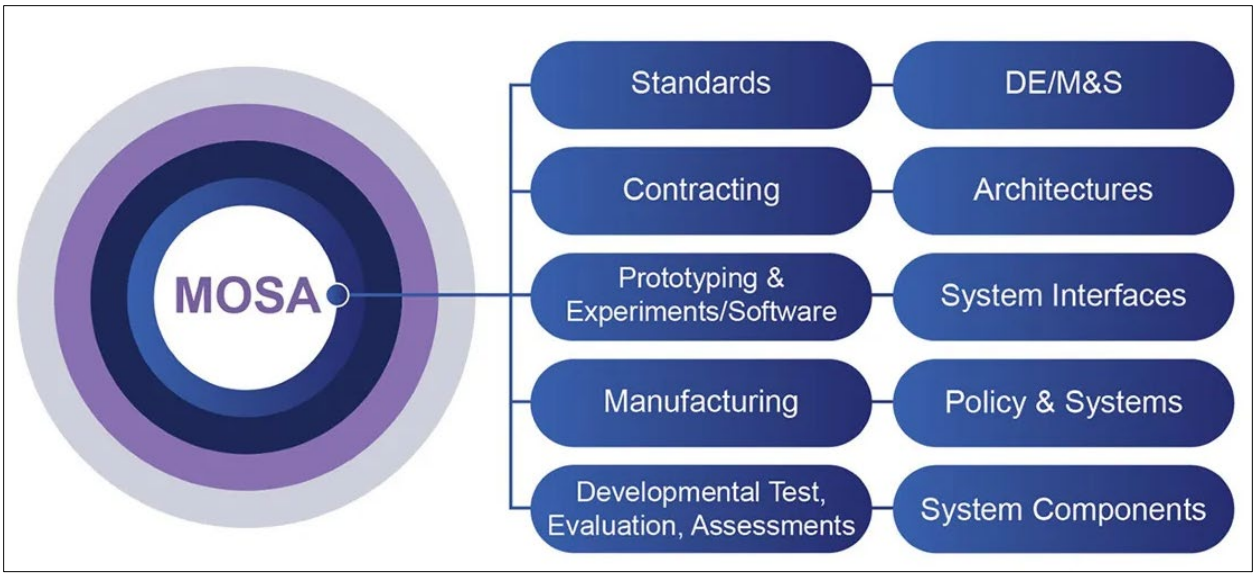


Figure 5-4. Features and Processes of a Modular Open Systems Approach

Details and Best Practices

Key features of MOSA for the T&E of autonomous systems include:

- Standardized government-owned open architectures with built-in hooks and capacity for internal cognitive T&E instrumentation.
- Subsystem, system, and SoS interfaces that allow the reuse and automation of T&E tools for data recording, processing, and analysis.
- Consistent subsystem purpose and functionality that supports modular T&E verification, as well as straightforward comparison between competitors.

Primary Outcomes and Additional Benefits

The primary outcomes and additional benefits of MOSA for the T&E of autonomous systems include the following:

- Primary outcomes:
  - Accurate T&E insight into internal system processes.
  - Portable T&E instrumentation and measures that can be reused.
- Additional benefits:
  - The ability to rapidly share information across domains, with quick and affordable updates or improvements to both hardware and software components.

- Allowance of severable major system components, affording opportunities for enhanced competition and innovation.
- Significant cost savings or avoidance.
- Schedule reduction and rapid deployment of new technology.
- Opportunities for technical upgrades and refresh.
- SoS interoperability and mission integration.

Costs, Limitations, and Assumptions

The use of open system architecture and related approaches may have the following negative impacts:

- Cost of infrastructure and personnel to organize, design, manage, implement, and maintain open architecture standards, interfaces, and tools.
- Limitations on system design uniqueness and potential efficiency.
- Potential for vulnerability and degradation of the technical edge due to adversary insight into system architecture and design.
- Obscuring of growing interdependencies between components by modular architecture, requiring ongoing management and maintenance efforts, especially in autonomous systems.
- Emergence of complex system failures when individual components interact in unintended ways. Hidden coupling can obscure the root causes of failures and lead to cascading failure propagation that is difficult to diagnose.
- Need for T&E to verify that open architecture goals have been achieved in system development and will effectively support future upgrades and maintenance.

Tools and Resources

For more information and tools that support open system architecture and its benefits for the DT&E of autonomous systems, see:

- Open Architecture Management (OAM) Website (https://www.vdl.afrl.af.mil/programs/oam/index.php).
- Open Mission Systems in a Nutshell (available on the OAM Website).
- Naval Air Systems Command (NAVAIR) MOSA Website (https://www.navair.navy.mil/MOSA).

- Unmanned Maritime Autonomy Architecture.
- AFSIM (https://dsiac.dtic.mil/models/afsim/).

Challenges Addressed by This Method

Open system architecture helps to address several challenges for the T&E of autonomous systems including:

- **T&E as a Continuum**. Open system architecture supports the reuse and automation of T&E tools.
- **T&E of the OODA Loop**. Open system architecture allows cognitive instrumentation for decision-making insights and root cause analysis.
- **Requirements**. Open system architecture simplifies and standardizes how requirements are tested.
- **Infrastructure**. Open system architecture allows reusable T&E tools that measure standardized processes.
- **Safety**. Open system architecture enables standardized checks and controls on autonomous behaviors.
- **Data**. Open system architecture standardizes data formats and mandates well-defined messaging.
- **HAT**. Open system architecture supports standardized human roles and authority delegation and provides a foundation for human expectations.
- **Black Box Components**. Open system architecture supports insight into direct inputs and outputs of hard-to-explain component performance and measurement.
- **Autonomy Integration and Interoperability**. Open system architecture provides interfaces and standards.

5.2.4 Autonomy Requirements and Specifications

In the requirements development process, it is essential to include individuals with T&E expertise in developing and evaluating autonomous behaviors. Often, stakeholders fail to recognize when they are imposing assumptions about how to do the task based on existing practice. The efficient or effective way for an autonomous system to do a task may be very different from the way a human would do the task. It is important to ensure that the requirements are driven by operational need, not stakeholder assumptions.

Description of Autonomy Requirements and Specifications

This method involves effective requirements and specifications development for autonomous systems employing a deep understanding of autonomous behaviors and operational needs—ensuring that requirements are not based on assumptions or how humans perform tasks, but rather on what the autonomous system needs to achieve.

Details and Best Practices

Key features of autonomy requirements and specifications for the T&E of autonomous systems include:

- Focusing primarily on what the system should do, not how it should do it—unless the how is also a clear requirement.
- Defining which decisions are the responsibility of the autonomous system.
- Ensuring that requirements reflect the minimum capability needed.
- Testing for internal conflicts and inconsistencies in requirements.
- Using measurable goals whenever possible.
- Following standards such as the Institute of Electrical and Electronics Engineers (IEEE) 1872.1-2024, "IEEE Standard for Robot Task Representation."
- Using ontologies to simplify requirement development and consistency checks.
- Documenting assumed operational constraints.
- Clearly defining responsibilities and interfaces between autonomy, runtime assurance, and fault management.
- Using formal methods where possible, but carefully documenting assumptions.
- Allowing for flexibility in requirements to accommodate changes and undocumented failure modes.

Primary Outcome and Additional Benefits

The primary outcome and additional benefits of the autonomy requirements and specifications method for the T&E of autonomous systems include the following:

- Primary outcome:
    - The development of clear, concise, and testable requirements that accurately reflect the desired capabilities of the autonomous system.

- Additional benefits:
  - Improved communication and understanding between stakeholders.
  - Reduced development time and cost.
  - Increased system reliability and safety.
  - Enhanced user trust and confidence.
  - Facilitated integration and interoperability with other systems.
  - Improved ability to manage complexity and risk.
  - Ensured support for ethical considerations in autonomous system design.

Costs, Limitations, and Assumptions

The use of the autonomy requirements and specifications method may have the following negative impacts:

- Increased initial effort required to develop comprehensive requirements.
- Potential difficulty in quantifying some qualitative goals.
- Need for specialized expertise in autonomous behavior and formal methods.

Tools and Resources

For more information and tools that support effective requirements for the DT&E of autonomous systems, see Requirements Management, one of the eight technical management processes included in the Defense Acquisition University (DAU) Systems Engineering Brainbook (https://www.dau.edu/tools/dau-systems-engineering-brainbook).

Challenges Addressed by This Method

The autonomy requirements and specifications method helps to address several challenges for the T&E of autonomous systems including:

- **Requirements**. The autonomy requirements and specifications method provides a structured approach for developing clear, concise, and testable requirements.
- **Safety**. By focusing on minimum capabilities and documenting assumptions, this method helps to ensure system safety.
- **HAT**. The autonomy requirements and specifications method promotes a clear definition of roles and responsibilities in HATs.

- **Autonomy Integration and Interoperability**. The autonomy requirements and specifications method supports the development of requirements that facilitate integration and interoperability with other systems.

5.2.5 Continuous Testing

Continuous testing for autonomous systems refers to an ongoing process that integrates testing throughout the development and operational life cycle of autonomous technologies. Continuous testing is essential to ensure that autonomous systems software and hardware components are reliable, safe, and capable of adapting to changing conditions and requirements. This approach helps track the growth and evolution of the autonomous capabilities over time as they take on more numerous and complex tasks. This approach is critical to establishing a CI/CD pipeline of capability.

Description of Continuous Testing

Continuous testing:

- Integrates what were traditionally separate and often consecutive test processes into an integrated agile framework.
- Focuses on collaboration and having a single test team.
- Is iterative, learning from each test to rapidly inform future testing.
- Strives to be adaptive to rapidly changing requirements and operational needs to enable CI/CD of capabilities.

Details and Best Practices

Key features of continuous testing for the T&E of autonomous systems include:

- Testing early and often: Begin testing at the earliest stages of development and continue throughout the life cycle and encourage rapid feedback loops to identify issues quickly.
- Modular testing: Break down systems into smaller, manageable components for more focused testing to make it easier to isolate and address specific issues.
- Cross-disciplinary collaboration: Involve cross-functional teams, including software developers, hardware engineers, and safety experts, to enhance the overall understanding of system interactions.
- Automated testing frameworks: Utilize frameworks that support automated unit, integration, and system testing and implement CI/CD pipelines for seamless updates.

- Simulation environments: Use realistic simulations to test various scenarios that an autonomous system may encounter (e.g., adverse weather, unexpected obstacles) because simulations can help in assessing performance and safety without real-world risks.
- Hardware-in-the-loop testing: Integrate real hardware with simulation models to validate the system's performance in a controlled setting to identify hardware-related issues early.
- Safety and compliance testing: Conduct thorough safety analyses including hazard analysis and risk assessments concurrent with testing.
- Continuous monitoring: Implement real-time monitoring of deployed systems to gather data on performance and identify potential failures.

Primary Outcome and Additional Benefits

The primary outcome and additional benefits of continuous testing for the T&E of autonomous systems include the following:

- Primary outcome:
  - Adaptation to change. Autonomous systems often need updates or improvements based on new data, algorithms, or environmental changes. Continuous testing ensures that these changes do not introduce new issues.
- Additional benefits:
  - Safety assurance. Autonomous systems operate in dynamic environments, where they encounter unpredictable scenarios. Continuous testing helps identify and mitigate risks, ensuring that the system behaves safely under various conditions.
  - Performance validation. As autonomous systems learn and adapt over time, continuous testing validates their performance against expected outcomes, ensuring they operate efficiently and effectively.
  - User trust. For users to trust and adopt autonomous systems, the systems must demonstrate consistent reliability and safety. Continuous testing helps build that trust through regular validation and performance checks.
  - Integration with other systems. Autonomous systems often need to interact with other technologies. Continuous testing ensures seamless integration and functionality in diverse environments.

Costs, Limitations, and Assumptions

The use of continuous testing may have the following negative impacts or trade-offs:

- Contract concerns. Providing adequate contract and sustainment support for continuous testing can be potentially problematic.
- Scalability. As systems grow in complexity and scale, maintaining effective continuous testing can become increasingly difficult.
- Integration. Assuming that CT, DT, OT, and operator training can be efficiently integrated through continuous testing across the life cycle of the system may be optimistic.
- Policy. Current policy does not always support continuous T&E processes.

Challenges Addressed by This Method

Continuous testing helps to address several challenges for the T&E of autonomous systems including:

- **T&E as a Continuum**. Continuous testing integrates traditionally separate testing phases into an iterative process that evolves with system development and deployment, ensuring ongoing validation and refinement.
- **T&E of the OODA Loop**. Continuous testing supports the evaluation of how an autonomous system observes, orients, decides, and acts over time, identifying potential latency, failure modes, or biases in decision-making.
- **Personnel**. Continuous testing reduces the reliance on large, periodic test events by enabling continuous validation, allowing personnel to focus on assessing emerging risks and system adaptations rather than reacting to unexpected failures late in development.
- **Exploitable Vulnerabilities**. Continuous testing enables the early detection of security weaknesses by continuously testing for adversarial threats, cyber vulnerabilities, and unexpected system behaviors.
- **Data**. Continuous testing ensures a steady influx of test data for training, validation, and performance monitoring, improving AI/ML model robustness and adaptability over time.
- **Black Box Components**. Continuous testing facilitates the ongoing testing of proprietary or opaque system components, ensuring that even if internal mechanisms are unknown, system behavior remains predictable and safe.
- **Dynamic Learning**. Continuous testing provides a structured approach to monitoring and testing autonomous systems that adapt and learn over time, ensuring that changes do not introduce unintended failures or regressions.

- **Test Adequacy and Coverage**. Continuous testing helps ensure comprehensive testing across a range of conditions by continuously expanding the test space, reducing gaps in evaluation and improving confidence in system readiness.

5.2.6 Code Isolation

Code isolation is a useful development strategy for ensuring the safety and security of autonomous systems. This method has benefits for T&E, saving costs and time by reducing the quantity of testing and the test rigor necessary to evaluate software changes by allowing smaller-scoped T&E for noncritical software changes.

Description of Code Isolation

Code isolation:

- Involves the use of a software code development framework that enables the separation of software components, reducing the risk of failures and vulnerabilities.
- Can separate safety-critical software code or mission-critical code from noncritical code to ensure operational safety and prevent unintended consequences.

Note: Code isolation is sometimes referred to as *software compartmentalization* in the software engineering community.

Figure 5-5 depicts the code isolation of critical and noncritical software by showing how air gaps separate different types of software development.

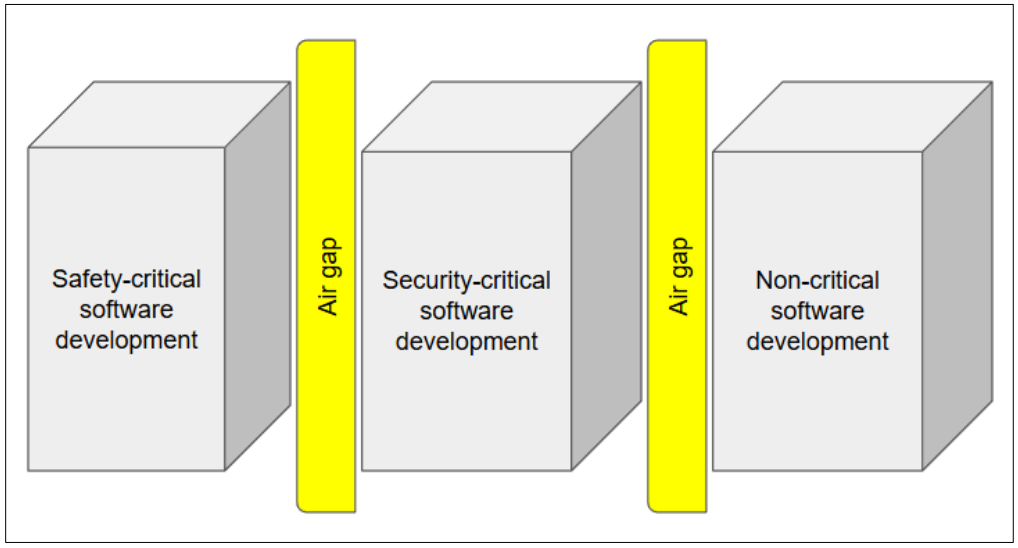


Figure 5-5. Code Isolation of Critical and Noncritical Software

Details and Best Practices

Key features of code isolation for the T&E of autonomous systems include:

- Isolating safety-critical software, such as flight control systems, from other task software.
- Isolating security-critical software, such as authentication and authorization modules, from other task software.
- Using separate development platforms and environments for critical and noncritical software to minimize interference and unintended interactions.
- Planning and executing robust, comprehensive T&E of software changes in critical modules.
- Planning and executing streamlined, limited T&E of noncritical software changes, based on the code isolation safeguards and protections set up for software development.

Primary Outcome and Additional Benefits

The primary outcome and additional benefits of code isolation for the T&E of autonomous systems include the following:

- Primary outcome:
  - Provides rapid and responsive T&E to a majority of (noncritical) software changes without the burdens of extensive testing.
- Additional benefits:
  - Facilitates rapid software development cycles while protecting key software from flaws and threats to software integrity.
  - Reduces T&E cost and schedule because of the ability to focus testing efforts on isolated components.
  - Supports a more modular and open architecture, which can improve system flexibility and maintainability.
  - Enhances system resilience by preventing failures in one component from cascading to others.

Costs, Limitations, and Assumptions

The use of code isolation may have the following negative impacts:

- Increases initial costs and time for setting up and implementing code isolation solutions.

- Requires an open, modular architecture to fully realize the benefits of code isolation.

Tools and Resources

For more information and tools that support code isolation and its benefits for the DT&E of autonomous systems, see the technical paper, "SoK: Software Compartmentalization" (Lefeuvre et al. 2024).

Challenges Addressed by This Method

Code isolation helps to address several challenges for the T&E of autonomous systems including:

- **T&E as a Continuum**. Code isolation enables targeted testing of isolated software components, reducing the need for full-system verification.
- **Exploitable Vulnerabilities**. Code isolation reduces the attack surface and mitigates the risk of malicious code compromising critical functions.
- **Safety**. Code isolation prevents failures in noncritical software from affecting safety-critical functions.

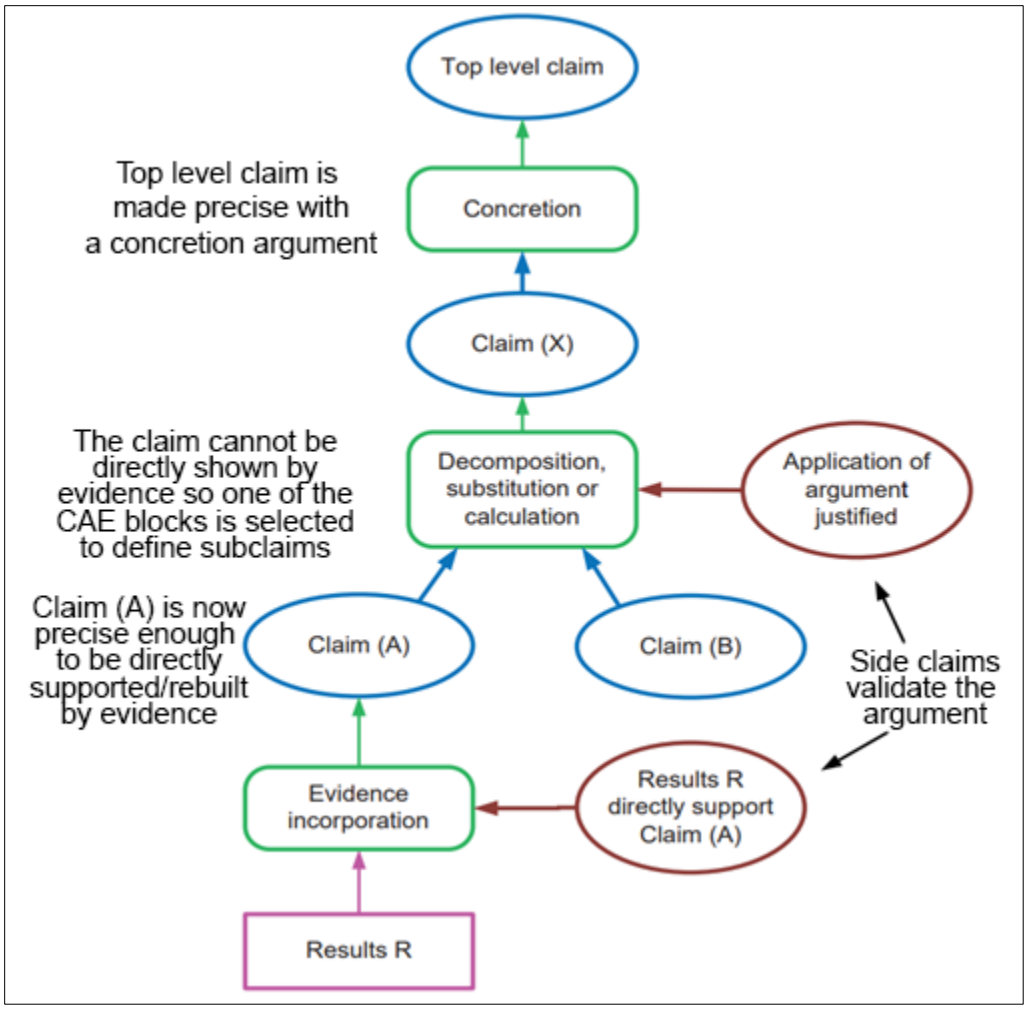
5.2.7 Assurance Cases

A useful method for organizing T&E data and other V&V information is the assurance case method. This approach provides a structured and documented body of evidence to demonstrate that an autonomous system satisfies its requirements and safety criteria, ultimately justifying confidence in its trustworthiness.

Description of Assurance Cases

The assurance case method is a structured argument, supported by evidence, that provides a compelling and valid case that a system is safe, secure, and fit for its intended purpose, and which is adaptable to the specific needs and context of the system being evaluated.

Figure 5-6 shows an example assurance case block diagram.



Source: Safety Case Templates for Autonomous Systems (Bloomfield et al. 2021)

Figure 5-6. Example Assurance Case Block Diagram

Details and Best Practices

Key features of assurance cases for the T&E of autonomous systems include:

- Identifying top-level claims: Clearly defining the desired properties of the system, such as safety, security, or reliability, as top-level claims in the assurance case.
- Decomposing and refining claims: Breaking down top-level claims into subclaims and refining them until they are specific enough to be supported by concrete evidence.
- Formulating arguments: Constructing clear and logical arguments that explain how the evidence supports the claims, addressing any uncertainties or challenges.
- Gathering evidence: Collecting diverse evidence from various sources, including testing, analysis, simulations, formal verification, and design documents.

- Addressing challenges: Identifying and documenting potential "defeaters" or challenges to the claims and providing counterarguments or mitigating factors.
- Promoting iterative development: Developing the assurance case in conjunction with the system development process, allowing for feedback and refinement of both the system and the case.
- Using a claims, arguments, and evidence (CAE) approach: Utilizing a CAE approach, one common structure for assurance cases, where claims about the system's properties are decomposed into subclaims, supported by arguments and evidence.
- Utilizing the Assurance of Machine Learning for use in Autonomous Systems (AMLAS) framework: Using the AMLAS process for developing assurance cases specifically for autonomous systems that incorporate ML components.

Primary Outcome and Additional Benefits

The primary outcome and additional benefits of assurance case methods for the T&E of autonomous systems include the following:

- Primary outcome:
  - Justified confidence: Providing stakeholders with a high level of confidence that the system is trustworthy and will operate as intended in its operational environment.
- Additional benefits:
  - Improved communication: Facilitating clear communication and understanding of the system's safety and capabilities among stakeholders.
  - Early issue identification: Enabling the early identification and mitigation of potential issues throughout the development life cycle.
  - Enhanced design: Informing and improving system design decisions by identifying potential weaknesses.
  - Support for certification: Providing a robust framework for demonstrating compliance with safety standards and regulations.
  - Traceability: Establishing clear links between requirements, design, and evidence.

Costs, Limitations, and Assumptions

The use of assurance cases may have the following negative impacts:

- Resource intensiveness. Developing and maintaining comprehensive assurance cases can be time-consuming and require significant resources.
- Complexity. Assurance cases for complex autonomous systems can become intricate and challenging to manage.
- Confidence calculation. Assessing the overall confidence in the assurance case, especially when dealing with uncertainties and subjective judgments, can be challenging.

Tools and Resources

For more information and tools that support assurance cases and their benefits for the DT&E of autonomous systems, see:

- AdvoCATE (Assurance Case Automation Toolset), a software tool, developed at the National Aeronautics and Space Administration (NASA) Ames Research Center, for recording and managing assurance cases using the Goal Structuring Notation, and the AdvoCATE User Guide.
- AMLAS Tool, a tool supporting the AMLAS process for assurance cases in autonomous systems with ML components (https://www.assuringautonomy.com/amlas/tool).
- Review of Potential Assurance Case Tool Options for DoD (Roback 2024).
- Safety Case Templates for Autonomous Systems (Bloomfield et al. 2021).

Challenges Addressed by This Method

Assurance case methods help to address several challenges for the T&E of autonomous systems including:

- **T&E as a Continuum**. Assurance cases support the concept of T&E as a continuum throughout the system life cycle, allowing for iterative evaluation and refinement.
- **T&E of the OODA Loop**. The structured argumentation in assurance cases can be used to analyze and evaluate the performance of autonomous systems within the OODA loop framework.
- **Personnel**. Assurance cases provide a structured approach to communicate system safety and performance, ensuring that decision-makers, testers, and operators understand system limitations and expected behaviors.
- **Exploitable Vulnerabilities**. By systematically considering potential hazards and vulnerabilities, assurance cases can help identify and mitigate exploitable weaknesses in autonomous systems.

- **Safety**. Assurance cases offer a rigorous framework for demonstrating and ensuring the safety of autonomous systems in their operational environment.
- **Ethics**. Assurance cases support transparency and accountability in autonomy decision-making by documenting how ethical considerations, such as bias mitigation and fairness, are addressed in system design and testing.
- **Data**. Assurance cases can address the challenges associated with data-driven autonomous systems by incorporating evidence from data collection, analysis, and model validation.
- **HAT**. Assurance cases ensure that interactions between autonomous systems and human operators are effectively tested and validated to promote trust and operational effectiveness.
- **Black Box Components**. Assurance cases provide a structured methodology to evaluate autonomous systems with opaque or proprietary components, ensuring that sufficient testing is conducted despite limited insight into internal behaviors.
- **Mission Evolution**. Assurance cases help assess system adaptability and robustness in evolving mission environments by structuring arguments that account for changing operational requirements.
- **Dynamic Learning**. Assurance cases support the V&V of autonomous systems with learning components by structuring arguments around how the system adapts over time and how learning processes are evaluated for safety and effectiveness.
- **Test Adequacy and Coverage**. Assurance cases establish a structured framework to ensure that testing is comprehensive and systematically covers all critical aspects of system behavior.
- **Autonomy Integration and Interoperability**. Assurance cases provide a means to document and evaluate how autonomous systems integrate with other platforms, ensuring reliability and consistency across various operational contexts.

5.3 Test Strategy

Test strategies for autonomous systems are critical for timely and effective evaluations that provide justified confidence in the system. This section discusses several practices related to an autonomous system's test strategy, which help to enable the effective, efficient, and robust T&E of autonomous systems:

- LVC testing.
- Experimentation T&E.

- Surrogate platforms.
- Formal verification methods.
- Adversarial testing.
- Post-acceptance testing.

These practices may not apply to every autonomy program, but where implemented, they help enable successful T&E of autonomous systems with reduced costs and time.

5.3.1 Live, Virtual, and Constructive Testing

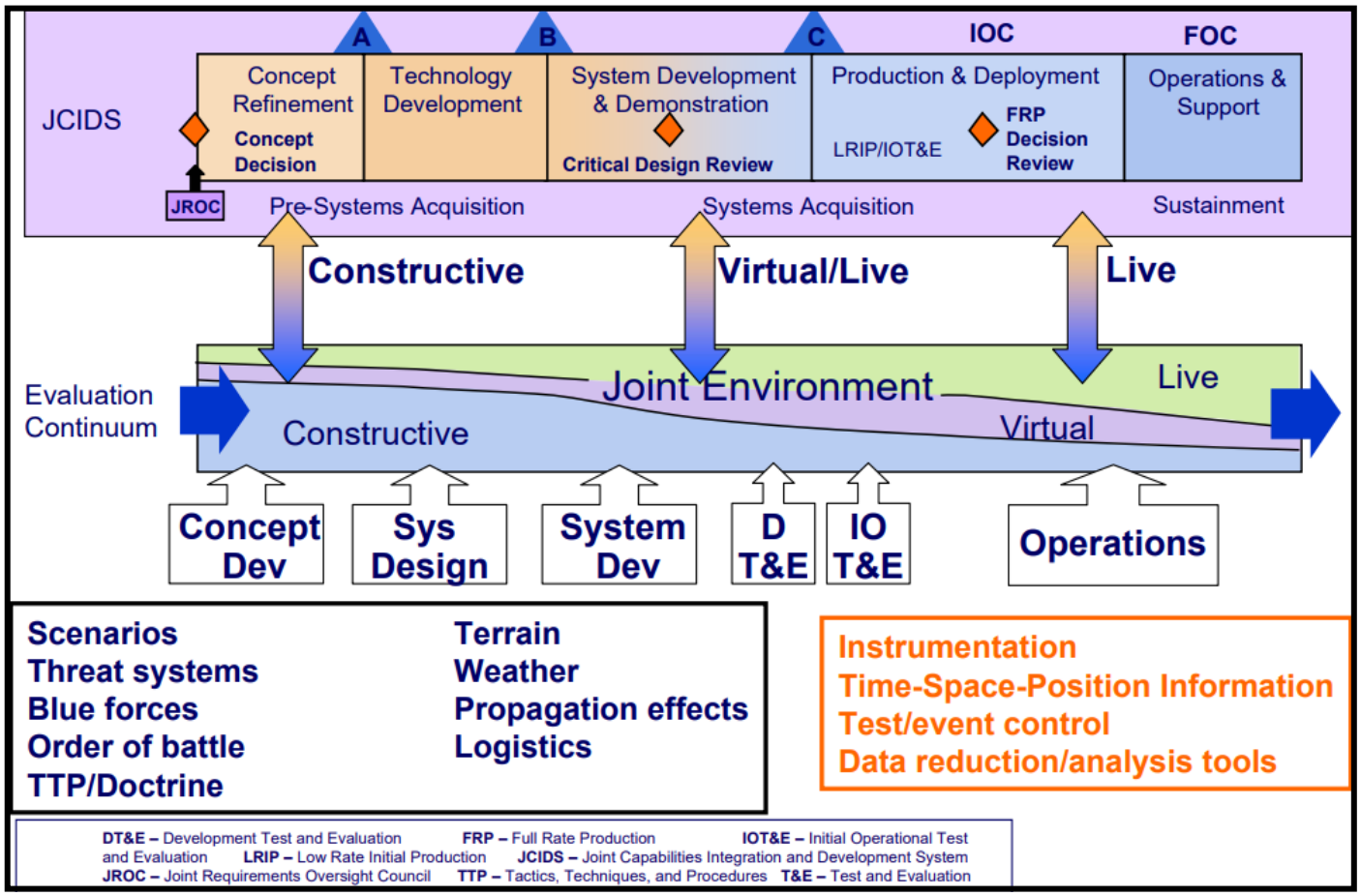
An increasingly useful method for DoD testing in recent years has been LVC testing, which is emerging as a proven method for the T&E of autonomous systems. LVC offers a powerful and flexible approach to assess these complex systems across a range of operational environments and scenarios, optimizing resources and minimizing risks.

Description of LVC Testing

LVC testing is a systems integration testing paradigm that incorporates a mix of real-world (live), simulated (virtual), and emulated (constructive) components:

- **Live (L)** refers to the use of actual systems and real-world environments in testing, which can include physical prototypes, operational personnel, and real-world locations.
- **Virtual (V)** refers to the use of simulated environments and systems, often involving HITL interactions, that can provide realistic representations of complex scenarios and allow for safe and controlled testing.
- **Constructive (C)** refers to the use of emulated or modeled systems, typically within a synthetic environment, which allows for large-scale, complex scenarios to be tested without the need for physical hardware or real-world locations.

Figure 5-7 depicts the use of LVC testing through the life cycle.



Source: Joint Test and Evaluation Methodology Overview (Bjorkman 2007)

Figure 5-7. Use of LVC Testing Through the Life Cycle

Details and Best Practices

Key features of LVC testing for the T&E of autonomous systems include:

- Progressive integration: Begin with predominantly constructive elements and gradually incorporate virtual and live components as the system matures to allow for early identification of issues in a controlled environment.
- Scenario variation: Utilize the flexibility of LVC to create a wide range of scenarios, including normal operating conditions, edge cases, and failure modes, to ensure comprehensive testing and robust system performance.
- HITL testing: Incorporate virtual and live components to evaluate human-machine interactions, assess operator workload, and refine human-machine interfaces.

- Data collection and analysis: Implement robust data collection and analysis procedures across all LVC domains to enable objective performance assessment, identification of deficiencies, and validation of system requirements.
- Common interfaces: Utilize open standards and protocols to ensure seamless data exchange between LVC components, using a "plug-and-play" approach that promotes interoperability and facilitates the integration of diverse test assets.
- Leveraging existing resources: Utilize government-owned and -managed platforms and frameworks for constructive simulation and testing whenever possible to optimize resource utilization and reduce costs.

LVC is not a new concept, but its importance is growing because of the increasing complexity of autonomous systems and the need for efficient and cost-effective testing. LVC provides a comprehensive and adaptable approach to T&E by combining the strengths of each of the three test processes.

Primary Outcomes and Additional Benefits

The primary outcomes and additional benefits of LVC testing for the T&E of autonomous systems include the following:

- Primary outcomes:
  - Enhanced system maturity. LVC enables rigorous testing across a wide range of scenarios, leading to improved system reliability, safety, and performance. By identifying and addressing potential issues early in the development life cycle, LVC helps reduce risk and ensure successful system deployment.
  - Accurate T&E insights. LVC provides valuable insights into the performance, effectiveness, and suitability of autonomous systems by enabling the comprehensive evaluation of autonomy tasks, components, subsystems, and capabilities.
- Additional benefits:
  - Reduced development costs. By leveraging simulation and emulation, LVC can reduce the need for expensive physical prototypes and real-world testing, leading to significant cost savings.
  - Increased test efficiency. LVC allows for the rapid iteration and testing of various configurations and scenarios, accelerating the development process.
  - Improved safety. LVC provides a safe and controlled environment for testing potentially dangerous scenarios, minimizing risks to personnel and equipment.

- Enhanced training. LVC simulations can be used to create realistic training environments for operators, improving their skills and familiarity with the autonomous system.
- Facilitated collaboration. LVC enables collaboration between different stakeholders, including developers, testers, and end users, by providing a common platform for T&E.

Costs, Limitations, and Assumptions

The use of LVC testing may have the following negative impacts:

- Simulation fidelity. Ensuring accurate representation of real-world environments and system behavior within VC components can be challenging.
- Integration complexity. Integrating LVC components can be technically complex and require specialized expertise.
- Cost of simulation tools. Acquiring and maintaining sophisticated simulation tools and infrastructure can be expensive.

Tools and Resources

For more information and tools that support LVC and its benefits for the DT&E of autonomous systems, see:

- Planning for LVC Simulation Experiments (Haase et al. 2014).
- Joint Test and Evaluation Methodology Overview (Bjorkman 2007).

Challenges Addressed by This Method

LVC testing helps to address several challenges for the T&E of autonomous systems including:

- **T&E as a Continuum**. LVC testing provides an integrated and iterative approach to testing across different phases of development and operational evaluation, ensuring that testing is not limited to isolated events but continuously informs system improvement.
- **T&E of the OODA Loop**. LVC enables evaluation of the autonomous systems' ability to observe, orient, decide, and act in dynamic environments by simulating realistic scenarios and stimuli.
- **Infrastructure**. LVC can reduce the reliance on physical infrastructure for testing, enabling a more efficient and cost-effective evaluation of autonomous systems.

- **Personnel**. LVC allows the testing of complex capabilities with minimum resources, which include personnel support and expertise.
- **Exploitable Vulnerabilities**. LVC facilitates the identification and mitigation of vulnerabilities in autonomous systems by enabling comprehensive testing in diverse scenarios, including adversarial attacks.
- **Safety**. LVC provides a safe and controlled environment for testing potentially hazardous autonomous systems, mitigating risks to personnel and equipment.
- **Ethics**. LVC allows for the examination of ethical considerations, such as bias, decision-making transparency, and compliance with rules of engagement, helping to shape ethical guidelines for autonomous system deployment.
- **HAT**. LVC supports the evaluation of human-machine interactions and the development of effective HAT strategies.
- **Black Box Components**. LVC can help to understand and evaluate the behavior of black box components within autonomous systems by observing their interactions with other system elements in various scenarios.
- **Mission Evolution**. LVC testing enables systems to be efficiently tested under various mission profiles and conditions, ensuring adaptability as mission requirements evolve.
- **Dynamic Learning**. LVC allows for controlled evaluations of autonomous systems with learning components, ensuring that adaptive behaviors align with mission objectives and do not introduce unintended consequences.
- **T&E Adequacy and Coverage**. LVC facilitates comprehensive testing across a wide range of scenarios, improving the adequacy and coverage of T&E efforts.
- **Autonomy Integration and Interoperability**. LVC supports the evaluation of how well autonomous systems integrate and interoperate with other systems in a complex environment.

5.3.2 Experimentation Test and Evaluation

Experimentation T&E is becoming more common across DoD as prototyping and experimentation proliferate. For autonomous systems, it emphasizes real-world data collection and analysis to drive system development and validation, particularly for systems with emergent behaviors operating in complex environments. Experimentation T&E is not about certification or evaluating knowns; it is a process of discovery, exploring unknowns and informing future decisions.

Description of Experimentation T&E

Experimentation T&E:

- Is an iterative process of designing, executing, and analyzing experiments in operationally relevant conditions to assess early capabilities and limitations of autonomous systems.
- Exposes the system to a range of scenarios, including edge and corner cases, to understand its performance and to uncover unexpected behaviors and vulnerabilities.
- Does not aim to confirm specifications—unlike traditional T&E—but to reveal unknowns and inform future development.

Details and Best Practices

Key features of experimentation T&E for autonomous systems include:

- Scenario-based testing: Develop diverse and representative scenarios that cover the expected operational domain, including nominal, off-nominal, and adversarial conditions.
- Data-driven analysis: Utilize comprehensive data collection and analysis techniques to evaluate system performance, identify failure modes, and track progress over time. Types of data include sensor data, internal system states, and performance metrics.
- Iterative refinement: Employ a continuous loop of testing, analysis, and refinement to progressively improve system capabilities and address identified shortcomings, allowing for adaptive test strategies and efficient use of resources.
- M&S: Leverage simulations and modeling to complement real-world testing, explore a broader range of scenarios, and reduce reliance on expensive or dangerous physical tests.
- Collaboration: Foster collaboration between testers, developers, and operational users to ensure that experiments are relevant, informative, and aligned with user needs in defining objectives and interpreting results.
- Flexibility and rigor: Maintain test flexibility to accommodate evolving requirements while maintaining sufficient rigor to answer the experiment's core questions.
- Acceptance of technical risk: Acknowledge and document the higher technical risks often associated with experimentation involving unproven technologies and methodologies, informing future experiments and decision-making.
- Test planning: Develop test plans that focus on exploring ideas and specific technology employment rather than just requirements. Traditional test plans are often requirements based; in experimentation, requirements may be fluid or nonexistent. Use objectives such

as “Explore the …” or “Perform the … experiment.” Success criteria should be flexible, and evaluation criteria may evolve during the experiment.

- Technical reporting: Focus reporting on revealing the experiment methodology, answering the experiment question, and increasing knowledge. Traditional rating scales may need careful tailoring. Recommendations may include continued research, lessons learned, and partnering with OT to bridge the gap between experimentation and acquisition.

Primary Outcome and Additional Benefits

The primary outcome and additional benefits of experimentation T&E for autonomous systems include the following:

- Primary outcome:
  - Informed decision-making: Provides decision-makers with data-driven insights into the system’s capabilities, limitations, and potential for operational use to support decisions on future development, acquisition, and deployment.
- Additional benefits:
  - Early identification of issues: Uncovers design flaws, performance limitations, and safety concerns early in the development life cycle.
  - Accelerated development cycle: Facilitates rapid iteration and learning by providing continuous feedback on system performance.
  - Enhanced user understanding: Improves user understanding of the system’s capabilities and limitations, leading to more effective training and HMT.
  - Increased transparency and trust: Builds trust in the system’s capabilities by providing stakeholders with clear and objective evidence of the system’s performance.

Costs, Limitations, and Assumptions

The use of experimentation T&E may have the following negative impacts:

- Resource intensiveness: This method requires significant investments in time, personnel, test infrastructure, and data analysis capabilities.
- Test environment limitations: The creation of realistic and representative test environments can be challenging.

- Data bias: Test data may be biased or incomplete because of experimentation with limited conditions, potentially leading to inaccurate conclusions.

Tools and Resources

For more information and tools that support experimentation T&E and its benefits for the DT&E of autonomous systems, see the DoD Experimentation Guidebook.

Challenges Addressed by This Method

Experimentation T&E helps to address several challenges for the T&E of autonomous systems including:

- **T&E as a Continuum**. Experimentation T&E allows for iterative learning and refinement throughout the system life cycle, helping adapt to evolving requirements and technology.
- **Data**. Experimentation T&E generates data to inform effective data management and analysis techniques, which can be reused in more comprehensive follow-on tests.
- **HAT**. Experimentation T&E facilitates the evaluation of human-machine interactions in operationally relevant contexts and provides insights into effective teaming strategies and potential challenges.
- **Test Adequacy and Coverage**. Experimentation T&E expands the breadth of testing beyond predefined requirements by exposing the system to a wider range of scenarios, including edge cases, emergent behaviors, and unforeseen conditions.
- **Autonomy and Interoperability**. Experimentation T&E supports the assessment of how autonomous systems interact with other autonomous and nonautonomous platforms in a joint operational environment, identifying potential gaps in integration and coordination.

5.3.3 Surrogate Platforms

A valuable approach for the T&E of autonomous systems is the use of surrogate platforms. This method employs stand-ins to represent the actual system or environment, facilitating comprehensive testing in a controlled and safe setting.

Description of Surrogate Platforms

Surrogate platforms for the T&E of autonomous systems involve the use of substitute systems, simulations, or environments in place of the actual autonomous system or its intended

operational environment, allowing testers to assess and refine autonomous system capabilities before testing with new assets.

Details and Best Practices

Key features of surrogate platforms for the T&E of autonomous systems include:

- Early problem identification. By utilizing surrogate platforms, the potential design flaws, safety concerns, and performance limitations can be identified and addressed early in the development life cycle.
- Flexible system representation. A surrogate can be chosen or modified to meet the unique needs of the autonomy software. A mature surrogate allows for testing the autonomy software independent from the integrated operational system.
- Reduced risk and increased data quality. Well-characterized surrogates with known dynamics, payloads, instrumentation, and interfaces support safer, cheaper, faster, and more effective development; mitigate platform risk; and lead to reduced integration costs and increased speed, ultimately yielding higher-quality data.
- Standardized testbeds. Utilizing standardized government-owned testbeds and test surrogates that are well-characterized, are highly available, and include instrumentation can streamline the T&E process.
- Common interfaces. Common interfaces are crucial for "plug-and-play" compatibility, enabling seamless transitions between platforms to demonstrate increasing maturity and provide a streamlined risk mitigation ramp from low-cost to high-cost surrogates.
- Scalable complexity. Autonomy can be stimulated using state data from a simple foam remote-controlled hobby aircraft or a complex platform such as a QF-16. Effective use of surrogates could involve a progression from simulated physics models to low-cost small unmanned aircraft systems (such as the RQ-23A TigerShark), and then to complex fighter surrogates (such as the X-62A VISTA), before final integration with the operational platform.

Primary Outcomes and Additional Benefits

The primary outcomes and additional benefits of surrogate platforms for the T&E of autonomous systems include the following:

- Primary outcomes:

  - Enhanced system reliability and safety. Early surrogate testing leads to improved system robustness, reduces unexpected behaviors, and minimizes the potential risks associated with autonomous systems T&E.
  - Accelerated development cycles. Early identification of issues through surrogate testing allows for the rapid iteration and refinement of system design and capabilities.
- Additional benefits:
  - Reduced development costs. Surrogate platforms reduce the need for extensive field testing and minimize potential damage to expensive prototypes.
  - Facilitated collaboration. Surrogate platforms provide a common framework for developers, testers, and stakeholders to collaborate and evaluate system performance.
  - Enhanced understanding of system behavior. Detailed data analysis from surrogate testing provides valuable insights into system behavior, aiding in the refinement of algorithms and decision-making processes.

Costs, Limitations, and Assumptions

The use of surrogate platforms may have the following negative impacts:

- Simulation fidelity. The accuracy and realism of surrogate platforms compared with the intended final platform may not support universally meaningful test results.
- Initial investment. Developing and maintaining sophisticated surrogate platforms can require significant upfront investment.
- Technical expertise. Implementing and utilizing surrogate platforms effectively necessitates specialized expertise in those platforms and their limitations.

Tools and Resources

Future updates to this guidebook will provide additional information and tools that support surrogate platforms and their benefits for the DT&E of autonomous systems, including resource information from DoD test centers on available test surrogate platforms.

Challenges Addressed by This Method

Surrogate platforms help to address several challenges for the T&E of autonomous systems including:

- **T&E as a Continuum**. Surrogate platforms support a continuous T&E process, allowing for iterative T&E throughout the development life cycle.

- **T&E of the OODA Loop**. Surrogate platforms allow evaluation of the observe, orient, decide, and act process in realistic but controlled environments, ensuring that autonomous decision-making can be assessed before deployment.
- **Personnel**. Surrogate platforms reduce the burden on personnel by allowing for early autonomy evaluations in testbeds, decreasing the reliance on operators and warfighters for live testing.
- **Safety**. Surrogate platforms enable safe and controlled testing of autonomous systems with potentially unexpected behaviors, minimizing the risks to personnel and equipment.
- **HAT**. Surrogate platforms facilitate the evaluation of HAT concepts, allowing for the assessment of collaboration and interaction between human operators and autonomous systems.

5.3.4 Formal Verification Methods

Formal verification is a mathematically rigorous technique used to prove or disprove the correctness of a system's design with respect to a certain formal specification or property. This method is particularly relevant for autonomous systems where safety and reliability are paramount.

Description of Formal Verification

Formal verification involves the use of mathematical techniques to prove the correctness of a system's design.

- Exhaustive analysis. Unlike traditional testing that relies on sampling system behaviors, formal verification aims to analyze all possible states and transitions within a system.
- Mathematical proof. Formal verification provides a mathematical proof that the system will behave as intended under all circumstances defined by the specification.

Details and Best Practices

Key features of formal verification for the T&E of autonomous systems include:

- Model checking: Building a finite model of the system and using automated tools to check whether the model satisfies desired properties. Model checking is particularly useful for verifying safety-critical aspects of autonomous behavior.
- Theorem proving: Using interactive software tools to construct a mathematical proof that the system design adheres to its formal specifications. This approach allows for the verification of complex systems and intricate properties.

- Static analysis: Employing techniques to analyze the system's code or design without executing it. Static analysis can help identify potential issues such as deadlocks, race conditions, or buffer overflows early in the development cycle.
- Runtime verification: Monitoring the system's behavior during operation to ensure that it conforms to the specified properties. Runtime verification complements other formal verification techniques by providing real-time assurance.
- Contributing evidence: Using formal verification to contribute to the evidence needed for autonomous system assurance, along with T&E results, as part of a larger assurance case argument (see Section 5.2.7 for more information on assurance cases).

Primary Outcomes and Additional Benefits

The primary outcomes and additional benefits of formal verification for the T&E of autonomous systems include the following:

- Primary outcomes:
  - Increased confidence: Providing a high level of assurance that the system behaves as intended, minimizing the risk of unexpected or hazardous actions.
  - Early detection of defects: Identifying design flaws and potential errors in the early stages of development, reducing the cost and effort of fixing them later.
- Additional benefits:
  - Improved system safety: Rigorously verifying safety-critical aspects, leading to more dependable and trustworthy autonomous systems.
  - Reduced development costs: Detecting errors early to significantly reduce the costs associated with debugging and rework.
  - Enhanced system reliability: Ensuring the system's consistent and predictable behavior, even in complex and unforeseen situations.
  - Facilitated certification: Providing evidence of compliance with safety standards and regulations, easing the certification process.
  - Improved communication: Using formal specifications to serve as a precise and unambiguous means of communication between stakeholders.
  - Increased trust: Providing verifiable guarantees about the system's behavior, increasing user trust and acceptance.

Costs, Limitations, and Assumptions

The use of formal verification may have the following negative impacts:

- High initial investment: Requires specialized expertise and tools, which can be expensive to acquire and maintain.
- Scalability challenges: Can become computationally expensive for very large and complex systems.
- Applicability limitations: May not be suitable for all aspects of autonomous systems, particularly those involving complex interactions with the physical world.

Tools and Resources

For more information and tools that support formal verification and its benefits for the DT&E of autonomous systems, see the Defense Advanced Research Projects Agency (DARPA) published research on the High-Assurance Cyber Military Systems (HACMS) Website (https://www.darpa.mil/research/programs/high-assurance-cyber-military-systems).

Challenges Addressed by This Method

Formal verification helps to address several challenges for the T&E of autonomous systems including:

- **Requirements**. Formal verification precisely captures and analyzes system requirements, ensuring that they are complete, consistent, and unambiguous. This approach reduces the risk of misinterpretations and errors arising from ambiguous or incomplete requirements.
- **Exploitable Vulnerabilities**. Formal verification supports the mitigation of security risks by proving high assurance of consistent and correct behavior.
- **Safety**. Formal verification provides strong guarantees about the system's safety by mathematically proving its adherence to safety requirements, mitigating the risk of accidents and malfunctions.
- **Test Adequacy and Coverage**. Formal verification complements traditional testing by providing a more exhaustive analysis of the system's behavior. It helps identify potential issues that might be missed by test cases, improving test coverage and increasing confidence in the system's reliability.

5.3.5 Adversarial Testing

Adversarial testing uses simulated adversary forces and AI to identify system vulnerabilities and assess potential impacts. This method helps ensure that autonomous systems are resilient against threats and adaptable to hostile environments.

Description of Adversarial Testing

Adversarial testing involves simulating adversarial conditions to understand system vulnerabilities:

- Adversarial AI and simulated threats probe weaknesses and identify failure points.
- Impact analysis evaluates the consequences of potential system breaches or malfunctions to understand operational risks.

Details and Best Practices

Key features of adversarial testing for the T&E of autonomous systems include the following:

- Cyclical vulnerability testing routinely evaluates system and attack surfaces using diverse attack vectors, adapting as system insights evolve.
- Simulated adversary scenarios implement realistic adversary AI to replicate potential threats, assessing system responses to a range of hostile conditions.
- Layered defense validation verifies the effectiveness of built-in defenses and countermeasures, ensuring system resilience against multiple types of attacks.

Primary Outcomes and Additional Benefits

The primary outcomes and additional benefits of adversarial testing for the T&E of autonomous systems include the following:

- Primary outcomes:
  - Enhanced system resilience and performance in contested environments with unpredictable adversaries.
  - Identification and mitigation of vulnerabilities, strengthening system readiness for real-world threats.
- Additional benefits:
  - Improved mission-based risk assessment by understanding the operational impact of system weaknesses against a full spectrum of threats.

- Support for mission planning by revealing likely adversary tactics and response strategies.
- Contributions to iterative system improvement by documenting adversarial interactions and system responses.
- Early identification of necessary system upgrades or modifications to counter potential threats.

Costs, Limitations, and Assumptions

The use of adversarial testing may have the following negative impacts or trade-offs:

- Assumptions about adversary behavior that may not fully represent real-world threats, impacting test effectiveness.
- Cost of infrastructure and personnel to design, manage, and maintain adversarial testing assets and tools.
- Potential for increased system wear and stress from repeated adversarial scenarios, which may require additional systems for testing.

Tools and Resources

For more information and tools that support adversarial testing and its benefits for the DT&E of autonomous systems, see the technical paper, "Simulation-based Adversarial Test Generation for Autonomous Vehicles with Machine Learning Components" (Tuncali et al. 2018).

Challenges Addressed by This Method

Adversarial testing helps to address several challenges for the T&E of autonomous systems including:

- **T&E of the OODA Loop**. Adversarial testing enables insights into decision-making speed and adaptability under adversarial conditions.
- **Exploitable Vulnerabilities**. Adversarial testing identifies potential weak points by probing system defenses and responses to hostile forces.
- **Black Box Components**. Adversarial testing evaluates the reliability and resilience of system outputs and responses when internal processes are obscured, ensuring robust performance despite limited visibility.
- **Mission Evaluation**. Adversarial testing evaluates system performance in simulated contested environments to ensure mission success under real-world threat scenarios.

- **Dynamic Learning**. Adversarial testing supports iterative improvements by documenting adversary interactions, enhancing the system's adaptability over time.

5.3.6 Post-Acceptance Testing

Future autonomous systems will likely require post-acceptance testing in operationally relevant environments after the system has been accepted for fielding. The complexity of the systems' use cases often makes complete testing impossible before deployment, and many high-consequence failures occur at very low frequencies. This fact necessitates a shift from the traditional DoD T&E paradigm of separate developmental and operational testing to a continuous evaluation process throughout the system life cycle.

Description of Post-Acceptance Testing

Post-acceptance testing involves operational realistic data-driven assessments, including red teaming, of the autonomous system after it has been fielded, supplemented by continuous monitoring to assess its performance and identify any deficiencies that may not have been apparent during earlier testing phases.

Details and Best Practices

Key features of post-acceptance testing for the T&E of autonomous systems include:

- Continuous monitoring: Implementing continuous monitoring of the fielded system's performance using data logging, telemetry, and user feedback to allow for the identification of emerging issues and long-term performance trends.
- Simulated operational scenarios: Conducting regular testing in realistic operational scenarios, including edge cases and challenging environments, to assess the system's resilience and adaptability.
- Red teaming: Employing dedicated red teams to actively try to exploit vulnerabilities or induce failures in the fielded autonomous system, providing valuable insights into potential weaknesses.
- Data-driven assessment: Utilizing the data collected during OT to perform detailed analysis of the system's performance, enabling data-driven assessments and improvements.

Primary Outcome and Additional Benefits

The primary outcome and additional benefits of post-acceptance testing for the T&E of autonomous systems include the following:

- Primary outcome:
  - Enhanced system trustworthiness. Identifying and mitigating issues that emerge only in real-world operational environments leads to significant improvements in the reliability and safety of the autonomous system.
- Additional benefits:
  - Increased user confidence. Demonstrating the system's capabilities in realistic operational settings fosters trust and confidence among users and other stakeholders.
  - Improved operational effectiveness. Continuous evaluation and refinement based on real-world data enhance the system's overall operational effectiveness.
  - Accelerated technology maturation. Testing in operational environments provides valuable feedback for future development cycles, accelerating the maturation of autonomous system technologies.
  - Reduced life cycle costs. Early identification and resolution of issues in the field can reduce costly maintenance and upgrades down the line.
  - Enhanced training and doctrine development. Data and insights gained from OT can be used to inform and improve training programs and operational doctrine.

Costs, Limitations, and Assumptions

The use of post-acceptance testing may have the following negative impacts:

- Resource intensiveness. This method requires the ongoing commitment of resources, including personnel, infrastructure, and data analysis capabilities.
- Potential for operational disruption. Testing in operational environments may temporarily disrupt normal operations.
- Data security and privacy concerns. Collecting and analyzing data from fielded systems raises concerns about data security and privacy that must be addressed.

Tools and Resources

For more information and tools that support post-acceptance testing and its benefits for the DT&E of autonomous systems, see the DAU Post-Implementation Review Website.(https://www.dau.edu/glossary/post-implementation-review).

Challenges Addressed by This Method

Post-acceptance testing helps to address several challenges for the T&E of autonomous systems including:

- **T&E as a Continuum**. Post-acceptance testing reinforces the concept of T&E as a continuous process that extends beyond initial development and fielding, ensuring ongoing evaluation and improvement throughout the system's life cycle.
- **T&E of the OODA Loop**. Post-acceptance testing enables real-world evaluation of how the system processes observations, makes decisions, and takes actions in dynamic operational environments.
- **Personnel**. Post-acceptance testing assesses operator interactions, workload, and HAT effectiveness in operational settings, refining training and doctrine.
- **Exploitable Vulnerabilities**. Post-acceptance testing identifies previously unknown vulnerabilities that may emerge only in real-world use, enabling mitigation strategies to be implemented proactively.
- **Ethics**. Post-acceptance testing provides ongoing evaluation of ethical considerations, such as unintended biases in decision-making, compliance with rules of engagement, and impact on human oversight.
- **Black Box Components**. Post-acceptance testing monitors and analyzes system behavior in operational environments to uncover hidden dependencies, emergent behaviors, and decision-making anomalies.
- **Mission Evolution**. Post-acceptance testing allows for evaluation of the system's ability to adapt to evolving mission requirements and changing operational contexts.
- **Dynamic Learning**. Continuous monitoring and evaluation during post-acceptance testing enable dynamic learning from real-world performance, facilitating ongoing adaptation and improvement of the autonomous system.
- **Test Adequacy and Coverage**. Testing in operational environments with diverse scenarios and edge cases enhances test coverage and helps ensure the system's adequacy for its intended mission.

5.4 Test Planning

Test planning for autonomous systems can be very challenging. Effective test planning is incredibly valuable if it can address challenges such as safety, security, and human teaming, while speeding the development and fielding of highly capable systems. This section discusses

several practices related to test planning for an autonomous system, which help to enable effective, efficient, and robust T&E:

- AI model testing and metrics.
- STPA for autonomy.
- HAT performance methods and measures.
- Automatic domain randomization.
- Automated outlier search and boundary testing.
- Failure path testing.

These practices may not apply to every autonomy program, but where implemented, they help enable successful T&E of autonomous systems with reduced costs and time.

5.4.1 Artificial Intelligence Model Testing and Metrics

AI models are increasingly critical to the operation of autonomous systems, enabling capabilities such as perception, planning, and decision-making. The T&E methods and metrics for testing AI models are different from those used for traditional system components in many ways. AI model testing and metrics should inform and complement fully integrated autonomous systems T&E.

Description of AI Model Testing and Metrics

The AI model testing and metrics method is a systematic approach to evaluating the performance of AI models used in autonomous systems that includes designing specific test cases, collecting performance data, and applying relevant metrics to assess the model's effectiveness and identify areas for improvement. It emphasizes continuous testing and monitoring throughout the system's life cycle to account for the inherent uncertainties and complexities associated with AI.

Details and Best Practices

Key features of AI model testing and metrics for the T&E of autonomous systems include:

- Data management for AI model development and testing: Ensure that the AI test team appropriately partitions the dataset used for ML in accordance with AI T&E best practices.
    - Training data: Utilize large, diverse, and representative datasets for training AI models. Ensure data quality, address biases, and document data sources and preprocessing steps.

  - Test data: Employ separate test datasets that are independent of the training data to evaluate the model's generalization ability to unseen data. Regularly update test data to reflect evolving operational conditions and potential threats.
  - Validation data: Utilize a validation dataset to fine-tune model hyperparameters and prevent overfitting to the training data. Validation tuning helps ensure that the model's performance generalizes well to new data.
- Model-in-the-loop simulation: Evaluate the AI model's performance by testing in a controlled setting within a simulated environment before hardware integration.
- Scenario-based testing: Create realistic scenarios to assess the model's robustness to identify potential failures and limitations.
- Performance metrics: Utilize quantitative measures such as accuracy, precision, recall, F1-score, and latency to assess specific aspects of the AI model's capabilities, providing objective evidence of the model's effectiveness.
- Data-driven evaluation: Employ diverse and representative datasets, encompassing various operational conditions, environments, and potential biases, to ensure comprehensive testing.
- Test case design: Develop test cases that cover both nominal and off-nominal scenarios, including unexpected inputs, sensor failures, and adversarial attacks, to evaluate the model's resilience.
- Continuous monitoring: Implement ongoing monitoring and evaluation of the AI model's performance during development, testing, and deployment to track progress, identify regressions, and inform retraining efforts. Continuous monitoring is crucial because of the evolving nature of AI models and the impossibility of exhaustive pre-deployment testing.
- Documentation and reporting: Maintain detailed records of test procedures, datasets, metrics, and results to ensure traceability, reproducibility, and accountability.
- Formalized risk acceptance: Clearly document risk tolerance and specify acceptable ranges of behavior before deployment. This approach helps manage evaluation expectations and supports informed decisions about autonomous system deployment when communicated to users and decision-makers.
- Ontological standards: Utilize ontological standards, such as those provided by IEEE, to formalize assumptions and deployment conditions. Standards ensure a common understanding of the system's operating environment and interactions.

Primary Outcome and Additional Benefits

The primary outcome and additional benefits of AI model testing and metrics for the T&E of autonomous systems include the following:

- Primary outcome:
  - Evidence supporting the integrated system's test and use: Providing objective evidence of the AI model's effectiveness and robustness, supporting informed decision-making regarding the integrated autonomous system's evaluation, deployment, and operational use.
- Additional benefits:
  - Enhanced system reliability: Identifying and mitigating potential failures or limitations in the AI model's performance, leading to the improved safety and dependability of the autonomous system.
  - Accelerated development cycles: Enabling early detection of defects and performance bottlenecks, facilitating the rapid iteration and refinement of the AI model.
  - Reduced development costs: Minimizing the risk of costly failures or rework by identifying and addressing issues early in the development process.
  - Improved system performance: Optimizing the AI model's performance through data-driven insights and iterative refinement, leading to enhanced capabilities and efficiency.
  - Facilitated regulatory compliance: Providing evidence of compliance with safety and performance standards, supporting certification and approval processes.

Costs, Limitations, and Assumptions

The use of AI model testing and metrics may have the following negative impacts:

- Resource intensiveness. This method requires significant computational resources, data, and expertise to design and execute comprehensive test campaigns.
- Test coverage limitations. It may be challenging to achieve complete test coverage because of the complexity of real-world scenarios and the potential for unforeseen events.
- Metric selection challenges. Choosing appropriate metrics that accurately reflect the desired capabilities and performance criteria can be complex and context dependent.

Tools and Resources

For more information and tools that support AI model testing and metrics and their benefits for the DT&E of autonomous systems, see:

- DT&E of AI-Enabled Systems Guidebook.
- DARPA Assured Autonomy Website (https://www.darpa.mil/program/assured-autonomy).
- IEEE Standards Association Website (https://standards.ieee.org/).

Challenges Addressed by This Method

AI model testing and metrics help to address several challenges for the T&E of autonomous systems including:

- **T&E as a Continuum**. AI model testing and metrics support an iterative and continuous testing process, ensuring that AI models are evaluated throughout their life cycle and continuously refined as new data and operational conditions emerge.
- **T&E of the OODA Loop**. AI model testing and metrics provide evidence for evaluating the performance of AI models within the OODA loop, evaluating how the system can effectively perceive, process information, and make decisions in dynamic environments.
- **Personnel**. This method ensures that AI testing teams are equipped with the appropriate tools and methodologies to evaluate AI models, reducing the reliance on highly specialized expertise while improving training for AI T&E practitioners.
- **Exploitable Vulnerabilities**. AI model testing and metrics identify weaknesses in AI models, such as susceptibility to adversarial attacks, bias, and data poisoning, allowing mitigation strategies to be developed before deployment.
- **Black Box Components**. AI model testing and metrics promote the use of explainability and interpretability techniques to understand and evaluate the reasoning process of AI models, even when their internal workings are not fully transparent.
- **Dynamic Learning**. AI model testing and metrics support the evaluation of AI models that continuously learn and adapt, ensuring that their performance remains reliable and safe as they encounter new data and situations.

5.4.2 System-Theoretic Process Analysis for Autonomy

STPA is a hazard analysis method grounded in systems theory. STPA is based on the Systems-Theoretic Accident Model and Processes (STAMP), which is a modern accident causation model

that views safety as a dynamic control problem. STPA uses a top-down approach for analysis and delivers qualitative results that can be used to guide the design of today's complex sociotechnical systems, including autonomous systems.

Description of STPA

The STPA Handbook (Leveson and Thomas 2018) describes the four steps in applying STPA, as shown in Figure 5-8.

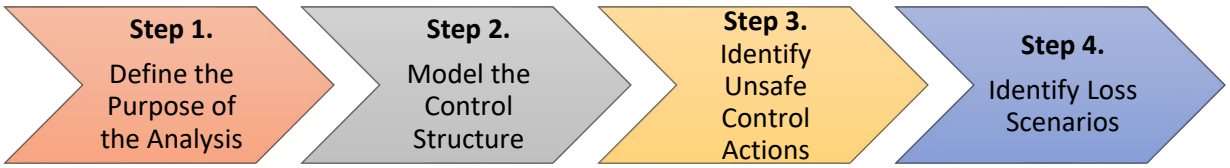


Figure 5-8. STPA Steps

Step 1: Define the Purpose of the Analysis. Identify the stakeholder losses, system-level hazards, and corresponding system-level constraints. Loss is anything of value to a stakeholder including mission-, safety-, security-, and resilience-related losses.

Step 2: Model the Control Structure. Model the as-is or to-be hierarchical control structure, which is composed of feedback and control loops needed to ensure that hazardous states are avoided. As shown in Figure 5-9, the control structure is a graphical depiction of the system components (controllers and controlled processes) and the interactions between them in terms of control actions, feedback, and other interactions such as coordination with another controller.

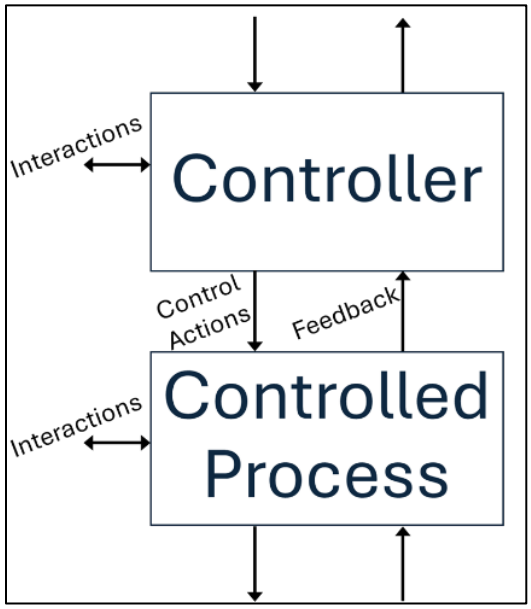


Figure 5-9. Generic Hierarchical Control Structure

Step 3: Identify Unsafe Control Actions. Given the control structure, Step 3 identifies unsafe control actions (UCAs) that can lead to hazardous states. Four UCA categories may lead to hazards: (1) not providing the control; (2) providing the control; (3) providing the control too early, too late, or in the wrong order; and (4) providing the control too long or too short. Step 3 also identifies the UCA's corresponding controller-level constraints.

Step 4: Identify Loss Scenarios. The practitioner identifies loss scenarios, or causal factors, that may lead to UCAs and hazardous states. This step benefits from having a deeper understanding of the system under investigation and system component interactions and provides a more detailed abstraction level than Step 3. The causal scenarios can be used to influence design and describe constraints needed to ensure desired system behavior.

Details and Best Practices

A case study of STPA (Bowers and Thomas 2023) in the developmental flight test phase demonstrated the use and benefits of applying STPA before flight testing a neural network-controlled uncrewed air vehicle.

- A team of experts conducted STPA after mandated airworthiness and safety processes but before actual flight test to ensure safe flight test execution.
- STPA findings from examining the human-autonomy system include the following:
  - Autonomy multi-axis flight control inputs may lead to hazardous scenarios.
  - The unmanned system envelope protection was inadequate.
  - Handoff procedures between human and autonomy were in some cases ambiguous.
  - Safety transition maneuver from autonomy to human may not be so safe.
- In summary, the team uncovered an additional 50 safety-critical issues that led to processing 49 changes to the flight test procedures.

Primary Outcome and Additional Benefits

The primary outcome and additional benefits of STPA for the T&E of autonomous systems include the following:

- Primary outcome:
  - Risk management. STPA is a top-down approach for analysis with qualitative results that can guide the design and testing of autonomous systems, managing and mitigating risks.

- Additional benefits:
  - Flexible applications. STPA is well-suited for the analysis of humans, software, human-machine interactions, and organizational influences.
  - Multidimensional analysis. STPA can identify safety concerns from flawed design and component behaviors, in addition to component failures.

Costs, Limitations, and Assumptions

The use of STPA may have the following negative impacts or trade-offs:

- Because of the qualitative nature of STPA, there may be a wide discrepancy in the quality of results, influenced by STPA methodology expertise and domain expert participation.
- Risk in terms of probability and severity is not a direct output of STPA, though risk assessments can be integrated into or derived from STPA results.

Tools and Resources

For more information and tools that support STPA and its benefits for the DT&E of autonomous systems, see:

- STPA Handbook (Leveson and Thomas 2018).
- Massachusetts Institute of Technology (MIT) Partnership for Systems Approaches to Safety and Security (PSASS) Website (https://psas.scripts.mit.edu/home/), which is an online repository for STPA-related material.
- Additional STPA guidance for autonomy teaming and coordination:
  - Systems-Theoretic Safety Analyses Extended for Coordination (Johnson 2017).
  - System-Theoretic Safety Analysis for Teams of Collaborative Controllers (Kopeikin 2024).
- Guidance for the security of autonomy:
  - Basic Introduction to STPA for Security (Young 2020).

Challenges Addressed by This Method

STPA helps to address several challenges for the T&E of autonomous systems including:

- **Requirements**. STPA identifies gaps or ambiguities that could lead to unsafe behaviors in autonomous systems.

- **Infrastructure**. STPA supports the structured analysis of complex system architectures, ensuring that control structures, communication pathways, and interdependencies are well understood to improve test infrastructure design.
- **Exploitable Vulnerabilities**. STPA identifies systemic weaknesses and potential failure modes that could be exploited, ensuring that both security and safety concerns are addressed holistically.
- **Safety**. STPA provides a top-down approach to hazard analysis, identifying UCAs and causal scenarios that traditional risk-based methods may overlook.
- **HAT**. STPA evaluates human-autonomy interactions, uncovering potential mismatches in expectations, ambiguous control transitions, and risks arising from human oversight or intervention.
- **Test Adequacy and Integration**. STPA ensures that testing covers not only component-level failures but also emergent risks stemming from system interactions, providing a more comprehensive evaluation of autonomous system behaviors.
- **Autonomy Integration and Interoperability**. STPA assesses how autonomous systems coordinate within larger SoS environments, ensuring that safety constraints and control dependencies are effectively managed across multiple agents.

5.4.3 Human-Autonomy Team Performance Methods and Measures

HAT performance methods and measures are essential for assessing and improving the mission effectiveness of autonomous systems. These methods focus on evaluating key factors between humans and autonomy, such as situational awareness, role clarity, communication, and collaboration to ensure effective and reliable team dynamics.

Description of HAT Performance Methods and Measures

HAT performance methods and measures include:

- Role clarity and control allocation methods to establish clear, testable requirements and guidelines for authority, responsibility, and handoff protocols between human operations and autonomous systems to reduce risk and increase mission success.
- Situational awareness measures of perception, comprehension, and projection of future states to evaluate how well human and autonomous team members perceive and interpret the situation, enabling improved projection of and response to future states.
- Collaboration and communication measures to evaluate how timely, necessary information is shared and to encourage proactive and responsible behaviors in both

human and autonomous systems, fostering coordinated actions and maintaining shared goals.

Details and Best Practices

Key features of HAT methods for the T&E of autonomous systems include:

- Leveraging mission engineering: Provide the scope of the human factors and HSI evaluations necessary to measure the contributions to mission outcomes for total system effectiveness at a mission level.
- Employing standardized metrics for situational awareness and role clarity: Utilize established T&E metrics to assess operator(s) and system(s) perception, comprehension, and projection within their environment, supporting consistency and comparability across evaluations.
- Conducting controlled testing of human-system communication and intent: Employ specific measures for observing implicit and explicit communication cues (e.g., gestures, speech, visual indicators) to evaluate how effectively intent is communicated between human and autonomous team members.
- Developing adaptive testing frameworks for varying operator states: Incorporate adaptive metrics that account for human factors such as stress, workload, and fatigue, ensuring that HAT performance remains stable across different operator conditions.
- Tailoring testing based on human interaction levels: Use HITL, HOTL, and HOOTL system testing, understanding that the systems differ in their best testing practices.
    - HITL system testing should focus on evaluating the user interface, ensuring that it effectively communicates the system's recommendations and allows humans to make informed decisions.
    - HOTL system testing should include assessing the system's ability to detect anomalies and alert humans, as well as evaluating the human's ability to respond effectively to these alerts.
    - HOOTL system test activities center on verifying the system's ability to operate reliably, within the scope of the human's commanded intent, and make accurate decisions without human intervention.
- Understanding human decision-making: Develop a deep understanding of how humans make decisions in complex, dynamic environments and how autonomy agents can support or hinder this process.

- Assessing human-autonomy collaboration: Evaluate the effectiveness of human-autonomy collaboration, including the exchange of information, coordination of actions, and resolution of conflicts or errors. Useful test metrics may include accuracy, response time, and degree of meeting the commander's intent or human instructions.
- Evaluating human performance in context: Develop metrics and methods to assess human performance within the context of HMT, considering factors such as cognitive load, situational awareness, and decision-making.
- Applying human-centered T&E methods: Prioritize the human element of performance contribution, incorporating human-centered evaluation objectives.

Primary Outcome and Additional Benefits

The primary outcome and additional benefits of HAT methods for the T&E of autonomous systems include the following:

- Primary outcome:
  - Evaluation of trustworthiness: Characterization of human-autonomy interactions, enabling appropriately calibrated trust and cohesion between human operators and autonomous systems, improving team effectiveness and mission outcomes by establishing reliable, transparent interactions.
- Additional benefits:
  - Risk management: Reduced operational risk through rigorous assessment of shared situational awareness, role clarity, and control protocols, leading to more predictable and safe autonomous behaviors.
  - Operational assurance: Improved adaptability and resilience by enabling systems to respond effectively to human states, such as stress and workload, ensuring stable performance across diverse conditions.

Costs, Limitations, and Assumptions

The use of HAT may have the following negative impacts or trade-offs:

- Evaluating factors such as communication cues, trust calibration, and role transitions in HAT introduces additional complexity, increasing the time and resources needed for thorough analysis.

- HAT methods rely on assumptions regarding predictable and generalizable human responses and stable systems behaviors, which may not hold true in dynamic or stress-intensive environments, potentially impacting test reliability.
- Significant investment is required to validate and verify situational awareness, role clarity, and trust metrics within HAT, necessitating specialized personnel and infrastructure.
- Test environments and scenarios may differ based on human involvement in various tasks, with HITL and HOTL often requiring simulated or controlled environments, whereas HOOTL systems may require more realistic and dynamic testing scenarios for robustness in operational contexts.
- Testing of autonomous systems with different levels of human interaction demands a deep understanding of human factors, cognitive biases, attention, and decision-making processes.

Tools and Resources

For more information and tools that support HAT performance methods and measures and their benefits for the DT&E of autonomous systems, see:

- Scientific Measurement of Situation Awareness in Operational Testing (Green et al. 2023).
- Communicating Intent to Develop Shared Situation Awareness and Engender Trust in Human-Agent Teams (Schaefer et al. 2017).
- Trust Measurement in Human-Autonomy Teams: Development of a Conceptual Toolkit (Krausman et al. 2022).

Challenges Addressed by This Method

HAT performance methods and measures help to address several challenges for the T&E of autonomous systems including:

- **T&E as a Continuum**. HAT performance methods and measures enable continuous assessment and reuse of T&E tools to measure control, situational awareness, and collaboration throughout the life cycle.
- **T&E of the OODA Loop**. HAT performance methods and measures ensure effective human insight into decision-making processes and projected future states for autonomous systems.

- **Requirements**. This method simplifies and standardizes the testing of HAT-related metrics such as situational awareness and role clarity in human-autonomy interactions.
- **Infrastructure**. This method supports the development of test environments and interfaces that facilitate HAT performance evaluation and human-system collaboration.
- **Personnel**. This method employs tailored HAT measures to provide test personnel with clarity and rigor in autonomy test strategy, planning, and analysis.
- **Safety**. HAT performance methods and measures establish and evaluate standardized protocols for safe and predictable interactions between human operators and autonomous systems.
- **Ethics**. HAT performance methods and measures support transparent and ethical interactions by providing measurable indicators for system awareness, predictability, and other HAT objectives.
- **HAT**. HAT performance methods and measures define and evaluate standardized roles, authority delegation, and expectations to enhance cohesion and performance.
- **Dynamic Learning**. This method provides measures of ongoing learning and HAT team effectiveness as operator experience and system capabilities evolve.
- **Autonomy Integration and Interoperability**. HAT performance methods and measures provide and evaluate standardized interfaces and protocols to ensure seamless integration and effective interaction across diverse autonomous systems and human teams.

5.4.4 Automatic Domain Randomization

A method used in training ML models is automatic domain randomization. This method can also enhance the T&E of autonomous systems by automatically generating variations that help provide diverse and challenging test scenarios, which strengthen the system's robustness and readiness for real-world deployment.

Description of Automatic Domain Randomization

Automatic domain randomization leverages algorithms to automatically create variations in a test or simulation environment, including alterations to environmental conditions, sensor parameters, and the physical characteristics of objects and surroundings (location, speed/vector, size, etc.). This technique exposes the autonomous system to a wide array of conditions, significantly exceeding the practical limits of manual scenario generation. Automatic domain randomization incorporates both environment and agent parameter randomization, providing a comprehensive approach to testing.

Details and Best Practices

Key features of automatic domain randomization for the T&E of autonomous systems include:

- Automated scenario generation. Automatic domain randomization algorithms automatically generate numerous test scenarios, saving time and resources compared with manual creation.
- Diversity of scenarios. Automatic domain randomization produces a wide range of scenarios, including edge cases and unusual situations, which may be overlooked in traditional testing.
- Fine-grained control. Automatic domain randomization allows for the introduction of minor variations in input conditions, thereby mitigating unintended overtraining and promoting the generalization of real-world disturbances.
- Dual parameter randomization. Automatic domain randomization includes both environment randomized parameters (e.g., lighting, weather, terrain) and agent randomized parameters (e.g., sensor noise, actuator limitations), providing a more holistic evaluation.
- Customization. Testers can define the parameters and randomization ranges to focus on specific aspects of the system's performance.
- Scalability. Automatic domain randomization can readily scale to create complex scenarios for sophisticated autonomous systems.
- Reproducibility. The process can be easily replicated, ensuring consistency and facilitating regression testing.

Primary Outcome and Additional Benefits

The primary outcome and additional benefits of automatic domain randomization for the T&E of autonomous systems include the following:

- Primary outcome:
  - Enhanced T&E robustness. By exposing the system to diverse and challenging scenarios, including minor variations and randomized agent parameters, automatic domain randomization increases T&E insight to evaluate the system's ability to perform reliably in unpredictable real-world environments.
- Additional benefits:

- Improved trustworthiness. Rigorous testing under varied conditions helps identify and mitigate potential failures and assess system robustness, leading to more trustworthy system deployment.
- Accelerated development. Automatic domain randomization can speed up the testing process, enabling faster iteration and refinement of autonomous systems.
- Reduced costs. Automation reduces the need for manual scenario creation and execution, leading to cost savings.
- Improved performance. Exposure to diverse scenarios can lead to improved system performance and generalization.
- Facilitated data collection. Automatic domain randomization generates vast amounts of data that are valuable for training and refining ML models within autonomous systems.

Costs, Limitations, and Assumptions

The use of automatic domain randomization may have the following negative impacts:

- Computational resources. Running complex simulations with varied parameters can demand significant computational power.
- Bias in randomization. Care must be taken to ensure that the randomization process avoids unintended biases that may skew the test results.
- Definition of realistic parameters. Setting appropriate ranges for randomization requires domain expertise to ensure that scenarios remain relevant and representative of real-world conditions.

Tools and Resources

For more information and tools that support automatic domain randomization and its benefits for the DT&E of autonomous systems, see the technical paper, “Domain Randomization for Transferring Deep Neural Networks from Simulation to the Real World” (Tobin et al. 2017).

Challenges Addressed by This Method

Automatic domain randomization helps to address several challenges for the T&E of autonomous systems including:

- **T&E as a Continuum**. Automatic domain randomization supports continuous testing by enabling iterative and automated scenario variation, ensuring that the system is regularly exposed to diverse and evolving test conditions throughout its life cycle.
- **T&E of the OODA Loop**. Automatic domain randomization assists in evaluating the OODA loop by presenting the autonomous system with diverse and unexpected situations, forcing it to adapt and respond effectively.
- **Safety**. Automatic domain randomization helps to identify potential safety hazards and ensure that the system behaves reliably in challenging scenarios that better reflect the complexity and unpredictability of real-world environments, including edge cases.
- **Data**. Automatic domain randomization generates vast amounts of data that can be used to train, validate, and improve the performance of autonomous systems, especially those relying on ML.
- **Black Box Components**. By introducing minor variations and generalizing real-world disturbances, automatic domain randomization helps prevent overfitting and improves the system's ability to generalize to new situations to evaluate the performance and robustness of black box components within autonomous systems.
- **Test Adequacy and Coverage**. Automatic domain randomization enhances test coverage by automatically creating numerous and diverse scenarios, including those that may be overlooked in traditional testing approaches.

5.4.5 Automated Outlier Search and Boundary Testing

Automated outlier search and boundary testing for models of autonomous systems refers to a process to identify where model behavior is at or near the limits of its operating conditions or exhibits changing performance. These conditions can be related to environmental factors, sensor performance, decision-making capabilities, or other constraints that the system is designed to handle. These regions are also important because they can identify critical areas for real-world testing for validation.

Description of Automated Outlier Search and Boundary Testing

Automated outlier search and boundary testing encompasses the following areas:

- Outlier detection refers to identifying data points that deviate significantly from the expected behavior or patterns within a dataset. These outliers could represent errors, anomalies, or situations that may require special attention.

- Outlier detection tools typically employ statistical methods such as z-scores, standard deviation thresholds, or percentile-based methods to identify values that lie far from the mean or median.
- Boundary search tools focus on determining where the output of a model is rapidly changing. This approach is particularly important for ensuring that the system does not operate in unsafe conditions or locations.
- Statistical methods generally focus on detecting areas of high variability, gradients, or local deviations in the output.

Details and Best Practices

Key features of automated outlier search and boundary testing for the T&E of autonomous systems include the following:

- Effectively covers large test spaces to focus in on regions of interest.
- Determines focus areas for real-world testing.
- Helps evaluate the robustness of the model.
- Gives insight into how systems may react to edge cases.

Primary Outcome

The primary outcome of automated outlier search and boundary testing for the T&E of autonomous systems is that testers can efficiently characterize the safe and effective regions of autonomous system operation within the full operating environment.

Costs, Limitations, and Assumptions

The use of automated outlier search and boundary testing may have the following negative impacts or trade-offs:

- Requires sufficiently representative models and simulations.
- Incurs additional costs for integrating models with statistical tools.

Tools and Resources

For more information and tools that support automated outlier search and boundary testing and its benefits for the DT&E of autonomous systems, see:

- Range Adversarial Planning Tool described in the journal article, "Delivering Test and Evaluation Tools for Autonomous Unmanned Vehicles to the Fleet" (Mullins et al. 2017).
- JMP BEAST Mode: Boundary Exploration through Adaptive Sampling Techniques (Wisnowski et al. 2020).
- MARGInS: Model-based Analysis of Realizable Goals in Systems (Davies et al. 2014).

Challenges Addressed by This Method

Automated outlier search and boundary testing helps to address several challenges for the T&E of autonomous systems including:

- **Safety**. Automated outlier search and boundary testing helps identify potential areas of unsafe operation.
- **Black Box Components**. Even if the AI/ML is not fully understood, automated outlier search and boundary testing can effectively explore the space.
- **Test Adequacy and Integration**. Automated outlier search and boundary testing efficiently covers large test spaces to look for areas of interest.

5.4.6 Failure Path Testing

Failure path testing is a testing technique focused on identifying and analyzing the potential paths where a system may fail under specific conditions. Autonomous systems' high dependence on complex software creates a need for testing of the many potential ways that software faults, bugs, or poor designs could cause unexpected system failures or deficiencies.

Description of Failure Path Testing

Failure path testing encompasses the following:

- A "failure path" refers to a sequence of events or operations in a system that could lead to an error or undesired outcome. These paths are often considered during negative testing, where the system is intentionally subjected to invalid, unexpected, or boundary inputs to see how it behaves.
- This technique is particularly useful in identifying edge cases, vulnerabilities, or areas where the system may not behave as expected when subjected to failure scenarios. The primary goal is to test the system's ability to handle errors or unexpected situations gracefully, ensuring that it does not fail catastrophically and that appropriate error-handling mechanisms are in place.

Details and Best Practices

Key features of failure path testing for the T&E of autonomous systems include:

- Testing for failure conditions. The focus is on testing or simulating conditions that may lead to failure by defining clear test scenarios to understand potential failure modes, such as:
  - Invalid input or incorrect user actions such as fuzz testing.
  - Network failures or slow responses.
  - System crashes or resource limitations (e.g., memory, CPU).
  - Database connection failures.
  - Boundary condition violations (e.g., overflow or underflow).
  - Concurrent access or race conditions.
  - Security vulnerabilities.
- Error handling verification. A key concern during failure path testing is ensuring that when failures occur, the system handles them in a controlled way. This approach includes fuzz testing to evaluate:
  - Appropriate error messages.
  - Logging and reporting errors.
  - Recovery mechanisms (e.g., retries, fallback options).
  - Graceful degradation of system functionality.
- Automated failure path testing. This methodology involves test automation and regression testing:
  - Test automation: Automate failure path tests wherever possible to ensure coverage for negative scenarios across various conditions (e.g., invalid inputs, network failures).
  - Regression testing: Include negative test cases in the regression suite to verify that failures are consistently handled after new code changes.
- Result monitoring and review. These processes include failure reporting and retrospective reviews:
  - Failure reporting: Ensure that failures are well-documented, with clear information on what caused each failure, and any steps needed to reproduce it.

- Retrospective reviews: Regularly review failures that have been discovered during testing to see if any patterns emerge and whether any systemic issues need to be addressed.

Primary Outcome and Additional Benefits

The primary outcome and additional benefits of failure path testing for the T&E of autonomous systems include the following:

- Primary outcome:
  - Robustness and resilience: Verifies the software's ability to recover from errors and continue operating or fail in a predictable and safe manner.
- Additional benefits:
  - Fault tolerance: Ensures that the system can tolerate common faults without compromising user experience or system integrity.
  - Security: Tests for vulnerabilities and ensures that the system does not expose sensitive data or fail in a way that could compromise security.
  - Usability: Ensures that users are presented with understandable error messages and have clear guidance when things go wrong.

Costs, Limitations, and Assumptions

The use of failure path testing may have the following negative impacts or trade-offs:

- Complexity. Mapping out all possible failure paths can be challenging because of the large number of scenarios that can lead to failure, especially in complex systems.
- Resource intensiveness. Simulating failure conditions may require special test environments or configurations (e.g., simulated network failure or resource exhaustion).
- False positives/negatives. The testing may sometimes result in false alarms (where a failure path is incorrectly identified) or miss potential failure points that do not immediately manifest under test conditions.

Tools and Resources

For more information and tools that support failure path testing and its benefits for the DT&E of autonomous systems, see the book, *Software Engineering: A Practitioner's Approach* (Pressman and Maxim 2020).

Challenges Addressed by This Method

Failure path testing helps to address several challenges for the T&E of autonomous systems including:

- **Safety**. Failure path testing helps identify potential areas of unsafe operation.
- **Test Adequacy and Integration**. With automation, failure path testing examines large test spaces to look for areas of interest.

5.5 Test Execution

Test execution provides multiple challenges with autonomous systems but provides great opportunity for learning as well as evaluation. This section discusses several practices related to test execution for an autonomous system, which help to enable effective, efficient, and robust T&E:

- Cognitive instrumentation.
- Runtime assurance.
- Test user interface.

These practices may not apply to every autonomy program, but where implemented, they help enable successful T&E of autonomous systems with reduced costs and time.

5.5.1 Cognitive Instrumentation

An emerging solution for the evaluation of autonomous systems is cognitive instrumentation, which focuses on understanding and assessing the cognitive processes of autonomous systems, ensuring their reliable and predictable operation in real-world scenarios. By providing insights into the "why" behind an autonomous system's actions, cognitive instrumentation enables testers to diagnose the root causes of performance deficiencies and ensure dependable operation.

Description of Cognitive Instrumentation

Cognitive instrumentation is a method to gain insight into the internal state and decision-making processes of autonomous systems. Imagine being able to see inside the "mind" of an autonomous system. Cognitive instrumentation makes this possible by monitoring and analyzing data related to perception, reasoning, and planning, helping testers understand why a system behaves in a particular way. This "internal workings" refers to the machine's ability to perceive, reason, decide, and team in its dynamic OODA loop.

Details and Best Practices

Key features of cognitive instrumentation for the T&E of autonomous systems include:

- Embedded system instrumentation: Integrate specialized tools within the autonomous system to capture real-time data during both testing and operation. This instrumentation allows for the distinction between coding errors, inadequate algorithms, insufficient training data, or even sensor/hardware problems.
- Data acquisition: Capture data from the autonomous system's sensors, processors, and algorithms. This data may involve logging sensor readings, internal representations of the environment, and decision-making pathways.
- Visualization and analysis: Develop tools and techniques to visualize and analyze the acquired data. These techniques could include creating graphical representations of the system's internal state, highlighting areas of uncertainty, and identifying potential biases in decision-making.
- Experimentation and manipulation: Design specific test scenarios and manipulate environmental factors to observe how the autonomous system responds. This approach helps in evaluating the robustness and adaptability of the system's cognitive processes.

Primary Outcomes and Additional Benefits

The primary outcome and additional benefits of cognitive instrumentation for the T&E of autonomous systems include the following:

- Primary outcome:
  - Performing autonomy evaluation: Gain a deeper understanding of the cognitive processes within autonomous systems, enabling more effective evaluation of their performance and safety. This evaluation includes diagnosing the causes of incorrect behavior or inadequate performance by tracing issues back to their origin within the system's perception, reasoning, or decision-making processes.
  - Characterizing reliability and trustworthiness: Identify and enable the mitigation of potential vulnerabilities in the system's decision-making.
- Additional benefits:
  - Enabling early issue detection: Identify potential problems in the design and development phase, reducing the risk of costly failures later in the life cycle.
  - Supporting performance optimization: Fine-tune algorithms and improve the system's overall performance by analyzing cognitive data.

- Enhancing human-machine collaboration: Foster better understanding between humans and autonomous systems, facilitating smoother interaction and collaboration.
- Informing regulatory frameworks: Provide valuable data for developing safety standards and regulations for autonomous systems.

Costs, Limitations, and Assumptions

The use of cognitive instrumentation may have the following negative impacts:

- Increased complexity. Implementing cognitive instrumentation can add complexity to the T&E process, requiring specialized tools and expertise.
- Data overload. The volume of data generated may be overwhelming, necessitating efficient data management and analysis techniques.
- Insufficient explainability. Cognitive instrumentation may not be enough to provide understandability of the system in some cases, especially for AI components, so explainable artificial intelligence (XAI) tools or other evaluations may still be needed.
- Design and implementation costs. Developing and integrating the internal system framework required for cognitive instrumentation can incur significant costs, particularly in terms of specialized engineering and software development.

Tools and Resources

Currently, resources are lacking for autonomous system cognitive instrumentation; however, many of the same issues and terms are discussed for XAI methods. See the technical paper, "Explainable Artificial Intelligence (XAI): Concepts, Taxonomies, Opportunities and Challenges toward Responsible AI" (Arrieta et al. 2019).

Challenges Addressed by This Method

Cognitive instrumentation helps to address several challenges for the T&E of autonomous systems including:

- **T&E as a Continuum**. Cognitive instrumentation enables continuous monitoring and evaluation of cognitive processes throughout the system life cycle, improving transparency and adaptability.
- **T&E of the OODA Loop**. Cognitive instrumentation provides insights into each stage of the OODA loop.

- **Personnel**. Cognitive instrumentation supports operators and analysts by improving system interpretability, reducing cognitive load, and informing the training requirements for effective human-machine collaboration.
- **Exploitable Vulnerabilities**. Cognitive instrumentation identifies weaknesses in the system's cognitive processing that could be exploited, enabling proactive risk mitigation strategies.
- **Ethics**. Cognitive instrumentation promotes the ethical development and deployment of autonomous systems by increasing transparency and accountability.
- **Data**. Cognitive instrumentation provides valuable data for understanding the behavior of autonomous systems and informing the development of more robust and reliable AI algorithms.
- **HAT**. Cognitive instrumentation enhances trust and collaboration by making the system's reasoning and intent more understandable to human operators.
- **Black Box Components**. Cognitive instrumentation mitigates the black box problem by providing visibility into the internal workings of AI algorithms and their decision-making processes.
- **Mission Evolution**. Cognitive instrumentation ensures that the system's decision-making processes remain effective as mission requirements and operational environments evolve.
- **Dynamic Learning**. Cognitive instrumentation facilitates the evaluation of how autonomous systems learn and adapt over time, ensuring their continuous improvement and safe operation in dynamic environments.

5.5.2 Runtime Assurance

A prominent method for the T&E of autonomous systems is runtime assurance, which focuses on monitoring and verifying system behavior during operation. Runtime assurance focuses on real-time monitoring and intervention capabilities to ensure safe and reliable system behavior, especially during complex and unpredictable testing scenarios.

Description of Runtime Assurance

Runtime assurance is a continuous process of monitoring an autonomous system's performance, detecting anomalies, and initiating appropriate responses to maintain safe and effective operation. It acts as a deterministic "wrapper" around the autonomy under test, with the authority to intervene and guide the system to a fail-safe condition if necessary. Runtime assurance allows for the safe exploration of complex autonomous behaviors without the risk of catastrophic failures.

- Real-time system health verification. Runtime assurance focuses on evaluating the system's internal state and external performance during operation to ensure that it functions within defined parameters and safety limits.
- Dynamic adaptation. Runtime assurance allows for adjustments to the system's behavior based on real-time feedback and changing environmental conditions.
- Automated safety and security. Runtime assurance leverages monitoring and feedback tools to manage risk, allowing for graceful degradation of system capabilities in response to anomalies and automating safety and security reporting.

Details and Best Practices

Key features of runtime assurance for the T&E of autonomous systems include:

- Continuous monitoring: Employing sensors, data logging, and analysis techniques to track key performance indicators (KPIs) and system health metrics in real time. This approach includes monitoring both the software and hardware components.
- Anomaly detection: Implementing algorithms and mechanisms to identify deviations from expected behavior, potential failures, or unsafe conditions. This approach may involve ML techniques for pattern recognition and predictive analysis.
- Response mechanisms: Developing and integrating procedures to mitigate or recover from detected anomalies. Response mechanisms can range from simple alerts to complex autonomous recovery maneuvers or safe shutdown procedures.
- Data recording and analysis: Providing comprehensive data logging of system performance, events, and anomalies for post-mission analysis, fault diagnosis, and future system improvement.
- Safe recovery: Providing a reliable backup mechanism for the safe recovery of the test vehicle if unanticipated problems occur, allowing for quick takeovers for restarts and continued testing.
- Algorithm agnosticism: Enabling the test of any algorithm that meets its interface requirements, regardless of complexity, allowing for flexibility in evaluating different autonomous behaviors.
- Failure detection and recovery: Reliably detecting problems (hardware, software, or environmental) and switching to a recovery/safe mode in the event of a failure.

Primary Outcomes and Additional Benefits

The primary outcomes and additional benefits of runtime assurance for the T&E of autonomous systems include the following:

- Primary outcomes:
  - Enhanced system safety. By continuously monitoring system health and implementing response mechanisms, runtime assurance aims to minimize the risk of accidents or unintended consequences during autonomous operation, especially during testing.
  - Increased system reliability. Through real-time anomaly detection and mitigation, runtime assurance helps to ensure the consistent and dependable performance of autonomous systems in various operational scenarios.
- Additional benefits:
  - Improved performance optimization. Real-time data analysis can be used to fine-tune system parameters and optimize performance for specific tasks or environments.
  - Accelerated testing cycles. Continuous monitoring and automated anomaly detection can expedite the identification of system weaknesses, leading to faster iterative testing and development cycles.
  - Increased user confidence. Demonstrating robust runtime assurance capabilities can build trust in the safety and reliability of autonomous systems, facilitating their wider adoption.
  - Increased test safety. Runtime assurance allows more aggressive exploratory testing through increased test safety confidence.
  - Reduced development costs. Early detection of anomalies during testing can prevent costly failures and rework later in the development life cycle.
  - Enhanced data-driven decision-making. The rich data collected through runtime assurance provides valuable insights for system design improvements, operational planning, and maintenance scheduling.
  - Improved support for certification and accreditation. Runtime assurance data can be used to demonstrate compliance with safety standards and regulations, aiding in the certification and accreditation of autonomous systems.
  - Enhanced test security. Runtime assurance adds another layer of defense against adversarial actions by providing a counter to corrupted code, verifying commands, and providing redundant control.

Costs, Limitations, and Assumptions

The use of runtime assurance may have the following negative impacts:

- Increased system complexity. Implementing comprehensive monitoring and response mechanisms can add complexity to the system design and software development.
- Computational overhead. Using real-time data processing and analysis may require significant computational resources, potentially impacting system performance.
- Challenges in defining safety boundaries. Determining and characterizing the multidimensional safety boundary and interrogating it in real time is a significant challenge.
- Runtime overhead. Observing an executing system typically incurs some runtime overhead. It is important to minimize this overhead, particularly when monitors are deployed with the system.
- Additional V&V requirements. Implementing runtime assurance introduces additional V&V requirements for the autonomy program.

Tools and Resources

For more information and tools that support runtime assurance and its benefits for the DT&E of autonomous systems, see:

- Safe Testing of Autonomy in Complex, Interactive Environments described in the technical paper, “Safe Testing of Autonomous Systems Performance” (Scheidt et al. 2015).
- R2U2: Tool Overview (Rozier and Schumann 2017).

Challenges Addressed by This Method

Runtime assurance helps to address several challenges for the T&E of autonomous systems including:

- **T&E as a Continuum**. Runtime assurance supports the concept of dTEaaC, allowing for consistent monitoring and evaluation of system performance throughout development and even during deployment.
- **Infrastructure**. Reusable runtime assurance systems can streamline testing infrastructure needs across multiple programs.

- **Personnel**. Runtime assurance automates aspects of testing, potentially reducing the need for large testing teams and specialized expertise.
- **Exploitable Vulnerabilities**. Runtime assurance helps mitigate potential vulnerabilities in the system by providing backup and fail-safe control, improving security.
- **Safety**. Runtime assurance is crucial for ensuring the safety of autonomous systems, especially during testing when unexpected behaviors may emerge.
- **Ethics**. Runtime assurance ensures the ethical deployment of autonomous systems by enforcing predefined safety constraints and preventing unintended harmful actions.
- **HAT**. Effective runtime assurance can increase human confidence and trust that the system will not engage in unsafe or dangerous behavior.
- **Black Box Components**. Even with complex black box components within the autonomous system, runtime assurance can monitor overall system behavior and identify anomalies.
- **Mission Evolution**. Runtime assurance supports the adaptation of autonomous systems to new mission parameters by ensuring continuous compliance with safety and operational constraints.
- **Dynamic Learning**. Runtime assurance provides the ability to detect and respond to anomalies in dynamically learning systems.
- **Test Adequacy and Coverage**. Runtime assurance expands the testing scope by enabling the safe exploration of edge cases, ensuring broader test coverage without increased risk.
- **Autonomy Integration and Interoperability**. Runtime assurance facilitates integration with other autonomous and human-operated systems by ensuring consistent safety enforcement across multiple platforms.

5.5.3 Test User Interface

Among the many humans interacting with autonomous systems, test personnel can sometimes be overlooked. Test user interfaces are important for the T&E of autonomous systems, allowing for the controlled T&E of the system's performance and trustworthiness under various conditions and mission scenarios.

Description of Test User Interface

The test user interface:

- Provides testers with tools to interact with, manipulate, and evaluate the autonomous system in a safe and repeatable environment.
- Allows for the injection of various scenarios, environmental conditions, and system disturbances to assess the system's robustness and ability to handle unexpected situations.
- Enables the collection of valuable data on system performance, human-machine interactions, and even operator workload for tasks involving human partners.

Details and Best Practices

Key features of test user interfaces for the T&E of autonomous systems include:

- Scenario generation. The test user interface should allow testers to create and execute various operational scenarios, including normal operations, unexpected events, and emergency situations, to enable assessment of the autonomous system's performance under diverse conditions.
- System manipulation. The test user interface should enable testers to manipulate the autonomous system's state, inputs, and outputs, by adjusting sensor readings, injecting software faults, or overriding control algorithms to test the system's resilience and safety mechanisms.
- Data acquisition and analysis. The test user interface should be equipped with robust data logging capabilities to capture relevant information during testing, including cognitive instrumentation data, to support performance analysis, identify areas for improvement, and validate system requirements.
- Operator controls. The test user interface should provide appropriate controls and displays for human operators to interact with the autonomous system, including setting mission parameters, monitoring system status, and assuming manual control when necessary.
- Real-time feedback. The test user interface should provide real-time feedback to testers and operators on the system's performance, behavior, and responses to various stimuli, which is crucial for understanding the system's capabilities and limitations.
- Safety and security features. The test user interface should incorporate safety and security features to protect the test team and equipment, which could include emergency stop buttons, system interlocks, and secure access controls.
- Insightful data visualization. The test user interface should present data in a clear and concise manner, allowing testers to gain insights into the autonomous system's

characteristics and behavior, which could involve using graphs, charts, and other visualizations to highlight KPIs and trends.

Primary Outcomes and Additional Benefits

The primary outcomes and additional benefits of test user interfaces for the T&E of autonomous systems include the following:

- Primary outcomes:
  - Enhanced test value and safety. By enabling rigorous testing under various conditions, the test user interface helps identify potential issues and vulnerabilities early in the development cycle, leading to more reliable and safer autonomous systems.
  - Improved HAT. The test user interface facilitates the evaluation of human-autonomy interactions, leading to better understanding of operator workload, situational awareness, and trust in the autonomous system, which could also inform the design of more effective human-machine interfaces for optimal collaboration.
- Additional benefits:
  - Reduced development costs. Early identification of issues through the test user interface can reduce costly rework and redesign later in the development process.
  - Accelerated testing cycles. The test user interface enables efficient and repeatable testing, facilitating faster iteration and evaluation of system updates and modifications.
  - Better understanding of system behavior. The test user interface provides valuable insights into the autonomous system's decision-making processes, responses to stimuli, and overall behavior.
  - Improved training effectiveness. The test user interface can be used for operator training, allowing operators to familiarize themselves with the system's capabilities and limitations in a safe and controlled environment.

Costs, Limitations, and Assumptions

The use of test user interfaces may have the following negative impacts:

- Development cost. Designing and implementing a comprehensive test user interface can be expensive, requiring specialized expertise and resources.

- Complexity. The test user interface can be complex to develop and maintain, especially for highly sophisticated autonomous systems.
- Limited realism. Test injects may not be able to fully capture the complexities and uncertainties of real-world environments.

Tools and Resources

For more information and tools that support the test user interface and its benefits for the DT&E of autonomous systems, see:

- 10 Usability Heuristics for User Interface Design (Nielsen 2024).
- System Usability Scale on the Test Science Measuring Usability Website (https://testscience.org/measuring-usability/).

Challenges Addressed by This Method

Test user interfaces help to address several challenges for the T&E of autonomous systems including:

- **T&E of the OODA Loop**. Test user interfaces enable testers to evaluate the performance of the OODA loop within the autonomous system.
- **Data**. Test user interfaces facilitate the collection and analysis of data from autonomous system testing, enabling the identification of performance issues, trends, and areas for improvement.
- **HAT**. Test user interfaces facilitate an effective interface for test team personnel and improved insight into the collaboration between humans and autonomous systems.
- **Autonomy Integration and Interoperability**. Test user interfaces support the evaluation of autonomous system interoperability with other platforms by allowing test teams to simulate interactions, mission parameters, and cross-system communication.

5.6 Data Analysis and Evaluation

Data analysis and evaluation of autonomous systems is critical to iteratively improve and expand testing and provide valuable insights to achieve justified evidence of trustworthiness and inform data-driven decisions. This section discusses several practices related to an autonomous system's test data analysis and evaluation, which help to enable the effective, efficient, and robust T&E of autonomous systems as well as certification and accreditation:

- Human performance standards.

- Operational and mission-based testing.
- Task-based certification.
- Quantified risks and autonomy performance growth curves.

These practices may not apply to every autonomy program, but where implemented, they help enable successful T&E of autonomous systems with reduced costs and time.

5.6.1 Human Performance Standards

Human (operator) performance standards involve applying specific measures of performance, suitability, and effectiveness based on established training and proficiency standards. These standards ensure that autonomous systems can achieve mission effectiveness by meeting or exceeding human performance baselines, enabling reliable interaction with human operators.

Description of Human Performance Standards

Human performance standards encompass the following areas:

- Proficiency standards set clear benchmarks for human performance, providing a baseline for evaluating autonomous system capabilities in similar mission scenarios.
- Task performance metrics measure human effectiveness across key tasks, guiding the development of autonomous systems that can reliably perform these tasks.
- Suitability measures assess human ability to complete tasks within operational contexts, helping to define the level of reliability and effectiveness required from autonomous systems.

Details and Best Practices

Key features of human performance standards for the T&E of autonomous systems include:

- Established human performance standards to set baselines for autonomous system capabilities, ensuring these systems meet mission effectiveness requirements.
- Continuous proficiency assessments to monitor operator skills and adapt autonomous system development and testing as mission needs evolve.
- Task-specific performance metrics to define KPIs based on human proficiency, helping autonomous systems achieve comparable or superior effectiveness in mission scenarios.
- Scenario-based training integration to align operator standards with real-world mission conditions, enhancing readiness and system reliability.

Primary Outcome and Additional Benefits

The primary outcome and additional benefits of human performance standards for the T&E of autonomous systems include the following:

- Primary outcome:
  - Enhanced T&E insights into autonomous system perception, decision-making, learning, teaming, and control capabilities, benchmarked against proven, measurable standards of human operator performance.
- Additional benefits:
  - Improved alignment of human capabilities with system requirements, ensuring optimal operator–system interaction.
  - Enhanced design and development processes by using human performance as a reference, ensuring that autonomous systems meet mission-critical task needs.
  - Early identification of gaps in autonomous system functionality, allowing for system adjustments or training modifications.
  - Increased system reliability by benchmarking key operator tasks.

Costs, Limitations, and Assumptions

The use of human performance standards may have the following negative impacts or trade-offs:

- Assumption that existing human operator standards exist and are relevant and applicable to autonomous system interactions, which may not always be the case.
- Limitation that some subjective or nonquantitative performance standards require expert interpretation, potentially impacting consistency and objectivity in evaluations.
- Potential difficulty in adapting human performance metrics to novel or rapidly evolving autonomous system capabilities.

Tools and Resources

Future updates to this guidebook will include tools for human performance standards, which are currently in development.

Challenges Addressed by This Method

Human performance standards help to address several challenges for the T&E of autonomous systems including:

- **T&E of the OODA Loop**. Human performance standards support the evaluation of how human operators interact with autonomous systems in the OODA cycle.
- **Requirements**. Human performance standards ensure the alignment of human performance standards with system requirements to meet mission-specific goals.
- **Personnel**. Human performance standards establish performance benchmarks based on human task execution, helping to define the capabilities that autonomous systems must meet or exceed.
- **Safety**. Human performance standards establish clear performance benchmarks for safe human-system interactions, reducing operational risk.
- **Data**. Human performance standards provide consistent performance benchmarks derived from human task execution, enabling the evaluation of autonomous system capabilities against proven standards.
- **HAT**. Human performance standards establish measures for evaluating and improving collaboration, role allocation, and interaction between humans and autonomous systems.
- **Mission Evolution**. Human performance standards support mission success by confirming operator readiness to manage autonomous systems in evolving mission contexts.
- **Autonomy Integration and Interoperability**. Human performance standards establish human performance benchmarks to guide autonomous system design, ensuring smooth integration.

5.6.2 Task-Based Certification

A method gaining prominence in the T&E of autonomous systems is task-based certification. It offers a structured approach to evaluating system capabilities against specific tasks, ensuring operational effectiveness and safety. It allows for incremental certification as the system matures, mirroring the graded certifications common in human training.

Description of Task-Based Certification

Task-based certification:

- Is a capability-focused assessment method that shifts the focus from traditional pass/fail verification of individual requirements to evaluating the system's ability to perform mission-essential tasks in its intended operational environment, acknowledging the complex and adaptive nature of autonomous systems.

- Is an iterative certification process, similar to how human operators gain certifications incrementally as they progress through training, supporting the certification of autonomous systems for limited operations with specific tasks, with the expectation of expanded capabilities over time.

Details and Best Practices

Key features of task-based certification for the T&E of autonomous systems include:

- Scenario-based testing: Create realistic scenarios that represent the operational tasks the autonomous system will face, testing or simulating various environments, threats, and unexpected events.
- Task decomposition: Break down complex tasks into smaller, manageable subtasks to facilitate focused evaluation and identify specific areas for improvement, reducing risks for complex mission tasks.
- Metrics-driven assessment: Define clear, measurable metrics to evaluate task completion, such as accuracy, efficiency, time to completion, and safety.
- Iterative evaluation: Conduct testing in an iterative manner, allowing for adjustments to the system and scenarios based on the results of previous evaluations, supporting continuous learning and improvement.
- Evolving standards: Allow certification standards to evolve alongside advancements in autonomy capabilities and test methodologies.

Primary Outcome and Additional Benefits

The primary outcome and additional benefits of task-based certification for the T&E of autonomous systems include the following:

- Primary outcome:
  - Establishes certification standards tailored to appropriately match autonomous system performance and trustworthiness for specific missions and tasks, allowing for incremental fielding of capabilities as the system matures.
- Additional benefits:
  - Improves system design. This approach reveals design flaws and areas for improvement early in the development process by focusing on task performance.
  - Reduces development costs. Smaller, task-based capabilities can lead to faster development cycles and lower overall costs.

- Enhances human teaming. The demonstration of system competence through task-based evaluation can increase user coordination and acceptance.
- Streamlines acquisition. This method provides a clear framework for evaluating proposals and selecting the most suitable autonomy capabilities for specific missions.
- Addresses ethical considerations. This method helps assess the ethical implications of autonomous systems by evaluating their behavior in challenging scenarios and withholding certifications for those tasks with insufficient proven trustworthiness.
- Responds to evolving needs. The addition or adjustment of task-based performance supports evolving threats, tactics, and technologies.

Costs, Limitations, and Assumptions

The use of task-based certification may have the following negative impacts:

- Complexity. Developing realistic scenarios and defining appropriate metrics can be complex and time-consuming.
- Resource intensiveness. Implementing this method may require significant resources, including simulation tools, test environments, and subject matter expertise.
- Subjectivity. Evaluating task performance can involve some level of subjectivity, particularly for tasks that require complex decision-making.
- Continuous improvement. Task-based certification assumes that the autonomous system will improve and expand capabilities over time, which may not be necessary for some applications.

Tools and Resources

For more information and tools that support task-based certification and its benefits for the DT&E of autonomous systems, see the task-oriented requirements engineering (TORE) framework in the technical paper, “TORE: A Framework for Systematic Requirements Development in Information Systems” (Adam et al. 2014).

Challenges Addressed by This Method

Task-based certification helps to address several challenges for the T&E of autonomous systems including:

- **T&E as a Continuum**. Task-based certification supports the concept of T&E as a continuous process integrated throughout the system life cycle, from design and development to deployment and sustainment.
- **Requirements**. Task-based certification focuses not on verifying individual requirements but rather on assessing the system's ability to perform mission-essential tasks, ensuring that requirements are relevant and contribute to overall operational effectiveness.
- **Personnel**. Task-based certification enables test personnel to evaluate autonomous systems in a manner similar to human operator qualification, aligning with familiar training and certification models.
- **Safety**. Task-based certification helps to identify potential safety risks and ensure that the system performs tasks safely within its intended operating environment.
- **Ethics**. Task-based certification enables ethical use by evaluating behaviors in challenging task scenarios and withholding certifications for those tasks with insufficient proven trustworthiness.
- **HAT**. Task-based certification helps to evaluate the effectiveness of human-autonomous system collaboration by assessing joint task performance to test and evaluate task integration and cooperation and calibrate trust more appropriately to specific tasks.

5.6.3 Operational and Mission-Based Testing

Operational and mission-based testing focuses on evaluating autonomous systems within realistic mission scenarios, integrating them with other manned and unmanned assets to assess resilience against full-spectrum threats and their collective impact on mission success.

Description of Operational and Mission-Based Testing

Operational and mission-based testing focuses on evaluating autonomous systems in realistic mission settings alongside manned and unmanned assets. This method assesses the system's effectiveness, adaptability, and resilience under collective full-spectrum threats and encompasses the following:

- Integrated mission scenarios to test system performance in coordinated operations with other assets, ensuring interoperability and mission cohesion.
- Threat environment evaluation by simulating full-spectrum threats to analyze the system's response and impact on mission success.
- Operational/mission alignment to ensure that autonomous system capabilities meet the demands of dynamic, real-world mission conditions.

Details and Best Practices

Key features of operational and mission-based testing for the T&E of autonomous systems include the following:

- Mission CONOPS and desirable tactics are defined early in development and incorporated into the design and realization of the autonomous system.
- Scenario-based testing replicates realistic mission conditions, integrating the autonomous system with other assets to assess its interoperability and adaptability.
- Full-spectrum threat assessments are used to test the system's resilience against a range of adversarial conditions, assessing not only individual threats but also the combined effects and interactions among them, providing insights into system reliability and mission readiness.

Primary Outcomes and Additional Benefits

The primary outcome and additional benefits of operational and mission-based testing for the T&E of autonomous systems include the following:

- Primary outcome:
  - Final evaluation to confirm the autonomous system's effectiveness and suitability in an integrated, SoS mission test scenario.
  - Verification of mission readiness by ensuring that the system can operate reliably under realistic conditions alongside both manned and other unmanned assets.
- Additional benefits:
  - Increased confidence in system performance across varied operational environments.
  - Identification of improvements for system interoperability with other assets.
  - Enhanced mission planning capabilities by understanding system behavior in full-spectrum threat scenarios.
  - Support for iterative design improvements through feedback from mission-based testing.
  - Reduced risk of mission failure by identifying and addressing potential vulnerabilities early.
  - Validation of TTPs under realistic conditions to support operational readiness.

Costs, Limitations, and Assumptions

The use of operational and mission-based testing may have the following negative impacts or trade-offs:

- Cost of infrastructure and personnel to plan and execute realistic SoS tests.
- Organizational and planning difficulty with establishing early CONOPS, especially for a technology that is developing rapidly.
- Potential for incomplete threat representation, as some adversarial conditions may be challenging to fully simulate.
- Increased time requirements for coordinating and executing complex, integrated mission scenarios.

Tools and Resources

For more information and tools that support operational and mission-based testing and its benefits for the DT&E of autonomous systems, see the Air Force Test Center Orange Flag Website (https://www.aftc.af.mil/Test-Flag-Enterprise/Orange-Flag/).

Challenges Addressed by This Method

Operational and mission-based testing helps to address several challenges for the T&E of autonomous systems including:

- **Exploitable Vulnerabilities**. Operational and mission-based testing identifies weaknesses in system performance under realistic mission conditions, reducing the likelihood of adversary exploitation.
- **Safety**. Operational and mission-based testing identifies safety risks in operational contexts, especially in scenarios involving both manned and unmanned assets.
- **HAT**. Operational and mission-based testing evaluates human-autonomy collaboration under realistic operational conditions to improve team cohesion and trust.
- **Test Adequacy and Coverage**. Operational and mission-based testing provides thorough testing across mission scenarios to ensure reliable performance across operational contexts.
- **Autonomy Integration and Interoperability**. Operational and mission-based testing verifies seamless integration and cooperative performance with other autonomous and human-operated assets.

5.6.4 Quantified Risks and Autonomy Performance Growth Curves

This method quantifies various types of risks for an autonomous system by using statistical techniques such as reliability growth curves. By measuring relevant metrics over time, testers can use statistics to measure improvement and to make justified predictions of future capabilities.

Description of Quantified Risks and Autonomy Performance Growth Curves

Quantified risks and autonomy performance growth curves:

- Determine relevant metrics of interest:
  - Major failures.
  - Minor failures.
  - Loss of control.
  - Incorrect decisions.
- Measure and plot metrics over time similar to reliability growth curves.
- Use appropriate statistical techniques to evaluate trends over time and predict future performance:
  - Can be used to estimate when a system will reach a suitable level of performance.
  - Can support key program decisions.

Details and Best Practices

Key features of quantified risks and autonomy performance growth curves for the T&E of autonomous systems include:

- Clearly defining goals early.
- Using appropriate models for the data.
- Actively managing corrective actions.
- Ensuring that data quality is sufficient.
- Regularly communicating results to relevant stakeholders.

Primary Outcomes

The primary outcomes of quantified risks and autonomy performance growth curves for the T&E of autonomous systems include:

- Visually demonstrating and predicting performance over time.
- Evaluating the maturity of the system toward a goal.
- Showing when a system is lagging and may need additional efforts to improve performance.

Costs, Limitations, and Assumptions

The use of quantified risks and autonomy performance growth curves may have the following negative impacts or trade-offs:

- Model selection. When using reliability growth curve fitting, it is important to avoid choosing an inappropriate or unsuitable model for the data.
- Parameter estimation. Incorrectly or imprecisely estimating the parameters of the reliability growth curve fitting model can lead to inaccurate or unreliable results.
- Assumption validation. When using reliability growth curve fitting, it is important to avoid violating or ignoring the assumptions of the model or the method.
- Environment. Most models do not account for environmental changes.
- Complexity. Some simple models may not be appropriate for complex systems.

Tools and Resources

For more information about performance growth curves and their benefits for the DT&E of autonomous systems, see:

- DAU Reliability Growth Website (https://www.dau.edu/acquipedia-article/reliability-growth).
- Reliability Growth Guidance in the DOT&E TEMP Guidebook.

Challenges Addressed by This Method

Quantified risks and autonomy performance growth curves help to address several challenges for the T&E of autonomous systems including:

- **T&E as a Continuum**. This method captures the improvement over time of various metrics and estimates future performance.
- **Exploitable Vulnerabilities**. This method identifies recurring failure modes or degradation trends that may represent exploitable weaknesses in system design or behavior.

- **Safety**. This method helps identify trends in failure modes over time, enabling early detection of risk patterns before they become critical.
- **HAT**. This method highlights trends in system behaviors that affect teaming effectiveness, such as decision latency or coordination breakdowns.
- **Test Adequacy and Coverage**. This method uses statistical techniques to evaluate whether enough testing has been performed to support confidence in system maturity and reliability.
- **Autonomy Integration and Interoperability**. This method monitors how autonomous components perform within a larger SoS context, assessing whether integrated performance improves in line with expectations.

6 Test and Evaluation Resources

This section will be published separately as an addendum to follow the basic guidebook.

7 Conclusion

This guidebook has provided focused guidance and recommended practices for the early and developmental T&E of autonomous systems for the purposes of DoD. This guidebook addressed the novel challenges of removing human operators from DoD systems and empowering future autonomous systems, especially those that are AI enabled, to independently act in contested environments. These challenges demand iterative approaches to evaluating the growing capabilities of autonomous systems to ensure trusted mission capability across complex operational environments. The information provided in this guidebook includes:

- Definitions of key terminology in T&E, autonomy, and AI.
- Important U.S. Federal and DoD policies that relate to the T&E of autonomous systems.
- Background on recent and current technology and acquisition developments with major impacts on the future vision for DoD T&E of autonomous systems.
- Overarching and specific challenges that autonomous systems pose for T&E.
- Methods and best practices for autonomous systems that may help reach solutions to those T&E challenges, as well as provide additional benefits to other disciplines.

The guidance includes links and citations to references with more information about methodologies and practices. In future iterations of this guidebook, additional references, resources, tools, and examples will be provided to support the DoD autonomy T&E community. Expansions and improvements are planned on a relatively frequent basis. The authors and sponsors of this guidebook welcome inputs and recommendations from across the community.

In summary, this guidance leverages emerging best practices in agile and iterative testing to extend success throughout the T&E continuum. By applying these best practices to achieve efficient, effective, and robust DT&E, autonomous DoD systems will be primed for successful operational T&E and operational employment.

Glossary

To create meaningful contrasts and standardize concepts, a lexicon must be established. Achieving a common understanding among all English speakers is rare, and it presents an even greater challenge with emergent technologies. Nevertheless, to enable education, learning, and collaboration among DoD T&E, autonomy, and AI organizations, the following lexicon serves as a common reference. Authoritative references within DoD, the U.S. government, and industry were consulted in that order of precedence. These definitions should be considered "in use" definitions for relevant organizations within the autonomy T&E community, rather than immutable facts.

accuracy. When referring to ML, the number of correct classification predictions divided by the total number of predictions. A measure for indicating the overall correctness of a classification model's predictions. (Google Machine Learning Glossary: https://developers.google.com/machine-learning/glossary)

algorithm. A method or set of rules or instructions to be followed in calculations or other problem-solving operations, particularly by a computer. (AI Principles: Recommendations on the Ethical Use of AI by DoD)

algorithmic bias. Systematic bias in an AI system's outputs. Can be due to biased input or training data, a statistically biased estimator in the algorithm, off-label use, incorrect assumptions, or misinterpretation. (AI Principles: Recommendations on the Ethical Use of AI by DoD)

anomaly detection. The identification of rare occurrences, items, or events of concern due to their differing characteristics from majority of the processed data. (DeepAI Glossary: https://deepai.org/machine-learning-glossary-and-terms/anomaly-detection)

artificial intelligence (AI). The ability of machines to perform tasks that normally require human intelligence—for example, recognizing patterns, learning from experience, drawing conclusions, making predictions, or taking action—whether digitally or as the smart software behind autonomous physical systems. (Summary of the 2018 DoD AI Strategy)

autonomous weapon system. A weapon system that, once activated, can select and engage targets without further intervention by an operator. This includes, but is not limited to, operator-supervised autonomous weapon systems that are designed to allow operators to override operation of the weapon system, but can select and engage targets without further operator input after activation. (DoDD 3000.09)

autonomy. The ability of a system to achieve goals while operating independently of external control. Requires self-directedness (to achieve goals) and self-sufficiency (to operate independently). (Fong 2018)

Bayesian learning. The classifiers assume that the probability of the presence or absence of the state of a feature is modified by the states of other features. (Dukart and Hoffmann-La Roche 2015)

black box testing. Testing based on an analysis of the specification of the component or system. (International Software Testing Qualifications Board (ISTQB) Glossary: https://glossary.istqb.org/en/search/)

computer vision. The field of study surrounding how computers see and understand digital images and videos. Computer vision spans all tasks performed by biological vision systems, including "seeing" or sensing a visual stimulus, understanding what is being seen, and extracting complex information into a form that can be used in other processes. (22 Technologies Computer Vision Website: https://22-tech.com/computer-vision/)

cyber-physical systems (CPS). A special case of a cyber-system that interacts with its physical surroundings. A cyber-system that controls and responds to physical entities through actuators and sensors. (Refsdal et al. 2015)

deciding. Selecting a course of action or choosing how to implement an intended course of action. (DAU CLE 002: "Introduction to the Test & Evaluation (T&E) of Autonomous Systems")

decision tree. When referring to ML, a non-parametric supervised learning algorithm, which is utilized for both classification and regression tasks. It has a hierarchical, tree structure, which consists of a root node, branches, internal nodes and leaf nodes. (IBM Website: https://www.ibm.com/think/topics/decision-trees)

deep learning. Multiple layers of neural networks stacked "deep." (AI Principles: Recommendations on the Ethical Use of AI by DoD)

emergent behavior. Coherent patterns of high-level system behavior that would be difficult, if not impossible, to predict from an understanding of the lower-level component behaviors. (DAU CLE 002: "Introduction to the Test & Evaluation (T&E) of Autonomous Systems")

explainable artificial intelligence (XAI). A key term in AI design and in the tech community as a whole. It refers to efforts to make sure that AI programs are transparent in their purposes and how they work. XAI is a common goal and objective for engineers and others trying to move

forward with AI progress. (AI: A Glossary of Terms: https://link.springer.com/content/pdf/bbm%3A978-3-319-94878-2%2F1.pdf)

feature. When referring to ML, an input variable used in making predictions. (Google Machine Learning Glossary: https://developers.google.com/machine-learning/glossary)

formal methods. Mathematical techniques for specification, development, and verification of hardware and software systems. Formal methods typically rely on formal logic, discrete mathematics, or structured specification languages, and can be employed for modeling requirements or for analyzing and mathematically proving specified features of a system.

generative adversarial network. A system to create new data in which a generator creates data and a discriminator determines whether that created data is valid or invalid. (Google Machine Learning Glossary: https://developers.google.com/machine-learning/glossary)

gradient descent. A mathematical technique to minimize loss. Gradient descent iteratively adjusts weights and biases, gradually finding the best combination to minimize loss. (Google Machine Learning Glossary: https://developers.google.com/machine-learning/glossary)

human factors. The application of science and data to understand how human capabilities and limitations interact with other elements of a system.

human in the loop (HITL). A system architecture in which active human judgment and engagement are part of the operation of a system, and a human is an integral part of the system behavior. An example is the human operator of a remotely piloted vehicle or a decision support system that makes recommendations for a human to decide on.

human on the loop (HOTL). A system architecture in which a human has a supervisory role in the operation of the system but is not an integral part of the system behavior. An example is an operator monitoring a fleet of warehouse robots—they operate autonomously but can be shut down if the operator determines that something is wrong.

human out of the loop (HOOTL). A system architecture in which systems are fully automated and do not require any human input or oversight.

inference. When referring to ML, the process of making predictions by applying a trained model to unlabeled examples. In statistics, inference refers to the process of fitting the parameters of a distribution conditioned on some observed data. (Google Machine Learning Glossary: https://developers.google.com/machine-learning/glossary)

label. When referring to ML, specifically supervised learning, the "answer" or "result" portion of an example. Each example in a labeled dataset consists of one or more features and a label. (Google Machine Learning Glossary: https://developers.google.com/machine-learning/glossary)

live, virtual, and constructive (LVC). A taxonomy to broadly classify M&S. (1) Live Simulations, which represent the natural physical environment in which individuals or teams operate their systems and platforms for test or training purposes. (2) Virtual Simulations, which are synthetic environments that include the replication of warfighting equipment and operational environmental conditions; allows for the sharing of a common environment which multiple users can access; and supports interactions with simulated entities (including objects, avatars, and equipment) that mirror, in response fidelity, those that would occur in the real world. (3) Constructive Simulations which are entirely simulated forces. Typically, real human inputs are needed to fully operate these simulated forces which then carry out the resultant actions in a synthetic environment. (Mills 2014)

machine learning (ML). The capability of machines to learn from data without being explicitly programmed. (AI Principles: Recommendations on the Ethical Use of AI by DoD)

metrics. Used to measure the quality of the statistical or ML model. (DeepAI Glossary: https://deepai.org/definitions)

model. When referring to ML, the set of parameters and structure needed for a system to make predictions. (Google Machine Learning Glossary: https://developers.google.com/machine-learning/glossary)

natural language processing (NLP). The use of algorithms to determine properties of natural, human language so that computers can understand what humans have written or said. NLP includes teaching computer systems how to extract data from bodies of written text, translate from one language to another, and recognize printed or handwritten words. (DeepAI Glossary: https://deepai.org/definitions)

neural network or artificial neural network. A computing system made up of a number of simple, highly interconnected processing elements, which process information by their dynamic state response to external inputs. Typically organized in layers of interconnected "nodes" where data inputs are observed in the input layer, then communicated to and processed in one or more hidden layers, to finally link to an output layer. (AI Principles: Recommendations on the Ethical Use of AI by DoD)

offline learning. ML systems that have learned their approximate target functions or policies after initial training phase and no longer learn or are "frozen." (AI Principles: Recommendations on the Ethical Use of AI by DoD)

online learning. ML systems that learn and continue to learn on dynamic inputs in real time. (AI Principles: Recommendations on the Ethical Use of AI by DoD)

overfitting. Creating a model that matches the training data so closely that the model fails to make correct predictions on new data. (Google Machine Learning Glossary: https://developers.google.com/machine-learning/glossary)

pattern recognition. A technique to classify input data into classes or objects by recognizing patterns or feature similarities. (DeepAI Glossary: https://deepai.org/definitions)

perception. A blanket term for an autonomous system's ability to gather information about itself and its environment. The autonomous system receives raw information from its onboard sensors through communications with other systems and humans. (DAU CLE 002: "Introduction to the Test & Evaluation (T&E) of Autonomous Systems")

precision. When referring to ML, a metric for classification models that answers the following question: When the model predicted the positive class, what percentage of the predictions were correct? (Google Machine Learning Glossary: https://developers.google.com/machine-learning/glossary)

predictive analytics. A branch of advanced analytics that makes predictions about future outcomes using historical data combined with statistical modeling, data mining techniques, and ML. (IBM Predictive Analytics Website: https://www.ibm.com/analytics/predictive-analytics)

reasoning. The mechanism of using available information to generate predictions, make inferences and draw conclusions. (IBM Website: https://www.ibm.com/think/topics/ai-reasoning)

recall. When referring to ML, A metric for classification models that answers the following question: When ground truth was the positive class, what percentage of predictions did the model correctly identify as the positive class? (Google Machine Learning Glossary: https://developers.google.com/machine-learning/glossary)

receiver operating characteristic curve. A curve of true positive rate versus false positive rate at different classification thresholds. (Google Machine Learning Glossary: https://developers.google.com/machine-learning/glossary)

regression testing. Statistical software testing to rerun functional and nonfunctional tests to ensure that previously developed and tested software still performs after a change. (AI Principles: Recommendations on the Ethical Use of AI by DoD)

reinforcement learning. ML system where software agents learn to take actions in an environment through the requirement to maximize some notion of cumulative reward (often discounted for future rewards) through episodic training. (AI Principles: Recommendations on the Ethical Use of AI by DoD)

robot. A powered machine capable of executing a set of actions by direct human control, computer control, or a combination of both. At a minimum, it is comprised of a platform, software, and a power source. (Joint Concept for Robotic and Autonomous Systems)

runtime monitoring. A lightweight and dynamic verification technique that involves observing the internal operations of a software system and/or its interactions with other external entities, with the aim of determining whether the system satisfies or violates a correctness specification. (Cassar et al. 2017)

supervised learning. ML that learns a function that maps inputs to outputs based on known input-output pairs from labeled data in a training sample. (AI Principles: Recommendations on the Ethical Use of AI by DoD)

swarm intelligence. An AI approach which is inspired by natural behavior to solve optimization problems. (Raslan et al. 2020)

test and evaluation (T&E). The process by which a system or components are compared against requirements and specifications through testing. The results are evaluated to assess progress of design, performance, supportability, etc. (DAU Website: https://www.dau.edu/cop/pm/resources/test-and-evaluation-mgmt)

test data. When used in reference to ML data, the subset of the dataset used to test a trained model. (Google Machine Learning Glossary: https://developers.google.com/machine-learning/glossary)

training data. When used in reference to ML data, the subset of the dataset used to train a model. (Google Machine Learning Glossary: https://developers.google.com/machine-learning/glossary)

transfer learning. ML method where a model developed for one task is applied to another, often related, task.

unit testing. A software testing method by which individual units of source code—sets of one or more computer program modules together with associated control data, usage procedures, and operating procedures—are tested to determine whether they are fit for use. (Huizinga and Kolawa 2007, 75)

unsupervised learning.

- ML that learns the underlying structure or distribution of unlabeled input data. (AI Principles: Recommendations on the Ethical Use of AI by DoD)
- When referring to ML, training a model to find patterns in a dataset, typically an unlabeled dataset. (Google Machine Learning Glossary: https://developers.google.com/machine-learning/glossary)

validation. The assessment of a planned or delivered system to meet sponsor's operational need in the most realistic environment achievable. (AI Principles: Recommendations on the Ethical Use of AI by DoD)

validation data. A subset of the dataset—disjoint from the training data—used in validation of an ML model. This data is utilized during the ML training process to evaluate the quality of the ML model and fine-tune hyperparameters. Because the validation set is disjoint from the training set, validation helps ensure that the model's performance generalizes beyond the training set.

verification. The process of assessing how well a system meets a specification requirement. (AI Principles: Recommendations on the Ethical Use of AI by DoD)

white box testing. Cybersecurity testing which utilizes cooperative knowledge about how the system was designed and implemented.

Acronyms

AFSIM	Advanced Framework for Simulation, Integration, and Modeling
AI	artificial intelligence
AMLAS	Assurance of Machine Learning for use in Autonomous Systems
CAE	claims, arguments, and evidence
CD	continuous delivery
CDAO	Chief Digital and Artificial Intelligence Office
CI	continuous integration
COE	center of excellence
CONEMP	concept of employment
CONOPS	concept of operations
CPS	cyber-physical systems
CPU	central processing unit
CT	contractor test/testing
DARPA	Defense Advanced Research Projects Agency
DAU	Defense Acquisition University
DEM&S	Digital Engineering, Modeling and Simulation
DoD	Department of Defense
DoDD	DoD directive
DoDI	DoD instruction
DOT&E	Director, Operational Test and Evaluation
DT	developmental test/testing
DT&E	developmental test and evaluation
DTE&A	Developmental Test, Evaluation, and Assessments
dTEaaC	developmental Test and Evaluation as a Continuum
FTRT	faster than real time
GPS	Global Positioning System
HACMS	High-Assurance Cyber Military Systems
HAT	human-autonomy team/teaming
HITL	human in the loop

HMT	human-machine teaming
HOOTL	human out of the loop
HOTL	human on the loop
HSI	human systems integration
HW	hardware
IEEE	Institute of Electrical and Electronics Engineers
IMPRINT	Improved Performance Research Integration Tool
ISTQB	International Software Testing Qualifications Board
IWARS	Infantry Warrior Simulation
JCIDS	Joint Capabilities Integration and Development System
KPI	key performance indicator
LVC	live, virtual, and constructive
M&S	modeling and simulation
MIT	Massachusetts Institute of Technology
ML	machine learning
MOSA	Modular Open Systems Approach
NASA	National Aeronautics and Space Administration
NAVAIR	Naval Air Systems Command
OAM	Open Architecture Management
OODA	observe, orient, decide, act
OT	operational test/testing
PIL	processor in the loop
PSASS	Partnership for Systems Approaches to Safety and Security
R&D	research and development
RAM	random-access memory
SIL	system integration laboratory
S&T	science and technology
SoS	system of systems
STAT	scientific test and analysis techniques
STPA	System-Theoretic Process Analysis

SUT	system under test
SW	software
SysML	Systems Modeling Language
T&E	test and evaluation
TEMP	Test and Evaluation Master Plan
TORE	task-oriented requirements engineering
TRMC	Test Resource Management Center
TTP	tactics, techniques, and procedures
UCA	unsafe control action
UML	Unified Modeling Language
USD(P)	Under Secretary of Defense for Policy
USD(R&E)	Under Secretary of Defense for Research and Engineering
UTP	UML Testing Profile
V&V	verification and validation
VC	virtual and constructive
VCJCS	Vice Chairman of the Joint Chiefs of Staff
VV&A	verification, validation, and accreditation
XAI	explainable artificial intelligence

References

Adam, Sebastian, Norman Riegel, and Joerg Doerr. "TORE: A Framework for Systematic Requirements Development in Information Systems." Requirements Engineering Magazine, October 30, 2014. https://re-magazine.ireb.org/articles/tore

AdvoCATE (Assurance Case Automation Toolset) User Guide. Version 1.4. National Aeronautics and Space Administration, May 5, 2022. https://ntrs.nasa.gov/api/citations/20220009664/downloads/AdvoCATE User Guide_1.4.pdf

AI Principles: Recommendations on the Ethical Use of Artificial Intelligence by the Department of Defense. Defense Innovation Board, October 2019. https://govwhitepapers.com/whitepapers/ai-principles-recommendations-on-the-ethical-use-of-artificial-intelligence-by-the-department-of-defense

Arrieta, Alejandro Barredo, Natalia Diaz-Rodriguez, Javier Del Ser, et al. "Explainable Artificial Intelligence (XAI): Concepts, Taxonomies, Opportunities and Challenges toward Responsible AI." arXiv:1910.10045v2 [cs.AI], December 26, 2019. https://arxiv.org/pdf/1910.10045

Artificial Intelligence Acquisition Guidebook. Department of the Air Force/Massachusetts Institute of Technology Artificial Intelligence Accelerator, February 14, 2022. https://aia.mit.edu/wp-content/uploads/2022/02/AI-Acquisition-Guidebook_CAO-14-Feb-2022.pdf

Autonomy Community of Interest (COI) Test and Evaluation, Verification and Validation (TEVV) Working Group: Technology Investment Strategy 2015–2018. Office of the Assistant Secretary of Defense for Research and Engineering, May 2015. https://apps.dtic.mil/sti/citations/AD1010194

Bjorkman, Eileen. "Joint Test and Evaluation Methodology (JTEM) Overview: Precision Strike Testing in a Joint Environment." 2007 Precision Strike Program Executive Office Summer Forum, July 10, 2007. https://ndia.dtic.mil/wp-content/uploads/2007/psa_peo/Bjorkman.pdf

Bloomfield, Robin, Gareth Fletcher, Luke Hinde, Philippa Ryan, and Heidy Khlaaf. "Safety Case Templates for Autonomous Systems." D/1294/87004/1 v4.0. Adelard LLP, February 26, 2021. https://arxiv.org/ftp/arxiv/papers/2102/2102.02625.pdf

Bowers, Ryan, and John Thomas. "Safety Implications of Autonomous Vehicles – System Theoretic Process Analysis Applied to a Neural Network-Controlled Aircraft." Society of Flight Test Engineers, 54th Annual International Symposium, Patuxent River, Maryland, October 16–19, 2023. https://psas.scripts.mit.edu/home/wp-content/uploads/2023/11/SFTE-Paper-STPA-for-autonomous-vehicles.pdf

Cassar, Ian, Adrian Francalanza, Luca Aceto, and Anna Ingólfsdóttir. "A Survey of Runtime Monitoring Instrumentation Techniques." arXiv:1708.07229v1 [cs.LO], August 24, 2017. https://doi.org/10.48550/arXiv.1708.07229

Davies, Misty, Tom Pressburger, Yuning He, and Karen Gundy-Burlet. " MARGInS: Model-based Analysis of Realizable Goals In Systems." National Aeronautics and Space Administration (NASA) Ames Research Center, June 10, 2014. https://ntrs.nasa.gov/api/citations/20190032074/downloads/20190032074.pdf

Department of Defense Cyber Developmental Test and Evaluation Guidebook. Version 3.0. Office of the Under Secretary of Defense for Research and Engineering, June 2025. https://aaf.dau.edu/storage/2025/07/Cyber-DTE-Guidebook-V3-June2025Update_Final-OFF.pdf

Department of Defense Data, Analytics, and Artificial Intelligence Adoption Strategy: Accelerating Decision Advantage. Office of the Deputy Secretary of Defense, June 27, 2023. https://media.defense.gov/2023/Nov/02/2003333300/-1/-1/1/DOD_DATA_ANALYTICS_AI_ADOPTION_STRATEGY.PDF

Department of Defense Experimentation Guidebook. Office of the Under Secretary of Defense for Research and Engineering, October 2021. https://www.dau.edu/tools/dod-experimentation-guidebook

Deputy Secretary of Defense Memorandum, "Implementing Responsible Artificial Intelligence in the Department of Defense," May 26, 2021. https://media.defense.gov/2021/May/27/2002730593/-1/-1/0/IMPLEMENTING-RESPONSIBLE-ARTIFICIAL-INTELLIGENCE-IN-THE-DEPARTMENT-OF-DEFENSE.PDF

Developmental Test and Evaluation of Artificial Intelligence-Enabled Systems Guidebook. Office of the Under Secretary of Defense for Research and Engineering, February 2025. https://www.cto.mil/wp-content/uploads/2025/02/TE_of_AIES_Guidebook_Final_26Feb25.pdf

Director, Operational Test and Evaluation (DOT&E) Test and Evaluation Master Plan (TEMP) Guidebook. Version 3.1. January 19, 2017. https://www.dote.osd.mil/Portals/97/docs/TEMPGuide/TEMP_Guidebook_3.1aa.pdf?ver=18gu6knyY3gqXNXxcf4TpA%3d%3d

DoD Directive 3000.09, "Autonomy in Weapon Systems," January 25, 2023.

DoD Instruction 5000.61, "DoD Modeling and Simulation Verification, Validation, and Accreditation," September 17, 2024.

DoD Instruction 5000.89, "Test and Evaluation," November 19, 2020.

DoD Instruction 5000.90, "Cybersecurity for Acquisition Decision Authorities and Program Managers," December 31, 2020.

DoD Zero Trust Strategy. DoD Zero Trust Portfolio Management Office, October 21, 2022. https://dodcio.defense.gov/Portals/0/Documents/Library/DoD-ZTStrategy.pdf

Dukart, Juergen and F. Hoffmann-La Roche. "Basic Concepts of Image Classification Algorithms Applied to Study Neurodegenerative Diseases." *Brain Mapping: An Encyclopedic Reference*, Volume 3, pages 641–646, 2015. http://dx.doi.org/10.1016/B978-0-12-397025-1.00072-5

El Samaloty, Nazli N., Roger Schleper, Mary Anne Fawkes, and Dean Muscietta. "Infantry

Warrior Simulation (IWARS): A Soldier-Centric Constructive Simulation." *Phalanx* 40(2): 29–31, 2007.
http://www.jstor.org/stable/24909630

Executive Director, Developmental Test, Evaluation, and Assessments Memorandum, "Advancements in Test and Evaluation of Autonomous Systems Workshop Report," August 26, 2022.
https://www.afit.edu/docs/2022%20ATEAS%20workshop%20Report%20-%20Distro%20A%20Memo2.pdf

Executive Order 14179, "Removing Barriers to American Leadership in Artificial Intelligence," January 23, 2025.
https://www.whitehouse.gov/presidential-actions/2025/01/removing-barriers-to-american-leadership-in-artificial-intelligence/

Fong, Terry. "Autonomous Systems: NASA Capability Overview." National Aeronautics and Space Administration (NASA), August 24, 2018.
https://www.nasa.gov/sites/default/files/atoms/files/nac_tie_aug2018_tfong_tagged.pdf

Green, Elizabeth A., Miriam E. Armstrong, and Janna Mantua. "Scientific Measurement of Situation Awareness in Operational Testing." *The ITEA Journal of Test and Evaluation* 44(3), 2023.
https://itea.org/journals/volume-44-3/scientific-measurement-of-situation-awareness-in-operational-testing/

Haase, Casey L., Raymond R. Hill, and Douglas D. Hodson. "Planning for LVC Simulation Experiments." *Applied Mathematics* 5(14): 2153–2167, 2014.
http://dx.doi.org/10.4236/am.2014.514209

Hawley, John K. "Patriot Wars: Automation and the Patriot Air and Missile Defense System." Center for a New American Security, January 2017.
https://s3.amazonaws.com/files.cnas.org/documents/CNAS-Report-EthicalAutonomy5-PatriotWars-FINAL.pdf

Huizinga, Dorota, and Adam Kolawa. *Automated Defect Prevention: Best Practices in Software Management*. ISBN 978-0-470-04212-0. Wiley-IEEE Computer Society Press, 2007.

Human Systems Integration Test and Evaluation of Artificial Intelligence-Enabled Capabilities. Chief Digital and Artificial Intelligence Office, April 2024.
https://www.ai.mil/Portals/137/Documents/Resources%20Page/Human%20Systems%20Integration%20Test%20and%20Evaluation%20of%20AI-Enabled%20Capabilities%20Framework.pdf

Institute of Electrical and Electronics Engineers (IEEE) 1872.1-2024, "IEEE Standard for Robot Task Representation," June 18, 2024.

Johnson, Kip E. "Systems-Theoretic Safety Analyses Extended for Coordination." Massachusetts Institute of Technology, 2017.
https://dspace.mit.edu/handle/1721.1/108922

Joint Concept for Robotic and Autonomous Systems (JCRAS). Joint Chiefs of Staff, October 19, 2016.
https://jdeis.js.mil/jdeis/jel/concepts/robotic_autonomous_systems.pdf

Kopeikin, Andrew N. “System-Theoretic Safety Analysis for Teams of Collaborative Controllers.” Massachusetts Institute of Technology, 2024. https://dspace.mit.edu/handle/1721.1/153787

Krausman, Andrea, Catherine Neubauer, Daniel Forster, et al. “Trust Measurement in Human-Autonomy Teams: Development of a Conceptual Toolkit.” *ACM Transactions on Human-Robot Interaction* 11(3), Article 33, 58 pages, 2022. https://dl.acm.org/doi/full/10.1145/3530874

Lefeuvre, Hugo, Nathan Dautenhahn, David Chisnall, and Pierre Olivier. “SoK: Software Compartmentalization.” arXiv:2410.08434v1 [cs.CR], October 11, 2024. https://arxiv.org/pdf/2410.08434

Leveson, Nancy G., and John P. Thomas. “STPA Handbook.” MIT Partnership for Systems Approaches to Safety and Security (PSASS), March 2018. https://psas.scripts.mit.edu/home/get_file.php?name=STPA_handbook.pdf

Military Standard MIL-STD-3022, “Documentation of Verification, Validation, and Accreditation (VV&A) for Models and Simulations,” April 5, 2012.

Mills, Barron. “Live, Virtual, and Constructive Training Environment: A Vision and Strategy for the Marine Corps.” Naval Postgraduate School, September 2014.

Mitchell, Diane K. “Advanced Improved Performance Research Integration Tool (IMPRINT) Vetronics Technology Test Bed Model Development.” ARL-TN-0208. Army Research Laboratory, September 2003.

Mullins, Galen E., Paul G. Stankiewicz, R. Chad Hawthorne, et al. “Delivering Test and Evaluation Tools for Autonomous Unmanned Vehicles to the Fleet.” *Johns Hopkins APL Technical Digest* 33(4): 279–288, 2017. https://secwww.jhuapl.edu/techdigest/Content/techdigest/pdf/V33-N04/33-04-Hawthorne.pdf

Nielsen, Jakob. “10 Usability Heuristics for User Interface Design.” Nielsen Norman Group (NN/g), January 30, 2024. https://www.nngroup.com/articles/ten-usability-heuristics/

Open Mission Systems (OMS) in a Nutshell. Department of the Air Force Virtual Distributed Laboratory, 2024. https://www.vdl.afrl.af.mil/programs/oam/OMS_Marketing.pdf

Political Declaration on Responsible Military Use of Artificial Intelligence and Autonomy. Department of State, November 9, 2023. https://www.state.gov/political-declaration-on-responsible-military-use-of-artificial-intelligence-and-autonomy/

Pressman, Roger S., and Bruce R. Maxim. *Software Engineering: A Practitioner’s Approach*. 9th ed. McGraw Hill, 2020.

Raslan, Aliaa F., Ahmed F. Ali, and Ashraf Darwish. “Swarm Intelligence Algorithms and Their Applications in Internet of Things.” *Swarm Intelligence for Resource Management in Internet of Things*, pp. 1–19. Academic Press, 2020.

Refsdal Atle, Bjørnar Solhaug, and Ketil Stølen. “Cyber-systems.” In *Cyber-Risk Management*. SpringerBriefs in Computer Science. Springer, Cham, 2015. https://doi.org/10.1007/978-3-319-23570-7_3

Roback, Kevin P. “Review of Potential Assurance Case Tool Options for DoD.” IDA Publication D-33524 /2. Institute for Defense Analyses, January 2024. https://apps.dtic.mil/sti/trecms/pdf/AD1211550.pdf

Rozier, Kristin Y., and Johann Schumann. “R2U2: Tool Overview.” International Workshop on Competitions, Usability, Benchmarks, Evaluation, and Standardisation for Runtime Verification Tools, Seattle, Washington, September 13–16, 2017. https://ntrs.nasa.gov/citations/20190026747

Sacolick, Isaac. “What is CI/CD? Continuous integration and continuous delivery explained.” InfoWorld, April 1, 2024. https://www.infoworld.com/article/2269266/what-is-cicd-continuous-integration-and-continuous-delivery-explained.html

Schaefer, Kristin E., Edward R. Straub, Jessie Y.C. Chen, Joe Putney, and A.W. Evans. “Communicating Intent to Develop Shared Situation Awareness and Engender Trust in Human-Agent Teams.” *Cognitive Systems Research* 46: 26–39, 2017. https://www.sciencedirect.com/science/article/pii/S1389041716301802

Scheidt, David. “A Holistic Look at Testing Autonomous Systems.” 31st Annual National Test and Evaluation Conference, March 3, 2016. https://ndia.dtic.mil/wp-content/uploads/2016/Test/Scheidt.pdf

Scheidt, David, Robert Lutz, William D’Amico, Dean Kleissas, Robert Chalmers, and Robert Bamberger. “Safe Testing of Autonomous Systems Performance.” Interservice/Industry Training, Simulation, and Education Conference (I/ITSEC), 2015. http://www.iitsec.org/-/media/sites/iitsec/link-attachments/best-papers-and-tutorials-from-past-iitsec/15348_ecit_paper.ashx?la=en

Stumborg, Michael F., Timothy D. Blasius, Steven J. Full, and Christine A. Hughes. “Goodhart’s Law: Recognizing and Mitigating the Manipulation of Measures in Analysis.” COP-2022-U-033385-Final. Center for Naval Analyses, September 2022. https://www.cna.org/reports/2022/09/goodharts-law

Summary of the 2018 Department of Defense Artificial Intelligence Strategy: Harnessing AI to Advance Our Security and Prosperity. Department of Defense, 2018. https://media.defense.gov/2019/feb/12/2002088963/-1/-1/1/summary-of-dod-ai-strategy.pdf

Test Planning Guide. Scientific Test and Analysis Techniques Center of Excellence, 2022. https://www.afit.edu/images/pics/file/Final%200930_Test%20Planning%20Guide_2_2%20(1).pdf

Tobin Josh, Rachel Fong, Alex Ray, Jonas Schneider, Wojciech Zaremba, and Pieter Abbeel. “Domain Randomization for Transferring Deep Neural Networks from Simulation to the Real World.” arXiv: 1703.06907v1 [cs.RO], March 20, 2017. https://arxiv.org/abs/1703.06907

Trusted AI and Autonomy Roadmap – A Holistic, system of systems approach to development of resilient AI and autonomy. Office of the Under Secretary of Defense for Research and

Engineering/Critical Technologies, 2024.
https://dod-cta.s3.us-west-2.amazonaws.com/roadmaps/(U) TAIA Distro A Roadmap - DOPSR APPROVED.pdf

Tuncali, Cumhur Erkan, Georgios Fainekos, Hisahiro Ito, and James Kapinski. "Simulation-based Adversarial Test Generation for Autonomous Vehicles with Machine Learning Components." In Proceedings of the *2018 IEEE Intelligent Vehicles Symposium (IV)*, Changshu, China, pp. 1555–1562, 2018.
https://doi.org/10.1109/IVS.2018.8500421

United States Code, Title 10, Section 4401

Unmanned Maritime Autonomy Architecture (UMAA). Department of the Navy Program Executive Office for Unmanned and Small Combatants, 2020.
https://www.dsp.dla.mil/Portals/26/Documents/Conference/2020-StateofDSPConf_Rothgeb.pdf

Wisnowski, James, Andrew Karl, and Darryl Ahner. "JMP BEAST Mode: Boundary Exploration through Adaptive Sampling Techniques." Discovery Summit Americas, October 2020.
https://community.jmp.com/t5/Abstracts/JMP-BEAST-Mode-Boundary-Exploration-through-Adaptive-Sampling/ev-p/758511

Wojton, Heather, Kelly M. Avery, Laura J. Freeman, et al. "Handbook on Statistical Design & Analysis Techniques for Modeling & Simulation Validation." IDA Document NS D-10455. Institute for Defense Analyses, February 2019.
https://apps.dtic.mil/sti/pdfs/AD1122387.pdf

Yang, Xin-She. *Nature-Inspired Optimization Algorithms*. Academic Press, 2020.

Young, William. "Basic Introduction to STPA for Security (STPA-Sec)." 2020 System-Theoretic Accident Model and Processes (STAMP) Workshop, July 22, 2020.
https://psas.scripts.mit.edu/home/wp-content/uploads/2020/07/STPA-Sec-Tutorial.pdf

Websites

AFSIM.
https://dsiac.dtic.mil/models/afsim/

AI: A Glossary of Terms.
https://link.springer.com/content/pdf/bbm%3A978-3-319-94878-2%2F1.pdf

Air Force Test Center Orange Flag.
https://www.aftc.af.mil/Test-Flag-Enterprise/Orange-Flag/

AMLAS Tool.
https://www.assuringautonomy.com/amlas/tool

Britannica Reasoning.
https://www.britannica.com/technology/artificial-intelligence/Reasoning

DARPA Assured Autonomy.
https://www.darpa.mil/program/assured-autonomy

DARPA HACMS.
https://www.darpa.mil/research/programs/high-assurance-cyber-military-systems

DAU Post-Implementation Review.
https://www.dau.edu/cop/it/resources/post-implementation-review

DAU Reliability Growth.
https://www.dau.edu/acquipedia-article/reliability-growth

DAU Requirements Management.
https://content1.dau.edu/DAUMIG_se-brainbook_189/content/Management Processes/Requirements-Management.html

DAU Systems Engineering Brainbook.
https://www.dau.edu/tools/dau-systems-engineering-brainbook

DeepAI Glossary.
https://deepai.org/definitions

DEM&S Community of Practice.
https://www.cto.mil/sea/dems_cop/

DoD Issuances.
https://www.esd.whs.mil/DD/DoD-Issuances/

Google Machine Learning Glossary.
https://developers.google.com/machine-learning/glossary

IBM Predictive Analytics.
https://www.ibm.com/analytics/predictive-analytics

IEEE Standards Association.
https://standards.ieee.org/

IEEE Standards Reading Room.
https://ieeexplore.ieee.org/browse/standards/reading-room/page

ISTQB Glossary.
https://glossary.istqb.org/en/search/

MIT PSASS.
https://psas.scripts.mit.edu/home/

NAVAIR MOSA.
https://www.navair.navy.mil/MOSA

Open Architecture Management (OAM).
https://www.vdl.afrl.af.mil/programs/oam/index.php

STAT COE.
https://www.afit.edu/stat/index.cfm

STAT COE Ask-a-STAT resource.
https://www.afit.edu/STAT/page.cfm?page=498

Test Science Measuring Usability.
https://testscience.org/measuring-usability/

This page is intentionally blank.

Developmental Test and Evaluation of Autonomous Systems Guidebook

Office of the Director, Developmental Test, Evaluation, and Assessments
Office of the Under Secretary of Defense for Research and Engineering
3030 Defense Pentagon
Washington, DC 20301-3030
osd.r-e.comm@mail.mil
https://www.cto.mil/dtea/