INSTITUTE FOR DEFENSE ANALYSES

Test & Evaluation of AI-enabled and Autonomous Systems: A Literature Review

Heather M. Wojton, Project Leader

Daniel J. Porter
John W. Dennis

September 2020

IDA Document NS-D-14331

Log: H 2020-000326

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

About This Publication
This work was conducted by the Institute for Defense Analyses (IDA) under contract HQ0034-19-D-0001, Task 229990, “Test Science,” for the Office of the Director, Operational Test and Evaluation. The views, opinions, and findings should not be construed as representing the official position of either the Department of Defense or the sponsoring organization.

Acknowledgments
The IDA Technical Review Committee was chaired by Mr. Robert R. Soule and consisted of Rachel A. Haga, John T. Haman, Mark R. Herrera, and Brian D. Vickers from the Operational Evaluation Division, and Nicholas J. Kaminski from the Science & Technology Division

For more information:
Heather M. Wojton, Project Leader
hwoton@ida.org, (703) 845-6811

Robert R. Soule, Director, Operational Evaluation Division
rsoule@ida.org • (703) 845-2482

INSTITUTE FOR DEFENSE ANALYSES

IDA Document NS-D-14331

Test & Evaluation of AI-enabled and Autonomous Systems: A Literature Review

Heather M. Wojton, Project Leader

Daniel J. Porter
John W. Dennis

Executive Summary

This paper summarizes a subset of the literature regarding the challenges to and recommendations for the test, evaluation, verification, and validation (TEV&V) of autonomous military systems. This literature review is meant for informational purposes only and does not make any recommendations of its own.

A synthesis of the literature identified the following categories of TEV&V challenges:

1. Problems arising from the complexity of autonomous systems;
2. Challenges imposed by the structure of the current acquisition system;
3. Lack of methods, tools, and infrastructure for testing;
4. Novel safety and security issues;
5. A lack of consensus on policy, standards, and metrics;
6. Issues around how to integrate humans into the operation and testing of these systems.

Recommendations for how to test autonomous military systems can be sorted into five broad groups:

1. Use certain processes for writing requirements, or for designing and developing systems;
2. Make targeted investments to develop methods or tools, improve our test infrastructure, or enhance our workforce's AI skillsets;
3. Use specific proposed test frameworks;
4. Employ novel methods for system safety or cybersecurity;
5. Adopt specific proposed policies, standards, or metrics.

Table of Contents

Introduction

Over the past decade, advancements in computing and machine learning have led to the proliferation of artificial intelligence (AI)-enabled capabilities for industrial, civilian, and academic applications (e.g., Gil & Selman, 2019; Narla, Kuprel, Sarin, Novoa, & Ko, 2018; Silver et al., 2016; Templeton, 2019). Systems enabled by AI often behave autonomously in some sense: they may take over decisions traditionally made by humans or perform tasks with less supervision. However, a vacuum robot, a high-frequency stock trading system, or even an autonomous car making a bad choice is relatively recoverable with corrective action compared to a wrong decision during armed conflict. Military systems will have most of the same challenges as their civilian equivalents, but more often will operate in less structured environments, with shorter required reaction times, and in the context of an adversary actively seeking to exploit mistakes. AI-enabled and autonomous military systems will require robust testing to provide assurance that undesirable outcomes such as fratricide, collateral damage, and poor mission performance are unlikely and within acceptable risk parameters.

To confidently field autonomous military systems (AMSs),[1] we must trust that they will make appropriate decisions for both the foreseeable problems for which they were designed and the unforeseeable situations to which they must adapt. In short, these systems must be *proficient*, *flexible*, and *trustworthy*.[2] When AMSs are meant to operate in narrowly defined situations (e.g., asking a "smart" mine to explode when a certain pressure is applied for a particular duration at a given time of day), it is much easier to provide assurance that the system will behave as desired. The number of relevantly different situations it can encounter and its behavioral repertoire in response (i.e., the state-space of its decision-making) are both limited. Expanding that state-space makes assurance more difficult. For example, an autonomous base defense system meant to respond to any possible threats with appropriate force in line with the current ROEs can be expected to encounter a wider variety of situations, both designed-for and unforeseeable. To function appropriately in this situation requires more flexibility, which in turn requires more proficiency from the system and more trust from the humans who allow it to operate. The interaction of these needs is a core driver of many T&E difficulties for these systems.

AI-enabled technologies introduce a host of challenges to the process for testing and evaluating acquisition programs within the Department of Defense (DoD). First, the sheer technical complexity and novelty of these systems can be difficult to navigate. Additionally, the DoD acquisition process is optimized based on assumptions that may no longer hold true with autonomy (Tate & Sparrow, 2018). For example, separating contractor, developmental, and operational testing assumes we have discrete, relatively linear stages of development that lead to

[1] We limit ourselves to the consideration of military systems in this document. There are many challenges unique to national security, and many civilian recommendations do not transfer to the military domain.

[2] See Zacharias (2019a) *Autonomous Horizons: The Way Forward* for an in-depth discussion of the properties of *proficiency, trust, and flexibility* along with how they interact to produce useful autonomous behavior.

a “production representative” version of the system. This may not be true with AMSs, especially if they continue to learn throughout their lifecycles. Additionally, writing requirements before we have a system assumes we understand how it will be used in advance. Because the AMS’s proficiency, flexibility, and trustworthiness will evolve over time and can affect how humans use or interact with the system, Concepts of Operations (CONOPS) and tactics, techniques, and procedures (TTPs) will need to be co-developed with the system to a greater extent than with standard systems (Haugh, Sparrow, & Tate, 2018; Hill & Thompson, 2016; Porter, McAnally, Bieber, & Wojton, 2020; Zacharias, 2019b).

Even if DoD acquisition processes are updated, however, the specific methods, tools, and infrastructure that working-level DoD employees use for test and evaluation (T&E) will fail to provide assurance that systems will perform as expected. Development and design efforts which embrace testing—building in testability through internal instrumentation; enhancing the transparency, traceability, or explainability of the software; and executing good governance and validation of training and other data—can improve development processes while also paving the way for T&E, but they are not universally adopted. Furthermore, the policies and standards that would help programs overcome all these challenges are either lacking or non-existent.

Recently, a number of T&E groups have begun identifying these challenges and proposing possible solutions to overcome them. This paper summarizes a portion of that literature. In Appendix A, we provide summaries of the individual studies, with specific extracts of the challenges or recommendations their authors call out. For Appendix B, we created tables that categorize these challenges and recommendations, aggregating which authors advocate for these different viewpoints. The main text of this paper explores these tables at a high level. Some perspectives are overrepresented by the processes used to acquire articles and presentations, and readers should not assume that the number of different cited authors equates to the weight of support for a perspective. Readers should also note that although the challenge and recommendation sections are often structured in parallel, they are organized by topic, and there is not a one-to-one mapping between challenges identified and recommendations made. By completing this review, we hope to bring this conversation to a broader community and identify gaps to fill in with future work. However, before proceeding to these challenges and recommendations, it is important to clarify what is meant when we discuss autonomy in military systems.

What is Autonomy?

Definitions of autonomy abound, and some are less useful to DoD than others. Many incorporate concepts of independence, acting without external control or oversight, or separation from other entities (e.g., Oxford English Dictionary, 2020). However, the assumption that any participant will operate without control or oversight, even a human warfighter, is anathema to DoD policy and thinking on command and control (C2). We do not want our autonomous systems to have complete freedom to select courses of action, but rather to have some constrained freedom within their assigned tasks.

As with our warfighters, we likely want to have a C2 or principal-agent relationship with our autonomous systems. We will want to:

1. Specify the goals or objectives of the specific task and/or the overall mission, and possibly the larger reasons for those goals, like the commander's intent (i.e., *what* to do and *why),*
2. Specify the constraints associated with the task, such as Rules of Engagement (ROEs, i.e., *what not* to do).
3. *Not* specify the methods to use or give explicit contingencies for every situation, like reacting to the adversary's response (i.e., *how* to do the task).

Whether a system is empowered to make these 'how' decisions for a task is how this paper will differentiate autonomous from non-autonomous systems.

Making useful, desirable choices for *how*, within the constraints of *what*, *what not*, and *why*, presumes some level of intelligence. Because these are machines, this presumes the presence of some level of AI. The need for AI to enable useful autonomy for non-trivial tasks likely explains why AI and autonomy are often conflated. In this document, we will refer to autonomy as what the system *does* in its operational environment, and AI as the "under-the-hood" enabler of meaningful interactions with that environment.

T&E Challenges for Autonomy

Challenge #1: System Complexity

The sheer complexity of AMS is one of the primary challenges not only for design, but also for testing. These systems perform difficult, complicated, and consequential tasks in unpredictable environments, and they may make non-deterministic, dynamic responses to those environments. This combination of complexity and stochasticity leads to a "state-space explosion," radically increasing the number of scenarios across which we must evaluate system performance. AMS will also likely involve learning-enabled components (LECs). LECs further exacerbate the state-space explosion and introduce additional capabilities to evaluate. AMSs will not operate in isolation, and testers must assess whether they can effectively interact with other systems, adding the potential for emergent behavior on top of standard interoperability concerns. The novelty of these systems compounds the above problems. Though we have designed and fielded systems of considerable complexity before, they emerged from domains with strong theoretical underpinnings and a long history of hard-won empirical knowledge. Finally, exacerbating all of these challenges is the issue of transparency. These systems are not only complex, but often that complexity is hidden within a 'black box', making them complex to an unknown degree.

Task Complexity

The complexity of system tasks makes them difficult to evaluate. First, testers need to understand what tasks a system *can* do in order to define the space of possibility from which to sample test points (Hernández-Orallo, 2016). This is a hard enough challenge by itself (Baker et al., 2019), but to evaluate AMS behaviors, the T&E community must have definitions of not just what the system *could* do but of what it *should* do in those situations, which is even harder (Deonandan, Valerdi, Lane, & Macias, 2010; Porter et al., 2020; Roske, Kohlberg, & Wagner, 2012) because real-world tasks often have multiple viable solutions (Laverghetta, Leathrum, & Gonda, 2018). Furthermore, they would need to quantitatively define those outcomes (Roske et al., 2012), as qualitative evaluation quickly becomes intractable (Baker et al., 2019). On top of this, the more autonomy a system is given, the more flexible we expect its behavior to be, increasing both the number of possible behaviors and the variety of 'appropriate' ones (Ilachinski, 2017; Zacharias, 2019a). There is also an innate tension between evaluating a systems' capacity to react appropriately to unexpected and/or unforeseeable situations and the testers' desire to pre-plan, select test conditions with forethought, and control the environment. As the required level of flexibility grows, it becomes increasingly difficult to demonstrate that an AMS has sufficient proficiency across the range of situations (Micskei, Szatmári, Oláh, & Majzik, 2012; Zacharias, 2019a). Current DoD T&E processes need to be improved to handle unscripted, stochastic, or dynamic testing well (Ahner, Parson, Thompson, & Rowell, 2018; Lenzi, Bachrach, & Manikonda, 2010; Macias, 2008).

The State-Space Explosion

There is strong consensus that the state-space explosion resulting from the interaction of tasks' and systems' growing complexity will make it impossible, under any realistic assumptions, to exhaustively test all scenarios (Ahner & Parson, 2016; Ahner et al., 2018; Defense Science Board, 2012, 2016; Deonandan et al., 2010; Giampapa, 2013; Goerger, 2004; Haugh et al., 2018; Ilachinski, 2017; Menzies & Pecheur, 2005; Micskei et al., 2012; Porter et al., 2020; Porter et al., 2018; Sparrow, Tate, Biddle, Kaminski, & Madhavan, 2018; Wegener & Bühler, 2004; Zacharias, 2019a, 2019b). As we add conditions, factors, or interactions which must be tested, this state-space grows exponentially (Ahner & Parson, 2016), and even the use of modeling and simulation (M&S) cannot enable us to cover it comprehensively (Defense Science Board, 2012), at least not before the heat death of the universe (Zacharias, 2019b). These systems will require a different approach to assurance (Tate & Sparrow, 2019).

Stochastic/Non-Deterministic/Chaotic Processes

The stochasticity of the environments and systems also hinders using mathematical or formal proofs to provide assurance. The environments themselves are unstructured and unpredictable (Defense Science Board, 2012), making it difficult to specify all states for these formal methods (Goerger, 2004). The existence of other decision agents (human or artificial) in an environment can also dynamically change it through their actions, making it even more volatile and difficult to specify(Lenzi et al., 2010; Menzies & Pecheur, 2005; Micskei et al., 2012). Furthermore, systems might have more than one possible response to the same situation (Ahner & Parson, 2016) or otherwise incorporate non-deterministic/stochastic elements such as true or hardware random number generation to select between behaviors (Hernández-Orallo, 2016; Laverghetta et al., 2018), making them difficult to model (Goerger, 2004; Hernández-Orallo, 2016). Even though AI-enabled systems can be deterministic at a very detailed level (e.g., feeding the exact same pixel inputs produces the same classification), or might be formally chaotic, this determinism is usually below the level at which humans would consider situations to be equivalent (i.e., rotating an image by five degrees changes the pixels, but a human would consider identifying each image to be the same task), and these exact states may never occur again in the system's lifetime. It may be more useful to model these systems as stochastic even if they technically are not (Porter et al., 2020). When it comes to unpredictable environments and systems sensitive to perturbation, our standard T&E methodologies will probably *not* be sufficient (Subbu, Visnevski, & Djang, 2009).

Learning Systems

LECs introduce a number of challenges to T&E. Problems in training data—whether caused by adversarial interference (Qiu, Liu, Zhou, & Wu, 2019; Streilein et al., 2019) or shortcomings in the collection, labeling, or use of data (Gil & Selman, 2019)—will often lead to problems in behavior. T&E must confirm the integrity and operational validity of the data used to train a system, but there are not yet standard processes for doing so (Haugh et al., 2018; Qiu et al., 2019; Streilein et al., 2019). This is true whether the system is static after development or continues to learn. However, systems that continue to learn pose greater challenges, because they can change the state-space of their decision-making, expanding to new possibilities or creating novel responses to old ones. This is problematic for testing because previously observed performance can quickly stop being representative of the system's behavior and could demand recertification or regression testing (Ahner & Parson, 2016; Haugh et al., 2018; Micskei et al., 2012). Although some might infer that these systems will only improve from this process, and therefore not need recertification, learning in an ML context should be read simply as 'change.' Systems can learn negative behaviors (Haugh et al., 2018; Ilachinski, 2017) or experience catastrophic forgetting (Kirkpatrick et al., 2016), and this might happen over a longer period than is typical for testing a system (Ahner & Parson, 2016). Standard T&E processes are not suited for dynamic or learning-enabled systems (Visnevski & Castillo-Effen, 2010).

Multi-Agent/Component Evaluation

Another major consensus is that T&E capabilities will be challenged by the emergent properties and interoperability issues resulting from interactions between multiple autonomous entities. The interaction of decision-making entities, either within a system (e.g., AI modules) or between discrete artificial or biological agents, will exacerbate the state-space explosion and stochasticity problems by introducing the possibility of emergent behavior (Ahner & Parson, 2016; Baker et al., 2019; Defense Science Board, 2016; Ferreira, Faezipour, & Corley, 2013; Greer, 2013; Haugh et al., 2018; Lenzi et al., 2010; Luna, Lopes, Tao, Zapata, & Pineda, 2013; Mueller, Hoffman, Clancey, Emrey, & Klein, 2019). When these entities interact, they can produce behavior or capabilities that could not be predicted or enabled by any single entity alone, the definition of emergence (O'Connor & Wong, 2012). These emergent properties are not always undesirable (e.g., teaming or synergy; Ferreira et al., 2013), and so T&E will need to demonstrate that systems' desirable emergent behaviors are reliable while undesirable ones are unlikely (Porter et al., 2020). Further, AMSs also need to interact with our current legacy systems, which requires interoperability compliance testing (Visnevski, 2008), and they will eventually *be* the legacy systems, creating the potential for ever-evolving emergence and interoperability requirements (Deonandan et al., 2010; Harikumar & Chan, 2019).

System Opacity

Finally, all these issues of complexity are compounded by a lack of transparency in AI-enabled systems. As discussed earlier, exhaustive testing is not just possible for AMS, so we must rely on inference from a more limited set of test points. Agents can make a correct decision but do it for the wrong reason (e.g., decide not to fire because it did not see any object, not because it made a correct identification). As the example illustrates, *why* an AMS makes a decision is more important for generalizing behavior than *what* decision it makes, but testers fundamentally cannot access the "why" of a black-box system (Porter et al., 2020). To provide assurance for AMS, we will need to understand its decision-making process (Ahner & Parson, 2016; Arnold & Scheutz, 2018; Giampapa, 2013; Gunning, 2017; Haugh et al., 2018; Subbu et al., 2009), but methods to do so are still in their infancy (Gunning, 2019; Tate & Sparrow, 2019). This transparency challenge is compounded at the system and environmental level: we rarely have the capability to capture for comparison both how the system represented a situation and the ground truth in the environment (Ahner & Parson, 2016; Ahner et al., 2018; Haugh et al., 2018; Roske et al., 2012; Zhou & Sun, 2019). When systems are black boxes, it is extremely difficult to know their state-space, how stochastic they are, how sensitive they are to perturbations, how or what they learn, and what happens when they interact with other agents.

Novelty

The novelty of these systems is a challenge in and of itself. DoD will need to integrate separate sets of existing knowledge about challenges and solutions for testing complex physical and cyber systems (Eaton et al., 2017; Harikumar & Chan, 2019; Ilachinski, 2017; Luna et al., 2013; Woods & Dekker, 2000). Many of those lessons were learned through slow experience, or rely on historical empirical data for comparison, both of which we do not yet have for AMS. Furthermore, we must recognize that many challenges are simply 'unknown unknowns' at this time (Deonandan et al., 2010; Luna et al., 2013; Scheidt, 2017). Lack of experience also hinders our ability to know when it is or is not safe to reuse evidence, either within or across programs (Ahner & Parson, 2016; Deonandan et al., 2010; Durst, 2019; Giampapa, 2013; Lede, 2019).

Challenge #2: Acquisition System Limitations

AMS will compound the existing shortcomings of the DoD acquisition system while simultaneously introducing new problems. The requirements process is ill-suited to adaptive, fast-paced technology; the processes for designing and running tests may not apply well to AI; and the stovepiping that is standard in acquisition will exacerbate other challenges.

Requirements

DoD has historically struggled to write operationally relevant system requirements that are concrete and testable while also being acceptable to contractors. While this fight is not new, its urgency is. The defense community has relied on the flexibility of human decision-making to overcome shortcomings in requirements and design, whether through workarounds (Endsley, 2015), or by changing tactics as adversaries learn to exploit those shortcomings (Zacharias, 2019a). However, the point of autonomy is to reduce the human element, and therefore it removes this plasticity. This flexibility needs to be designed into AMSs themselves, and so it must be a system requirement. The current rigidity—in both the specifications themselves and the process by which they are created—will make this difficult (Ahner & Parson, 2016; Deonandan et al., 2010; Lede, 2019; Luna et al., 2013; McLean, Bertram, Hoke, Rediger, & Skarphol, 2016). Technical specifications like bit/sec or latency, while necessary, are not sufficient for these systems to be operationally successful (Ahner & Parson, 2016; Durst & Gray, 2014; Kapinski, Deshmukh, Jin, Ito, & Butts, 2016; Micskei et al., 2012; Schultz, Grefenstette, & Jong, 1993; Visnevski & Castillo-Effen, 2010; Zhou & Sun, 2019). The acquisition community needs, but does not have, a process for writing operationally relevant, mission-focused requirements that are also testable, verifiable hypotheses (Durst & Gray, 2014; Hess & Valerdi, 2010; Lede, 2019; Micskei et al., 2012; Zhou & Sun, 2019). Furthermore, AMSs will introduce the need for new types of requirements that will be particularly difficult to define, such as for legal, moral, and ethical (LME) behavior (Hill & Thompson, 2016; Roske et al., 2012; Scheidt, 2017; US Department of Defense, 2019).

Processes

AMSs will challenge the T&E community's processes for test planning and execution. Currently, tests are often designed years in advance—a practice already the target of criticism for its lack of agility—and what is cumbersome for static systems will be unacceptable for dynamic ones (Macias, 2008). DoD acquisition assumes that experimentation is over once a contract is awarded (Tate & Sparrow, 2018), but testers will be learning about their systems as they proceed, and if LECs are involved, the system itself will be changing. . Rigid and slow planning, coordination, and approval steps will prevent test planners from adapting efficiently as information emerges (Ahner & Parson, 2016; Hess & Valerdi, 2010; Ilachinski, 2017; Macias, 2008; McLean et al., 2016; Tate & Sparrow, 2018). Because metrics and evaluation strategies are known to all stakeholders—sometimes even years in advance—we run an increased risk that developers[3] or the AI itself will be able to use this slow timescale to "game" the test, passing the letter but not the spirit of the evaluation (Arnold & Scheutz, 2018; Hernández-Orallo, 2016; Visnevski & Castillo-Effen, 2010). The history of AI is rife with such examples of "reward hacking" (Gray, 2015; Johnson, 1984; Lenat, 1983; Narla et al., 2018), but these are most often discovered post-hoc, and it may not even be obvious when this has occurred. However, the solution to issues surrounding inagile processes cannot be to abandon rigorous evaluations of test adequacy. Fast tests that do not enable evaluation of performance do not provide value, and an agile but inadequate test is still inadequate (Porter et al., 2018).

Stovepiping

There is consensus that the stovepiping among different arms of the T&E community—e.g., contractor testing (CT), developmental testing (DT), and operational testing (OT)—will create difficulties for testing AMS. The most basic issue is the difficulty of tracing and transferring data, evidence, and knowledge across these different activities (Zacharias, 2019b). This is true for any handoff, but because AMS will likely be developed iteratively and sequentially, with capabilities maturing at different rates as breakthroughs are made, components may cross what would traditionally be the threshold for the next stage of testing at different times (McLean et al., 2016). Furthermore, effective development of AI capabilities will require injecting operational realism much earlier in the development process (Porter et al., 2018; Visnevski, 2008), as well as continuing it much later: systems that continue to evolve over their lifetimes need to be tested over their lifetimes, from the cradle to the grave. Tester participation would therefore be needed throughout the entire system lifecycle (Ahner & Parson, 2016; Porter et al., 2018; Sparrow et al., 2018; Tate & Sparrow, 2019; Visnevski, 2008). Maintaining the traditional distinction between CT, DT, and OT will likely be unsustainable for AMS (Haugh et al., 2018; Hess & Valerdi, 2010; McLean et al., 2016; Porter et al., 2018; Visnevski, 2008). Leaving them separate will make coordinating activities across the different sites and workforces too difficult (Deonandan et al., 2010; Haugh et al., 2018; Hernández-Orallo, 2016; McLean et al., 2016; Ring, 2009), result in duplicated efforts (Hernández-Orallo, 2016; Porter et al., 2020; Porter et al., 2018), and create numerous opportunities for conflicts over resources such as range time, compute cycles, talent,

[3] This does not necessarily imply nefarious intent on anyone's part; perfectly well-meaning developers can overoptimize their systems to these metrics, but the result is similarly problematic for evaluating system adequacy.

and funding (Ahner & Parson, 2016; Goerger, 2004; Macias, 2008; McLean et al., 2016; Visnevski & Castillo-Effen, 2010; Zhou & Sun, 2019).

Challenge #3: Lack of Methods, Tools, or Infrastructure

In addition to the larger structural challenges AMS presents to the DoD acquisition pipeline, at a working level, testers do not have what they need to do their jobs. Many of the methods and tools used for test planning, execution, and analysis require adaptation for AMS, and the workarounds that have been proposed may not be scalable. Even with these methods, testers will lack the data to feed these techniques because systems are currently not instrumented as needed. Furthermore, the physical and digital venues where systems would produce these data are currently insufficient. Finally, it is unclear whether the T&E workforce has the capacity and skillsets demanded by the complexity of AMS.

Method Needs

Many of the methods and tools T&E employs today will require changes to work for AMS. DoD testing currently relies on making inferences between sparse and aliased test points, but this assumes we understand the causality of the underlying processes, which will not be the case for black-box learning systems (Porter, 2019). We will need methods for generalizing system performance and replicating test results for black-box systems (Air Force Scientific Advisory Board, 2017; Laverghetta et al., 2018; Lennon & Davis, 2018). Having a model of system decision-making will enable generalization (Porter et al., 2020), but we will need rigorous, replicable methods to evaluate the adequacy of the system’s reasoning or “thought” process (Ahner & Parson, 2016; Air Force Scientific Advisory Board, 2017). Traditional formal methods for providing logical or mathematical proof of model adequacy will be insufficient for this and many other AMS purposes (Defense Science Board, 2016; Giampapa, 2013; Goerger, 2004; Kapinski et al., 2016; Menzies & Pecheur, 2005; Micskei et al., 2012; Office of the US Air Force Chief Scientist, 2011).

AMS will require different methods for the verification, validation, and accreditation (VV&A) of M&S, and this may be especially difficult for a system of systems (SoS; Goerger, 2004; Ilachinski, 2017; Lennon & Davis, 2018; Menzies & Pecheur, 2005; Office of the US Air Force Chief Scientist, 2011; Schultz et al., 1993). For example, compositional verification is one method used to get around the combinatorial intractability of complex systems. In this method, individual components are verified, and if they all work on their own, they are assumed to work together. However, this method is intended for deterministic systems, and some argue we will need to adapt it for AMS (Durst & Gray, 2014; Lede, 2019; Luna et al., 2013; Office of the US Air Force Chief Scientist, 2011). Others argue that while compositional verification is a potentially useful step, it is not sufficient alone (Scheidt, 2017). For example, because neural networks produce approximate solutions (Funahashi, 1989), they will almost invariably produce errors. Solutions from decision-making systems of many types may also be fuzzy (e.g., an 84 percent chance this image is a turtle), rather than discrete. These errors and uncertainties can compound and cascade through a network of modules or agents, invalidating a core assumption of compositional verification. Furthermore, testers are estimating module performance, not perfectly measuring it, and so there will be measurement uncertainty as well. Some recommend that all three sources of uncertainty—errors, fuzziness, and estimation—should be propagated when evaluating downstream modules in integrative verification (Porter & Wojton, 2020), but methods for this are still under development (Porter & Wojton, 2020; Stracuzzi et al., 2020).

Methods for disentangling the contributions of different modules or agents to mission success or failure will be particularly critical in AMS (Ahner & Parson, 2016; Baker et al., 2019; Goerger, 2004; Greer, 2013; Harikumar & Chan, 2019; Haugh et al., 2018; Porter et al., 2020; Roske et al., 2012; Sparrow et al., 2018; Visnevski, 2008). This may be as simple as root-cause analysis in any system (Greer, 2013; Haugh et al., 2018; Roske et al., 2012), but it will also extend to teaming with humans or other systems, which is particularly difficult (Goerger, 2004; Porter et al., 2020). For example, if a football pass is incomplete, is it the quarterback's fault or the receiver's? The search for causation must often extend beyond the immediate pair of interest—for example, perhaps there was pressure on the quarterback from a collapsed offensive line. If the only data we record for a team are their shared outcomes (e.g., completed passes), we could need massive amounts of data to make distinct performance attributions.

The increased data demands of AMS will push our methods for test efficiency to the limit. We will need better methods for prioritizing our test points (Ahner & Parson, 2016; Deonandan et al., 2010; Haugh et al., 2018; Hernández-Orallo, 2016; Hess & Valerdi, 2010; Mullins et al., 2017; Sparrow et al., 2018), better ways of understanding the adequacy of our coverage (Ahner & Parson, 2016; Micskei et al., 2012), and better ways of sequentially designing tests to handle exploratory testing of adaptive systems (Ahner & Parson, 2016; Porter et al., 2018; Porter & Wojton, 2020; Simpson, 2020). Compounding this test point prioritization challenge is the need to—potentially continuously—perform regression testing on systems that evolve over time. Testers need methods to help identify when and in what ways regression testing is necessary (Ahner & Parson, 2016; Defense Science Board, 2016; Deputy Secretary of Defense, 2012; Haugh et al., 2018; Ilachinski, 2017; Luna et al., 2013; McLean et al., 2016). For autonomous weapons systems specifically, DoD Directive 3000.09 assigns responsibility for setting these regression testing standards to the Director, Operational Test & Evaluation (DOT&E; Deputy Secretary of Defense, 2012).

Scalability

In some cases, the challenge is not that the methods do not exist, but that the solutions may not be scalable. For instance, formal methods may not scale even where they are applicable (Haugh et al., 2018; Tate & Sparrow, 2019). Using incredibly high-fidelity simulations can get around some VV&A issues for the environment, but these require deep investment in computation resources and may run more slowly than real time (Brabbs, Lohrer, Kwashnak, Bounker, & Brudnak, 2019; Durst, 2019; Kapinski et al., 2016; Kwashnak, 2019; Lenzi et al., 2010; Mullins et al., 2017; Sparrow et al., 2018; Subbu et al., 2009). Other hard problems can be solved with flexible human reasoning, but the scalability challenges are inherent to manual testing or subject matter expert (SME) evaluations (Goerger, 2004; Hernández-Orallo, 2016; Kapinski et al., 2016; Laverghetta et al., 2018; Schultz et al., 1993; Visnevski & Castillo-Effen, 2010; Wegener & Bühler, 2004). Multi-agent testing exacerbates these scalability issues (Baker et al., 2019; Giampapa, 2013; Lenzi et al., 2010; Luna et al., 2013).

Instrumentation

Building automatic embedded instrumentation into the decision software of AMSs could mitigate some scalability challenges, but technical hurdles to implementation remain, and it is not a universal practice (Sparrow et al., 2018). Furthermore, even when such instrumentation exists, it is not always available to testers due to proprietary concerns (Porter et al., 2020). Testers need this instrumentation in order to trace the causes of system behavior (Ahner & Parson, 2016; Defense Science Board, 2016; Haugh et al., 2018; Porter et al., 2020; Sparrow et al., 2018) and to capture environmental conditions in order to replicate problems or findings (Ahner & Parson, 2016; Defense Science Board, 2016; Goerger, 2004; Laverghetta et al., 2018; Visnevski, 2008). The lack of instrumentation intersects most other challenges to providing assurance for AMS, and its criticality may mean testability needs to be a requirement and integrated directly into the system, not an "orange-wire" test harness tacked on solely for test (Ahner et al., 2018; Haugh et al., 2018; Porter et al., 2018).

Simulation

Though many high-level recommendations for AMSs turn to M&S as the solution, many working level challenges must be overcome to enable it. Foremost among these is that of defining the necessary fidelity or resolution of the simulation. Models are necessarily imperfect, but should be useful (Box, 1976, 1979). How much detail is sufficient to be useful for AMSs is unknown at this time (Ahner & Parson, 2016; Caseley, 2018; Goerger, 2004; Sparrow et al., 2018). However, the fundamental tradeoff between resolution and computational power required means playing it safe and using more detail than necessary can impose significant monetary, temporal, and productivity costs (Brabbs et al., 2019; Durst, 2019; Gil & Selman, 2019; Kwashnak, 2019; Sparrow et al., 2018). On the other end, oversimplifying can prevent the simulation from validly recreating system behavior (Goerger, 2004; Greer, 2013; Laverghetta et al., 2018; Lenzi et al., 2010). The optimal balance of these factors will likely be idiosyncratic to individual programs and dependent on the system's maturity (Porter et al., 2020). Furthermore, testers will face the technical challenge of how to feed these systems valid inputs (Ahner & Parson, 2016; Goerger, 2004; Laverghetta et al., 2018), especially for monolithic, sub-symbolic systems like neural networks (Porter, 2020b).

Ranges and Infrastructure

T&E will have a critical need for improved physical and digital ranges on which to develop and test AMSs (Ahner & Parson, 2016; Gil & Selman, 2019; Lennon & Davis, 2018; H. Miller, 2019; Tate & Sparrow, 2019; Zacharias, 2019a, 2019b). Our physical ranges are insufficiently instrumented to reproduce or deconstruct observed system behaviors (Ahner & Parson, 2016; H. Miller, 2019), and in many cases the necessary digital testbeds do not yet exist or have insufficient capacity (Gil & Selman, 2019; Lennon & Davis, 2018). The lack of a common architecture will hinder many of these efforts (Zacharias, 2019b), particularly when we try to test systems together (Porter et al., 2020). There is yet little centralized effort to develop these capabilities (Gil & Selman, 2019), and nascent organizations such as the Joint Artificial Intelligence Center (JAIC) for DoD are at the moment more limited in scope than would be recommended for centralization (Trent, 2019).

Personnel

The need for all of these activities raises the question of who will actually do the work. AI work is inherently transdisciplinary and requires experts in many different domains, essentially all of which are in high demand and short supply, especially within DoD (Macias, 2008; Zacharias, 2019a). The T&E workforce already struggles to attract, train, and retain talented people with the right skillsets, and AI will make existing T&E tasks more complicated, demanding a more sophisticated understanding of the domains we already work in, while simultaneously expanding the types of knowledge our workforce must obtain (Lennon & Davis, 2018; Macias, 2008). In many ways, our tests are only as good as our testers (Wegener & Bühler, 2004). This implies several organizational and personnel challenges, including the identification and development of required skillsets within existing personnel, recruitment of qualified candidates, and certification of T&E methods and practitioners (Ahner & Parson, 2016; Haugh et al., 2018; Roske et al., 2012). Methodological challenges exist as well: inadequate or inefficient traditional test methods may overwhelm our workforce capacity if applied to AMS (Air Force Scientific Advisory Board, 2017).

Challenge #4: Safety & Security

The inherent nature of autonomy implies that use of an autonomous system is often associated with elevated risk, indicating the need to ensure the system is operating within intended bounds and will operate safely in unforeseen circumstances. Further, advances in exploitation and the need to protect the decision model from adversaries can result in complex testing scenarios, presenting challenges for T&E regarding the assurance of safety and security.

Safety

Autonomous systems are often associated with elevated risk. In particular, autonomous systems may be more likely to fail, and the consequences of failure may be worse than with a manned system, because the system's complexity and brittleness can cause failures from which a human could recover to cascade into larger, more consequential problems (Haugh et al., 2018; McLean et al., 2016; Micskei et al., 2012). Autonomous systems impose a need for high degrees of confidence in safety (Laverghetta et al., 2018), so extra safety precautions should be taken during DT and OT (Haugh et al., 2018). Testing must be ongoing, flexible, and broadly protective against these risks (Arnold & Scheutz, 2018; Micskei et al., 2012), and so assurance of safety can contribute significantly to the cost of the system (Deonandan et al., 2010). In some cases, it may simply be impossible or infeasible to obtain approval from the regulatory body to operate certain tests (McLean et al., 2016), which hinders our ability to execute performance testing. Further, the complexity involved in adequately testing autonomous systems generates additional safety considerations that must be addressed when designing the test environment (Ahner & Parson, 2016), a complexity that is complicated by lack of continuous human-in-the-loop control (McLean et al., 2016). Test, evaluation, verification and validation (TEV&V) often relies on human operators to compensate for brittleness (Lennon & Davis, 2018; R&E, 2015) despite such human operators not always being in the loop or being able to react if they are (McLean et al., 2016). TEV&V of AMS will require run-time analysis and some level of self-monitoring, as well as significant prior consideration to address contingencies, and a design that provides some recovery mechanism, or that can at least identify and constrain harmful emergent behavior and nongeneralizable behavior (Ahner & Parson, 2016; Arnold & Scheutz, 2018; Defense Science Board, 2012; Harikumar & Chan, 2019; Laverghetta et al., 2018; Lede, 2019; Lennon & Davis, 2018; Menzies & Pecheur, 2005; R&E, 2015).

Security

The development of autonomous systems brings new challenges to both security and testing. Advances in exploitation have and will continue to introduce a variety of methods to disrupt or alter the desired behavior of autonomous systems (Ahner & Parson, 2016; Eaton et al., 2017; Goodfellow, Shlens, & Szegedy, 2015; Haugh et al., 2018; Qiu et al., 2019; Streilein et al., 2019). This implies complexities in testing; for example, one must test not only the robustness of the autonomy engine against spoofing of its sensors, but also whether the autonomy engine can recognize that its perceived world view has been altered (Eaton et al., 2017). Adversaries will be looking to break or reverse engineer our AMSs' decision models, and thus protecting these models from capture or exfiltration (to make adversarial attacks harder to generate) as well as from adversarial attacks themselves will be critical for security; however, demonstrating these protective capability presents unique challenges for TEV&V (Qiu et al., 2019; Streilein et al., 2019). Inevitably, attackers will find methods to disrupt the algorithms driving the autonomous system, and testing must take into account the likelihood of such attacks (Haugh et al., 2018).

Challenge #5: Lack of Policy, Standards, or Metrics

Adequate TEV&V of AMS relies on metrics and standards for testing as well as policies that support TEV&V frameworks. Unfortunately however, metrics, standards, and policy appear to be lacking in many key areas; participants in TEV&V have had difficulty defining and measuring elements of performance, applied procedures inconsistently, and followed ad-hoc procedures.

Policy

There appears to be no common coherent theoretical framework for T&E with regard to autonomous systems (Durst, 2019; Ilachinski, 2017); instead, there is much disaggregation, with many ad-hoc procedures, bad habits, and loopholes regarding what is being measured and how (Hernández-Orallo, 2016). One reason for this is the ambiguity inherent in conceptualizing a framework for understanding the components underlying the processes that need to be modeled and tested for an unmanned autonomous system (UAS) (Ring, 2009). Regardless of the reason, a generalized framework supporting the design and development of TEV&V approaches could both improve T&E and reduce life cycle costs (Lenzi et al., 2010). Further, policies should support (1) recurring, periodic assessment, (2) recertification, and (3) T&E after regulation changes (Ahner & Parson, 2016).

Metrics

Quantifying the success of an algorithm’s decision-making process poses an ongoing challenge for T&E (Ahner & Parson, 2016), but such quantification is necessary to assess the appropriate level of confidence in an algorithm (Roske et al., 2012). Determining the success of a mission is difficult without appropriate metrics in place to evaluate performance (Deonandan et al., 2010). The lack of formal requirements, along with ambiguities in how failures are defined in developmental testing, leads to a reliance on "I know it when I see it" criteria (Hess & Valerdi, 2010) and a non-replicability issue that makes evaluation of testing results difficult (Laverghetta et al., 2018). This issue extends to measurements of trust, intent, system learning, perception, and reasoning (Ahner & Parson, 2016). Testing the range of system autonomy will likely require application-dependent trust metrics (Air Force Scientific Advisory Board, 2017); however, trust is not an innate characteristic of a system and is difficult to measure. This has led to testing systems in closed, scripted environments without any formal methodology for obtaining trust in the outputs or even defining trust (Durst, 2019; Ilachinski, 2017). These issues are further complicated when testing an Unmanned Autonomous System of Systems (UASoS) (Deonandan et al., 2010). Metrics for testing perceptual accuracy need to be able to distinguish between sensor performance and system inference based on the sensors (Giampapa, 2013; Harikumar & Chan, 2019; Roske et al., 2012). Quantification of system learning is also inherently difficult (Ahner & Parson, 2016). In general, a growing body of literature indicates that traditional metrics are insufficient for TEV&V of autonomous systems (Eaton et al., 2017; Hernández-Orallo, 2016; Ilachinski, 2017; Laverghetta et al., 2018; Macias, 2008; Mueller et al., 2019; Roske et al., 2012; Visnevski & Castillo-Effen, 2010); for example, the probability of hit or kill is frequently measured, but there is rarely (if ever) a question of whether something *should* have been a target in the first place.

Standards

There seems to be a consensus in the literature that there is a lack of standards for TEV&V of autonomous systems. Some authors discuss a lack of benchmarks relevant to the DoD (Hernández-Orallo, 2016; Hess & Valerdi, 2010; Mueller et al., 2019; Roske et al., 2012). Many existing benchmarks, while relevant for comparing one algorithm to another, do not necessarily carry over to system evaluation (Mueller et al., 2019). Further, there appear to be few if any widely used common architectures; however, some standards have been proposed for unmanned ground vehicles (UGVs) and adopted by the international community (Durst & Gray, 2014). The development and adoption of standards for a common modeling framework has also proved challenging (Durst & Gray, 2014; Goerger, 2004; Lennon & Davis, 2018; R&E, 2015; Ring, 2009; Visnevski, 2008). In particular, Lennon and Davis (2018) note the lack of modeling, design, and interface standards, and R&E (2015) notes that a standardized modeling framework spanning the lifecycle for autonomous systems does not exist. Further, non-standard criteria contribute to an inconsistently applied validation process (Goerger, 2004), the ambiguity in UAS modeling framework conceptualization creates problems for development (Ring, 2009), and the lack of centralization and non-compliance with standards can impede the ability of users to monitor complex systems (Visnevski, 2008).

Challenge #6: Human-System Interaction

The nature of autonomous systems implies that they can accomplish at least some of their assigned tasks without human direction, but this does not imply that they must never interact with humans. Indeed, AMSs often require some level of human interaction with their task, whether the human is in-the-loop, (offloading some of their cognition to the AMS but still performing the task), on-the-loop (allowing the system to perform the tasks while monitoring its performance), or merely initiates-the-loop (giving initial orders but thereafter letting it act with autonomy). These different types of human interaction may require different test methods than those typically executed in the DoD. Furthermore, effective system employment will require that operators appropriately trust their systems. If we expect operators to have some kind of C2 relationship with their autonomous systems, that relationship needs to be tested.

Trust

It is common for DoD draft documents to recommend design, development, and test practices to *increase* trust in AMS—however, there is consensus among experts that this is the wrong way to frame the problem. What we need is for operators to *appropriately calibrate* their trust—to know when, where, and the extent to which they can or cannot rely on the system (Gunning, 2017; Haugh et al., 2018; Hoff & Bashir, 2014; Lee & See, 2004; Porter et al., 2020; Wojton, Porter, & Lane, 2020). While conversations about trust often suffer from the lack of a common definition (Durst, 2019), here we refer to trust in autonomy as the belief that a system will help accomplish one's objectives in vulnerable or uncertain situations (Wojton et al., 2020). This belief is a psychological state that is distinct from the system-level trait of *trustworthiness* (Porter et al., 2020). Design and development practices should seek to maximize trustworthiness; the operator-level goal should be appropriate trust (Tate, 2020).

Both over- and under-trust are potentially problematic with autonomous systems (Lee & See, 2004): in the field, both relying on the system where one should not and ignoring the system where it could provide benefit can cause undesirable outcomes. Operators need to know the extent to which they can or cannot trust the system under different conditions. T&E of system performance is where we obtain the information to provide this conditional understanding, but testers must also assess whether training and experience with the system actually create this appropriate calibration of trust (Porter et al., 2020). In order to achieve this it may be necessary for operators to have a valid mental model of how the system makes its decisions (Endsley, 2019; Gunning, 2017), which will usually require testers to discover what that valid model is (Porter et al., 2020).

Teaming

Although human-machine teaming (HMT) is topic rapidly growing in popularity, a large component of this conversation may simply be a buzzword rebranding of traditional human-system integration (HSI). Testers need to be careful when assessing whether their AMS truly involves HMT, because the interaction of humans and systems pursuing common higher-level goals and coordinating their actions (i.e., teaming) introduces novel testing challenges that require novel methodologies beyond normal HSI (Porter et al., 2020). For example, effective teaming is enabled by a shared understanding of goals, situational awareness, and each other's actions, but measuring the extent of this alignment between humans and machines is non-trivial (Ahner & Parson, 2016; Haugh et al., 2018; Ilachinski, 2017; Lennon & Davis, 2018; Porter et al., 2020). Additionally, more work is needed to develop metrics for team performance (Ahner & Parson, 2016; Deonandan et al., 2010; Greer, 2013; Haugh et al., 2018; Ilachinski, 2017) as well as methods for disentangling individual contributions that support or detract from this performance. When discussing emergent behavior, authors are typically concerned with the systems themselves, but testers also need to be vigilant about how introducing AMS can create undesirable emergent *human* behavior (Harikumar & Chan, 2019; Ilachinski, 2017; Porter et al., 2020).

Human Judgment & Control

When a system's CONOPs calls for a human to have a C2 relationship with an AMS (e.g., in-, on-, or initiating-the-loop), testers must assess whether the human can appropriately or meaningfully influence the AMS. Although there is a great deal of debate over whether the quality these relationships should possess is *meaningful human control* or *appropriate human judgment* (Deputy Secretary of Defense, 2012; Horowitz & Scharre, 2015; Santoni de Sio & van den Hoven, 2018), and these arguments can become heated (Cook, 2019), from a testing standpoint, these distinctions are less relevant. Testing should assess whether the system and interactions with it (e.g., TTPs) actually accomplish the intent of the CONOPs. There are many reasons why a human might technically have oversight, control, or assigned judgment over its use without that being appropriate or meaningful. For example, it is bad engineering to rely on humans in long-term vigilance tasks, where they often grow complacent or inattentive. When this happens, their supervision of that task would no longer be meaningful (Ahner & Parson, 2016; Caseley, 2018; Porter, 2020a), as was seen in the fatal Uber crash in Arizona (National Transportation Safety Board, 2019). Similarly, if systems' decision cycles occur faster than human reaction times, a human supervisor would not be able to meaningfully intervene in the event of errors (Arnold & Scheutz, 2018; Caseley, 2018; McLean et al., 2016). Often the inability to exert appropriate judgment will result from the human's inability to predict the system's behavior (Ahner & Parson, 2016; Arnold & Scheutz, 2018; Ferreira et al., 2013; Gunning, 2017; Ilachinski, 2017; Subbu et al., 2009), and testers will need to examine this. Finally, humans are not always aware of how information affects their decision making (Nisbett & Wilson, 1977), and if this happens with systems such as tactical response recommenders or automatic target recognition, it calls into question whether operators can still exert appropriate human judgment over the use of force. Testers would need to assess if and where these "cognitive prosthetics" negatively affect human decision-making (Ghassemi, 2020; Porter, 2020a).

Summary of Challenges

The complexity of AMS is the root of many of our other challenges. The state-spaces and stochasticity of these systems mean there is simply too much to test exhaustively or with our traditional methods. Other aspects of complexity and system novelty prevent us from knowing exactly what or how to test. This implies that both development and testing will need to be more of an experimental effort like what happened early for what are now well-understood systems (e.g., airplanes and nuclear reactions), where much of it is learned as we go. However, this kind of experimentation has two major challenges. First, the rigidity, sluggishness, and firewalls in the DoD acquisition process prevent adaptive, rapidly responsive, sequential testing. Secondly, even if DoD processes could be changed, at a technical level we lack the methods, tools, infrastructure, and personnel to effectively prosecute this testing. This exacerbates the problem of attaining enough assurance to be confident even that the testing itself will be safe to perform. Given the scarcity of expertise in the AI arena, especially within the DoD, having policies, metrics, and standards to draw on could accelerate individual experimentation efforts; unfortunately, by and large these are not currently in place, and those that are present are subject to change. Any one of these problems would be difficult on its own, and the combined challenges are daunting. Fortunately however, there may be solutions to most of these issues—at least enough to get started. In the next section, we summarize the recommendations various authors have made to enable the T&E of AMS.

T&E Recommendations for Autonomy

A common call-to-arms at AI-related conferences, meetings, and working groups is that "it is time to stop admiring the problem and move towards concrete, implementable solutions." However, anyone who participates in these same events has likely observed concrete, implementable solutions being attacked without a counter-proposal, and so conversations rarely progress past problem portraiture. While it is easier to cast stones than build houses, criticism alone will not save us. A meta-recommendation for recommendations is that they should always be solution-oriented: they should either propose a different solution to the same challenge, or recommend a way to overcome the shortcoming they point out (Porter et al., 2020). As a community, we testers need to be better at moving forward. The acquisition of AMS does and will proceed unabated despite the lack of readiness of TEV&V, and we cannot wait for perfect solutions. The recommendations described in this section are imperfect, but that does not mean they should be abandoned; instead we should ask how they can be improved. At the same time, the authors did not include only recommendations they agreed with, so readers should not necessarily interpret inclusion as endorsement.

Recommendation #1: Requirements, Design, & Development Pipeline

Ultimately, the goal of TEV&V is to provide assurance that the system will be adequate for the operational uses and challenges it will encounter. A number of activities that occur before formal testing can help us enable the provision of assurance, and may even contribute directly to our confidence in the system. Getting the requirements right will be critical, and we may need to consider alternative methods for evaluating whether they have been met. A number of design choices are available that enhance the system's testability and help us to successfully use these alternative methods. Parallel to those design choices, there are development practices that integrate system testing more tightly with the development processes. All three of these recommendations will likely require much more significant coordination than is currently the norm between testing and development activities—while still maintaining the ability of oversight activities to provide independent evaluations.

Requirements

Part of minimizing the risks of fielding unworthy AMS will be writing requirements that allow us to reject such systems. Traditional requirements typically include technical specifications that, while important, do not define operational success. In a manned system, much of the operational success comes from how the operator wields these technical capabilities, and so it does not need to be a system requirement. However, autonomous systems must in some sense wield themselves, so operational success needs to be built into these systems, which then implies that operational success must/should be a system requirement (Porter et al., 2020). Defining these requirements may be particularly difficult for ethically-related behaviors, and the generation process for these requirements should additionally include SMEs such as lawyers and ethicists (Porter, 2020a). Additionally, although these systems are meant to be autonomous, requirements for human-system integration may need to be emphasized more than ever (M. J. Miller, McGuire,

& Feigh, 2017). Furthermore, some recommend that DoD should write all requirements, both operational and technical, in the form of testable, verifiable hypotheses, which will likely require alterations to the requirements generation process (Ahner & Parson, 2016; Lede, 2019; Lennon & Davis, 2018).[4] The general challenges of AMS may mean requirements are difficult to verify through purely quantitative or formal methods. Some are therefore recommending that we adopt a more holistic approach of "assurance arguments" that integrates multiple methodologies and sources of confidence—including less traditional ones—to evaluate whether requirements have been met (Lede, 2019; Lennon & Davis, 2018; Scheidt, 2017; Sparrow et al., 2018).

Assurance Aiding Designs

There are many design choices for AMSs that will make it easier to validate their safety and performance. Design choices could make them easier to test, for example by providing mechanisms for better data collection on the system, or by making their performance envelopes clearer by means such as explicitly bounding the systems' behaviors.

Safety Middleware

Some are recommending (Arnold & Scheutz, 2018) or even implementing (Thuloweit, 2019) the use of safety middleware to bound the behavior of the more complex and unverified AMS software. The concept is that by providing a middle-tier supervisory system that can supersede the true system's decisions under certain explicit, well-defined conditions, then the argument that Behavior X cannot occur under Condition Y becomes exponentially easier to verify. For example, something as simple as geofencing could activate this supervisory system to prevent crashes (intentional or accidental) near critical installations or prevent unintentional weapons discharge at test sites.

While these safety middleware systems could provide great risk-reduction value, developers and testers should avoid viewing them as a panacea, especially for actual operational control. If these simple middleware systems were sufficient to guarantee correct behavior across the AMS system's entire operational space, then we would simply be using them instead of the more complex system. Additionally, there are control and perception problems to which these simpler systems are not well-suited or would create other problems. For example, success has been historically elusive with *simple*, explicit attempts to execute machine perception (e.g., computer vision; Voulodimos, Doulamis, Doulamis, & Protopapadakis, 2018).

The benefits of safety middleware are much clearer for T&E than for actual operations. Explicit rules in middleware would be more vulnerable to both cyber and tactical exploitation. For example, if the middleware prevented the system from targeting anyone with a reverse-faced

[4] Some argue that it is already standard practice to write testable requirements, but that standards are too high for practicality. For example, a requirement for a probability of hit of 50% is theoretically testable and verifiable, but in reality it takes 96 shots to determine a 95% confidence interval with at most a 10% margin of error. This might be prohibitively expensive for live testing. (Haman, 2020).

American flag on their right shoulder, it would provide incentive for adversaries to engage in perfidy and wear this patch. Trying to combat these issues by adding more branching logic or competing supervisory controllers quickly spirals into a "*quis custodiet*"[5] problem (Porter et al., 2020). If these middleware systems are intended for operational use, testers must ensure they are an explicit component of the evaluation.

However, while more supervisory middleware can help make tests safer, the T&E community should keep in mind that their use may restrict testing of real parts of the operational space. If the middleware always kicks in under its specified conditions, then it becomes difficult to know what the more complex system would have done under those conditions. For example, if the middleware is something like an automatic ground collision avoidance system, it is harder to tell if a drone's high-speed dive was (or the drone believed it was) recoverable and/or related to a tactical decision.

Built-in Transparency and Traceability

If systems are recording data about their own decisions and internal processing, then stakeholders, including developers, testers, and even users, can gain more transparency into the system. From a TEV&V perspective, this instrumentation could be combined with safety middleware or disabled functionality to execute what some call "shadow testing," where the complex system makes decisions about what it *would* do in the current situation without being allowed to implement or execute those actions (Templeton, 2019).

Shadow testing, combined with a strategy of graded autonomy (slowly stepping up the permitted risks of unsupervised tasks, as with medical residents) and limited capability fielding (only initially certifying and enabling a subset of existing capabilities for fielding) could allow the services to get at least some useful functionality into warfighters hands while continuing the T&E process for features with a higher evidentiary burden (Porter et al., 2020). Once fielded, shadow testing on disabled capabilities could harvest data from actual operations and exercises to evaluate higher-risk tasks under realistic conditions without accepting that higher risk. However, the obvious critical requirement underlying this is internal instrumentation.

Many point to a need for systems to be instrumented such that the internal causes of the system's decisions are traceable, and assert that this instrumentation needs to be implemented throughout the entire lifecycle of the AMS (Ahner et al., 2018; Haugh et al., 2018; Luna et al., 2013; Porter et al., 2020; Sparrow et al., 2018; Zacharias, 2019b). Typically instrumentation is done as an "orange wire" harness solely for TEV&V activities and only covers hardware and sometimes elements of traditional software. While this traditional instrumentation is necessary, programs also need to include "cognitive instrumentation" on the internal decision processes (Haugh et al., 2018). This cognitive instrumentation could serve a host of activities, including fault diagnosis, live behavioral health monitoring, status reporting and explainability to operators, intrusion detection, model induction, data harvest for retraining, and even be reversed to allow

[5] "*Quis custodiet ipsos custodes?*" Latin for "Who will guard the guards themselves?"

injection for live, virtual, and constructive (LVC) events (Haugh et al., 2018; Porter et al., 2020). To ensure that this instrumentation is available from cradle to grave, it may be advisable to have it be an explicit program requirement for all AMS (Ahner et al., 2018; Porter et al., 2020).

Explainability

Several DoD policies require that AMSs be at least *transparent* to our warfighters (Defense Innovation Board, 2019; Deputy Secretary of Defense, 2012; Lopez, 2020); others call for a further step and advocate that they be *explainable* as well (Gunning & Aha, 2019; Haugh et al., 2018; Mueller et al., 2019). Though there are many different definitions of these terms, here we use transparent to indicate that a system's current operating state can be known and its behavior reasonably predicted from that state, while we use explainable to indicate that a system is capable of providing the reason for a behavior post-hoc. Though most focus on the value to operators of having explainability, it also provides a powerful tool for testers (Haugh et al., 2018). *Why* a system makes a decision is much more valuable for generalizing its behavior than repeated measurements of *what* it did (Porter et al., 2020), especially for legal, moral, and ethical behavior (Porter, 2020a), and a system that can explain its decisions can provide this "why" to testers. Functionally, a system that can explain itself is one that can trace its own decisions, and so explainability will be critically enabled by traceability (and in turn by the recommendations that support that feature).

"Common, Open Architectures with Reusable Modules"

In the discussions around AMS, both recommendations for common, open, modular architectures to support reusability (Air Force Scientific Advisory Board, 2017; Caseley, 2018; Defense Science Board, 2016; Zacharias, 2019a), and criticisms of those recommendations, (Atherton, 2019) are common. While these features of AMS design are mutually reinforcing, and they are often called for together—and thus conflated—the benefits and criticisms of each modifier are different and worth examining separately.

Modularity

In this review, we use modular to describe a feature of system architecture where a sub-system or component (i.e., a module) of the larger architecture can take certain inputs and, through its information processing or service, produce new outputs to be passed on to other modules. This contrasts with *monolithic* systems at the other end of the spectrum, where there is an absence of architecture and no structure or sequence to inputs and outputs (Porter et al., 2020). While modern development practice for ordinary software virtually guarantees some level of modularity, AI-enabling software may be much more or entirely monolithic, particularly when it is powered by neural networks trained with supervised, unsupervised, or reinforcement learning (OpenAI, 2018). However, many recommend that AI-enabling software be constructed around modularized cognitive architectures instead (Achler, 2013; Anandkumar, 2020; Chella, Cossentino, Gaglio, & Seidita, 2012; Krichmar, 2012; Kurup & Lebiere, 2012; Laird, 2012; Polk & Seifert, 2002; Samsonovich, 2012; Sandamirskaya & Burtsev, 2015; Simen & Polk, 2010). The extent of modularity is a gradient, not a binary attribute, and the level of granularity at which a system is modularized will likely be a program-specific choice (Defense Science Board, 2016; Laird, 2012; Porter et al., 2020).

Some recommend modularization primarily to address development or performance concerns (Air Force Scientific Advisory Board, 2017; Defense Science Board, 2016), but there are significant T&E benefits as well. First, modularization breaks down the system and its decisions and tasks into greater and more explicit granularity (Roske et al., 2012), which makes it easier to (1) write verifiable requirements for the system's capabilities (Scheidt, 2017), (2) design tests that focus on those tasks, and (3) create the hooks both for cognitive instrumentation and supervisory middleware (Porter et al., 2020). However, this can create the possibility of emergent effects internal to the system, and so intra-system emergence needs to be tested as well (Porter et al., 2020).

Modularity continues to be a fierce debate outside of T&E, however. Some critics contend that trying to modularize the AI controlling our autonomous systems creates insurmountable interoperability challenges (Atherton, 2019). Others criticize the recommendation by pointing to a general tradeoff between modularization and the ability to optimize the system, which potentially creates an operational performance tradeoff. However, more recent work in industry suggests this might be a false tradeoff: complex real-world tasks (i.e., the ones that matter with AMS) can benefit more from generalization than optimization, and bio-plausible architectures of neural network modules improved performance across the breadth of an operational space compared to more monolithic designs (Anandkumar, 2020). Optimizing to a small set of training data (i.e., overfitting a model) can also create performance problems, and modularity can help provide some robustness against this (Porter et al., 2020).

Open Architectures

In addition to recommending that systems have some kind of modular architecture, numerous groups are recommending that the specific system implementations be based on open, non-proprietary architectures that could be reused across multiple systems (Air Force Scientific Advisory Board, 2017; Eaton et al., 2017; Lenzi et al., 2010). Though most hope for operational interoperability and cost-savings from reusability with these open architectures (Defense Science Board, 2016; Zacharias, 2019a), or for protection against individual vendors failing at performance requirements or otherwise choosing to end participation (e.g., Google and Project Maven; Mitchell, 2019), there has also been some discussion of the T&E benefits. Open architectures help ensure some level of modularity and commonality for cognitive instrumentation and supervisory middleware, which will be helpful for *T&E* interoperability. Test interoperability will be especially desirable when testing systems-of-systems (Eaton et al., 2017; Lenzi et al., 2010) or looking for emergent behavior between systems not only in live but especially in simulated environments (Porter et al., 2020).

Capability Reuse

The topic of reusability is extremely controversial. From a T&E perspective, reuse reduces the amount of evidence that needs to be collected to make a credible assurance argument. There are calls to make risk-based certification and reuse of capability modules the standard model for AMS (e.g., Caseley, 2018). On the other end of the spectrum, some think the DoD should not even pursue modularity at all (e.g., Atherton, 2019). Some view reusability as desirable but unachievably aspirational for the moment, as even usability is a yet unsolved problem (Sparrow, 2020), or think that certifications will need to be case specific, and so the T&E savings may be overstated (Sparrow et al., 2018). Others point out that if the way a system makes decisions is not sufficiently robust to transfer to related problems, even if only as a starting point for transfer learning, then it is unlikely to be sufficiently robust across its own complex operational space in the first place. Some level of reusability will be a feature of robust, useful systems (Porter et al., 2018). Furthermore, AMS will have a higher evidentiary burden than standard systems, and boutique development solutions alongside custom T&E may not be viable from an enterprise cost perspective, making reusability potentially more necessary for AMS than for standard systems (Porter et al., 2018).

Design & Development Processes

Some design and development practices are better complements to testing than others. Designers who conduct *a priori* risk analyses to inform their designs, such as considering the type and likelihood of different destabilizing events and the advantages and risks of introducing autonomy there (Harikumar & Chan, 2019), are more likely to explicitly address those concerns (Ferreira et al., 2013) and to prioritize test points surrounding those risks (Deonandan et al., 2010). Conducting these risk analyses can improve testing and even become part of an assurance argument themselves. Other common process recommendations are to adopt or develop paradigms that closely integrate testing early on and throughout the development process, such as Model-Based Design (Ahner & Parson, 2016; Kapinski et al., 2016), statistical engineering (Ahner & Parson, 2016), or iterative or cyclic paradigms like Agile or DevSecOps (Air Force Scientific Advisory Board, 2017; Defense Science Board, 2016). Some recommend the use of digital twins—simulated copies of the cyber-physical system—operating in parallel to the real system for run-time assessment throughout the system's lifecycle (Arnold & Scheutz, 2018), something which could be particularly useful for shadow testing and discovering performance deltas between versions when embedded in the same chassis.

Coordinate, Integrate, & Extend Activities

It has been noted both informally and formally that DoD acquisition processes are ill-suited to implementing the development paradigms above. Firewalls—implemented by both law and custom—between contractor testing (CT), developmental testing (DT), and operational testing (OT) activities enforce a waterfall paradigm[6] across the system's lifecycle. Many attempts to adopt something like an Agile process merely result in "Agile BS," where the *go fast* principle is adopted, but the integration of testing and willingness to move backwards and correct issues discovered by it are not (Defense Innovation Board, 2018). In light of these issues, there have been numerous calls to desegregate the stovepipes of CT, DT, and OT to some extent (Ahner & Parson, 2016; Air Force Scientific Advisory Board, 2017; Deonandan et al., 2010; McLean et al., 2016; Porter et al., 2018; Roske et al., 2012). However, some of these firewalls exist for important reasons, and so methods that still preserve the independence of these organizations, such as the

[6] A category of linear development processes where a project is divided up into stages, and each stage completes their work and then provides what they have done as a deliverable to the next stage.

creation of cross-functional teams, may be preferable (Porter et al., 2018), because independent oversight is a likely feature of ethical system development (Porter, 2020a).

Desegregating these and later activities will likely be a necessary step to enable other recommendations. For example, some recommend earlier user involvement (Defense Science Board, 2012; Gunning, 2017; Mueller et al., 2019) or point out that effective development will require earlier operational realism in testing (Defense Science Board, 2012; Porter et al., 2018). However, access to the necessary resources and expertise is typically downstream from the time at which these processes need to occur. Furthermore, testing does not just have to shift left—it may also have to shift right and continue through post-fielding activities: many are recommending that the TEV&V process should not end when systems are fielded, but has to be a cradle-to-grave process, especially for learning systems that continue to adapt (Ahner & Parson, 2016; Defense Science Board, 2016; Deonandan et al., 2010; Ilachinski, 2017; Porter et al., 2020). This would require a closer association between test organizations and the Services. Finally, there is strong consensus that we should be using iterative, adaptive, or sequential methodologies for development and testing, where the path forward cannot and should not be charted years in advance, but rather needs to react to the shifting realities on the ground and even return to earlier stages when necessary (Ahner & Parson, 2016; Defense Science Board, 2016; Gunning, 2017; McLean et al., 2016; Micskei et al., 2012; Mueller et al., 2019; Porter et al., 2020; Simpson, 2020; Sparrow et al., 2018; Visnevski & Castillo-Effen, 2010). A firewalled waterfall approach cannot support these needs, and so recommendations to reform that approach typically follow.

Recommendation #2: Methods, Tools, Infrastructure, & Workforce

Authors frequently made recommendations alongside their observations that testers do not have what they need to effectively execute their duties. In most cases, authors call for investment or research in particular domains. Others make more specific suggestions for techniques that could be beneficial at least as starting points.

Leverage Existing Methods

Despite a common acknowledgement that our current methods are insufficient for the challenges of AMS TEV&V, few imply that they are without any value or that we need to start completely from scratch. For some uses, the current methods *are* adequate, and for some others, methodological solutions to AI challenges already exist in other fields, but simply are unknown or unused in DoD. Wherever possible, we should adopt those techniques when they are adequate, adapt them when they are not, and only invent new ones when these conditions cannot be met (Porter et al., 2018). Many of the challenges of AMS will be the same as those we have already solved for any complex system, and they can be directly lifted (Caseley, 2018; Roske et al., 2012). For instance, although formal methods are difficult to apply to many components of AMS, there will be times when testers can use them to significantly bolster an assurance argument (Air Force Scientific Advisory Board, 2017; Haugh et al., 2018; Luna et al., 2013; Tate & Sparrow, 2019). Though some critical statistical methods used in test planning require adaptation or invention, many of the techniques in testers' toolboxes will continue to be relevant for AMS (Porter et al., 2020; Porter et al., 2018; Roske et al., 2012).

Performance Testing Methods

Some authors have made suggestions for specific techniques that could help characterize the performance of AMS systems or their subcomponents. Some recommend using compositional verification on individual modules, for example (Ahner & Parson, 2016; Haugh et al., 2018). Others point out that while compositional verification is a useful starting point, integrative verification will be necessary for non-deterministic (or just more generally unpredictable) networks of modules (Durst, 2019; Harikumar & Chan, 2019). For example, modules can be tested in successive forward and backward cascades, and a Bayesian paradigm could allow statistical and module uncertainty to be propagated when evaluating these module chains (Porter et al., 2020).

To characterize that performance, authors have recommended a variety of different outcome metrics that, in whole, could be combined to be mutually reinforcing. Because of the focus on narrow applications of AI, most of the metrics used today are domain- or task-specific outcomes, and some recommend that collection of these continue (Baker et al., 2019). However, as we move towards more complex problems that might require the accomplishment of many subtasks, authors also recommend that domain-general skills might make for stronger generalization across these broader problems (Baker et al., 2019; Durst, 2019; Harikumar & Chan, 2019; Hernández-Orallo, 2016). For a human domain-general skill example, we might measure the speed at which someone can acquire new information and his or her ability to multi-task in order to judge how well they will perform in a demanding, dynamic task environment, rather than their ability to do all specific tasks that might arise. These domain-general skills might be hard to conceive or measure, and so others have suggested that domain-specific skill growth trajectories could be characterized as well (Ahner & Parson, 2016; Baker et al., 2019).

Prioritizing Test Points

There is consensus that limited test resources spread across the large state-spaces requiring coverage mean that we will need to prioritize our test points and maximize their efficiency, and some authors have made more specific recommendations to that effect. For example, one could optimize to get coverage of risky conditions identified *a priori* (Deonandan et al., 2010), or more generically, optimize based on the resources the test plan would save (Luna et al., 2013). Recommendations often focus on adopting or adapting some kind of optimal Design of Experiments (DOE; Air Force Scientific Advisory Board, 2017; Haugh et al., 2018; Hernández-Orallo, 2016; Hess & Valerdi, 2010; Porter et al., 2020; Roske et al., 2012), or indicate that testers should choose points adaptively or sequentially based on what has been learned to date (Hess & Valerdi, 2010; H. Miller, 2019; Porter et al., 2020; Visnevski & Castillo-Effen, 2010). Some work has been completed integrating these suggestions for the statistical implementation of sequential DOEs for autonomous systems (Porter et al., 2020; Simpson, 2020).

Developing Methods

Where the current methodological state-of-the-art will be insufficient for AMS, the cutting edge needs to be advanced. The problem area that method development recommendations most commonly target has been choosing or defining what the test dimensions or conditions should be, for example the test factors around which a DOE is run, and selecting test scenarios or cases (Ahner & Parson, 2016; Arnold & Scheutz, 2018; Defense Science Board, 2012; Hernández-Orallo, 2016;

Lenzi et al., 2010; Scheidt, 2017; Zhou & Sun, 2019). Some suggest that adapting our current DOE methods or inventing new ones should be the focus of this research (Ahner & Parson, 2016; Air Force Scientific Advisory Board, 2017; Porter et al., 2020; Simpson, 2020).

Beyond picking test dimensions or points, several authors point to the need for both broader experimental paradigms and more specific techniques for demystifying AMSs' internal decision models (Ahner & Parson, 2016; Gunning, 2017; Porter et al., 2020). For example, some have more generally recommended the use of symbolic meta-models (Alaa & Schaar, 2019) or visualization techniques (e.g., Heinrich, Zschech, Skouti, Griebenow, & Riechert, 2019), but their applicability and scalability for AMS remains untested. In many cases, these demystification techniques produce more abstract versions of the system's decision making. Relatedly, testers need to develop methods for determining the appropriate abstraction of M&S (Ahner & Parson, 2016), which likely will require some level of demystification (Porter et al., 2020).

AMS will produce massive quantities of data, especially if the system has any kind of cognitive instrumentation. Aside from the enterprise infrastructure these data flows demand, we will also need methods and procedures for managing, handling, and analyzing these data at the local T&E level (Ahner & Parson, 2016; Visnevski, 2008). While "big data" is not a unique problem to AI or the DoD, the Department has historically and continues to struggle with this challenge, and commercial solutions do not always transfer to military problems (Avery & Simpson, 2019).

Develop Test Tools

Many are calling for tools that will help automate testing, either to automatically evaluate performance (Micskei et al., 2012; Wegener & Bühler, 2004; Zhou & Sun, 2019), or for exploring the operational space (Air Force Scientific Advisory Board, 2017), both of which are likely necessary to implement the DevSecOps framework the JAIC intends to pursue (Pinelis, 2020; Trent, 2019). Several groups recommend that rather than trying to invent these tools from scratch for AI, we should continue to iterate and extend the automated test tools we already have (Air Force Scientific Advisory Board, 2017; H. Miller, 2019; Scheidt, 2017; Streilein et al., 2019).

Develop Test Infrastructure

Many recommend significant investment in improving our test assets and ranges. Often cited is the need to develop digital, M&S, or LVC test beds (Ahner & Parson, 2016; Air Force Scientific Advisory Board, 2017; Eaton et al., 2017; Gil & Selman, 2019; Haugh et al., 2018; McLean et al., 2016; Micskei et al., 2012; Porter et al., 2020; Streilein et al., 2019). Others recommend improving our physical ranges via improved instrumentation (Defense Science Board, 2012) or range realism (Defense Science Board, 2012; Gil & Selman, 2019).

Establish Partnerships

Some recommend centralizing the research and development of these methods, tools, and infrastructure (Gil & Selman, 2019; Hernández-Orallo, 2016). Even if research is not centralized, establishing partnerships between stakeholders and sectors could help combat stovepiping and the scattering of talent across groups. There are many different sources of expertise across industry, academia, and the national security community; tighter relationships between these sectors could enable cross-pollination of ideas (Air Force Scientific Advisory Board, 2017; Gil & Selman, 2019), and a central organization or activity dedicated to sharing AMS test techniques and lessons would help ensure this knowledge is disseminated, or at least available, to testers who need it (Gil & Selman, 2019; Hernández-Orallo, 2016).

Personnel

In order to actually execute the more sophisticated test and analysis techniques AMS will require, DoD will need to improve the skillsets of its workforce. Though the approaches are not mutually exclusive, some recommendations focus on creating pipelines for the needed skillsets from universities to the government (Gil & Selman, 2019), whereas others emphasize retraining our current employees (Ring, 2009) such as accrediting employees in AI skillsets through continuing education units (Goerger, 2004).

Recommendation #3: Specific Test Strategies

Some authors, instead of speaking generally about the TEV&V of AMS, have proposed specific testing frameworks. Though there has been a great deal of variation in their implementations and intervention points in the system's lifecycle, as well as many more individual recommendations we do not cover here, a number of themes emerge from these frameworks. In this section we cover the overlap among people's proposals for how to test AMS. Note that while our descriptions here are somewhat generic, the actual strategies often get into implementation-level details.

Develop a Strategy Based on Existing Practices

Many, including those who make specific proposals of their own, recommend that test strategies ground themselves in some existing field or practice as a starting point. These include suggestions that the behavioral sciences could offer useful insights into how to examine systems that will make their own decisions and perform their own behaviors (Durst, 2019; Goerger, 2004; Gunning, 2017; Porter et al., 2018); that many of the problems in AI overlap those in any complex software, and so we should start there (Caseley, 2018; Durst & Gray, 2014; Harikumar & Chan, 2019; Ilachinski, 2017; Roske et al., 2012), and that our current T&E practices should be advanced, not abandoned (Defense Science Board, 2016; Durst, 2019; Roske et al., 2012).

Automate Testing

Because of the consensus that the huge state-spaces in which AMS will operate will challenge our ability to test, many are also seeking ways to improve efficiency. A core suggestion uniting many testing frameworks is to significantly increase reliance on automated testing. Instead of having test planners run their own DOEs, for example, automated test tools could allow a program to automatically select the next test points in a sequential manner based on pre-specified criteria, allowing much more efficient coverage of the state-space (Haugh et al., 2018; Micskei et al., 2012; Visnevski & Castillo-Effen, 2010; Wegener & Bühler, 2004; Zhou & Sun, 2019). Test tools could also execute some level of automated performance evaluation on test outcomes (Micskei et al., 2012; Wegener & Bühler, 2004; Zhou & Sun, 2019), including training evaluation systems with machine learning to analyze AI performance (Micskei et al., 2012). Critical to enabling this automatic evaluation would be creating “test oracles” that know what the correct outcome should be for a given situation, and the problems where this is hard vastly outnumber the problems where it is easy (Sparrow et al., 2018). When combined, automated DOE and performance evaluation can permit automatic testing. However, if this is done via live test, where most of the budget is driven by logistics and physical execution, then the cost and time savings are likely minimal, and so many also recommend that testers rely heavily on simulation (Air Force Scientific Advisory Board, 2017; Defense Science Board, 2012; Micskei et al., 2012; Visnevski & Castillo-Effen, 2010; Wegener & Bühler, 2004).[7]

Low Fidelity Coverage, High Fidelity Validation

The extensive use of simulation begs the question of the nature of those simulations. There is a fundamental tradeoff between simulation fidelity and run speed. If the goal is to cover as much of the state-space as possible with these techniques, then low fidelity (LoFi) simulations would maximize that capability. However, LoFi simulations give much less credible assurance and are more likely to miss complex but dangerous interactions than high fidelity (HiFi). The compromise that most frameworks propose is to use both LoFi and HiFi simulations for their different strengths: achieve broad, efficient coverage with LoFi techniques to look for zones of interest, while using HiFi sims or live testing to follow up on those zones, validate the LoFi results, and/or assess critical test points. While this is hardly a novel strategy overall, there are different recommendations for its specific implementation in AMS.

There are different forms the LoFi simulation could take. For example, testers can reduce the fidelity of the environment and/or the AMS software itself. Some recommend using some simplification or abstraction of the system’s decision model for LoFi testing (Giampapa, 2013; Greer, 2013; Haugh et al., 2018; Micskei et al., 2012), or the use of agent-based modeling (Defense Science Board, 2016; Greer, 2013; Ilachinski, 2017). Others recommend something closer to software-in-the-loop testing where the actual decision software is embedded in a LoFi representation of the environment (Sparrow, 2020).

Once a LoFi simulation method is selected, testers also need to pick ‘zones of interest’ for additional testing, especially in automated test paradigms. Many have recommended that performance failures be the primary driver of follow-on testing (Deonandan et al., 2010; Giampapa, 2013; Greer, 2013; Haugh et al., 2018; Luna et al., 2013; Schultz et al., 1993; Subbu et al., 2009; Visnevski & Castillo-Effen, 2010; Wegener & Bühler, 2004). Others suggest that rather than simply finding failures, edge cases that define where performance boundaries change would provide the best value for additional investigation (Durst, 2019; Haugh et al., 2018). Some focus on the idea of unpredictability. For example, some recommend a test replicate paradigm where retested points that produce inconsistent results are the most interesting (Harikumar & Chan,

[7] Sequential or integrated testing can also just simply be more difficult to plan. Decision-makers and program planners are less likely to agree to a test that costs an indeterminate amount of money, even if that unknown amount is likely to be smaller at the end of the day (Haman, 2020).

2019; Zhou & Sun, 2019). This retest paradigm assumes the same simulation is used for both, whereas others recommend multi-method validation paradigms where zones of interest are those where one LoFi method cannot predict another (Porter et al., 2020).

Once these zones of interest are defined, testers can move through progressively more sophisticated tests, such as higher resolution simulations, software- and hardware-in-the-loop tests, LVC activities, and fully live testing (Defense Science Board, 2016; Laverghetta et al., 2018; Visnevski & Castillo-Effen, 2010). Some recommend that rather than a progressively increasing linear development of resolution, these can be iterative, cyclic processes that return to each other to build up and out (Porter et al., 2020).

Build a Body of Evidence

Embedded in some of these frameworks is an assumption that AMS will demand a greater burden of evidence that would be difficult to meet with the amount of testing conducted at any one stage of the system's lifecycle. Instead, we need to be trying to build a body of evidence or construct an assurance argument that draws on lines of evidence traditionally not included in those individual stages (i.e., DT or OT; Porter et al., 2020; Sparrow et al., 2018). That evidence will need to be accumulated over time (Ahner & Parson, 2016; Laverghetta et al., 2018; Lede, 2019; Lennon & Davis, 2018) and shared and reused among stakeholders (Ahner & Parson, 2016; Defense Science Board, 2016; Lede, 2019; Lennon & Davis, 2018).

Run Time Assurance

A completely separate strategy largely abandons the traditional structure of pre-fielding performance and safety testing. The argument is that because these state-spaces are so large, it will not be possible to meaningfully cover them. Instead of trying to anticipate and test these scenarios, the focus of assurance should be at runtime—on having systems that can recognize for themselves through self-monitoring when they are under cyberattack or in situations outside of their training bounds. Functionally, these are safety middleware on steroids, such as having an entire second gapped system evaluating the ethics of the first system's choices (Arnold & Scheutz, 2018). The recommendation is that the focus of TEV&V should not be on the system itself, but rather perhaps on the efficacy of the runtime monitors (Ahner & Parson, 2016; Arnold & Scheutz, 2018; Lede, 2019; Lennon & Davis, 2018). This is not necessarily an overall easier challenge, but changes the focus of factors, measures, and test structures (Tate & Sparrow, 2019).

Recommendation #4: Provide Risk Assurance for Safety and Security

The complexity associated with AMS and advances in adversarial methodology, paired with the dire consequences of failure, indicate that risk assurance is critical to ensure trust in the safe and secure operation of AMS. Recommendations for T&E must be able to broadly and consistently address system brittleness, system control, and vulnerabilities to adversarial exploitation.

Safety

Since system complexity, brittleness, and lack of continuous human-in-the-loop control can result in failures in AMS with dire consequences, recommendations for safety must address these concerns in a consistent and robust manner. Considerations include run-time assurance monitoring and real-time, continuous testing to identify unacceptable behavior (Eaton et al., 2017; Haugh et al., 2018). In particular, detection of unacceptable behavior should trigger the system to revert to a default or more deterministic mode that is less likely to violate behavioral boundaries (Haugh et al., 2018). This concept is similar to including a human in the loop to monitor and correct the behavior of the system, but replaces the human with another system. This recommendation leads to new testing challenges involving continuous monitoring, detection of behavioral violations, and the safe implementation of a behavior switching mechanism (Haugh et al., 2018).

Security

Continual and rapid development of the technology underlying AMS, paired with equally rapid developments in exploitation of that technology, indicates that research should continually explore the cyber implications of autonomous systems (Ilachinski, 2017), and policies should be adopted that facilitate cyber T&E throughout the lifecycle of the program (Air Force Scientific Advisory Board, 2017). This should include the use of red teaming, testing across the lifecycle of the program, and evaluation of the system's response to input generated through techniques like adversarial machine learning. While some models underlying AMS decision engines lack the capacity to resist adversarial perturbation (Goodfellow et al., 2015), many methods exist to generally make the decision engine more robust against adversarial perturbation (Qiu et al., 2019). Further, experimentation can be used to quantify the vulnerability of AMS to adversarial methods (Streilein et al., 2019), and adversarial testing methods can be used to quickly identify vulnerabilities in the decision engine (Haugh et al., 2018). Red teaming can be used to augment conventional DOE methods and evaluate systems under development (Haugh et al., 2018; Streilein et al., 2019). In particular, red teaming can be used in testing to identify cases that break the AMS (Haugh et al., 2018), and to test the ability of systems to counter adversarial attacks (Streilein et al., 2019).

Recommendation #5: Adoption of Policies, Standards, and Metrics

Given the documented discussions regarding the lack of policy, standards, and metrics, it seems natural to recommend the development of such items. Indeed, overcoming many of the challenges associated with T&E of AMS, such as ad-hoc and inconsistently applied procedures, will require the adoption of appropriate policies, standards, and metrics that can facilitate a common T&E framework. Experts broadly call for the creation of a common framework built upon relevant standards and utilizing metrics that appropriately quantify relevant factors. As is apparent in this review, the tasks embodied in these recommendations require careful thought and innovation.

Creation of a Common T&E Framework

Many experts call for the creation of a common, operationally meaningful, and understandable framework for T&E of autonomous systems (Defense Science Board, 2016; Ilachinski, 2017; Macias, 2008; Ring, 2009). However, given the complexity involved in testing autonomous systems, the creation of a common framework is easier discussed than implemented.

Some recommend using previously established frameworks as a guide (Durst, 2019), and some, such as Harikumar and Chan (2019), attempt to provide a guide. A wide variety of specific proposals exist, often focusing on specific cases (Eaton et al., 2017; Giampapa, 2013; Hess & Valerdi, 2010; Lenzi et al., 2010; Luna et al., 2013; McLean et al., 2016; Micskei et al., 2012; Porter et al., 2020; Scheidt, 2017; Visnevski & Castillo-Effen, 2010), but underlying each proposal is the principal need for a common T&E framework.

Develop Standards

Development of a common T&E framework requires creation and adoption of relevant standards. Recommendations acknowledge the need for standards, guidelines, and practices for continuous V&V for autonomous systems (Defense Science Board, 2016), accrediting autonomous systems (Ilachinski, 2017), data handling (Visnevski, 2008), system of systems (SoS) architecture (Luna et al., 2013), and even standard labeling (Ring, 2009), among other needs, because standards aid in effective decision making (Roske et al., 2012). In particular, the Defense Science Board (2016) recommends that (the now defunct) USD(AT&L) require a consistent and comprehensive M&S strategy throughout the lifecycle of the system. Further, standardized architecture can help testers and developers envision structural choices (Ring, 2009), and standardization of SoS architecture will enable replacement of constituent systems with minimal disruption to system interfaces (Luna et al., 2013). Standardized definitions are also recommended (Ferreira et al., 2013; Ring, 2009) as they facilitate common understanding (Ring, 2009) and cross-program comparison (Deonandan et al., 2010).

Develop Metrics

Experts broadly recommend the development of measures of effectiveness and performance to evaluate success and enable effective decision-making (Ilachinski, 2017; Ring, 2009; Roske et al., 2012; Streilein et al., 2019), noting that metrics can address many challenges (Harikumar & Chan, 2019; Scheidt, 2017). In general, metrics must address coverage adequacy (Ahner & Parson, 2016) and embody standardized language (Ferreira et al., 2013). Quantification of risks can help prioritize elements in the testing process (Deonandan et al., 2010), and the use of methods that consider the trade-offs involved in the allocation of testing activity can increase the agility and decrease the duration of testing (Air Force Scientific Advisory Board, 2017). For example, sequential analysis methods, such as using currently known information to inform when to stop testing (Hess & Valerdi, 2010) can reduce the required time to test without sacrificing the quality of the test.[8] In many cases, metrics can be based on tasks that must be completed in order to establish a measure of effectiveness (Visnevski & Castillo-Effen, 2010). Given the nature of autonomous systems and the challenges they pose, metrics must also be able to provide insight into topics that are often difficult to quantify, such as trust and the characterization of learning (Ahner & Parson, 2016; Defense Science Board, 2012; Harikumar & Chan, 2019; Porter et al., 2020; Tate, Grier, Martin, Moses, & Sparrow, 2016; Wojton et al., 2020).

[8] However, there is no free lunch, and going from a fixed sample size to a random sample size typically results in some loss of information or ability to control error rates (Haman, 2020).

Conclusions

The advent of AI capabilities in military systems will put serious strain on the standard processes, methods, and procedures DoD employs to evaluate acquisition programs. The complexity of these systems opens up vast state-spaces that testers need to explore. This test point expansion will likely require the use of some combination of two general approaches, both of which require methodological research. First, we need to increase the evidentiary value each test point provides. Part of this can come from sequential, adaptive design of experiment to maximize the information test points provide, and part of it can be improving our ability to make inferences between and beyond our test points. Both of these would enable us to achieve the same level of confidence with smaller overall tests. In parallel, we need to increase the number of test points we can collect for a given amount of resources. Numerous authors recommend leveraging cheaper, lower-fidelity methods to cover this space while simultaneously developing new techniques that improve our ability to validate these models or simulations. All of which raises the problem that agile, learn-and-plan-as-you-go testing does not fit well into the current acquisition paradigm, a reality that has generated many recommendations that this process needs to be reformed. Finally, adopting common policies, technical standards, or even system architectures across mission domains could help programs that are struggling to figure it out on their own without adequate analytic methods and AI talent in the workforce.

The most consistent recommendation theme across all authors was identifying the need for research. Significant gaps remain in our ability to evaluate AMS, and we will need to bridge them. Some problems can likely be addressed through basic academic, or partially applied science and technology outreach. Others may simply require hard-won trial-and-error lessons as we attempt to field these systems. It will be incumbent upon our T&E communities to address what issues we can before the problems they produce permit inadequate systems to slip through evaluation and fail in the field.

Our own organization, IDA,[9] will attempt to contribute to this gap-filling effort. To begin, we will be prioritizing methods advancement research where we can balance the importance and tractability of the problems. While there are many hard-to-solve challenges that will require significant lead time to address, and investment in these areas should start immediately, this investment may be better suited to other organizations and activities. IDA's immediate work will focus on creating better DOE methods for sequential experimentation, a framework for evaluating the performance of human-machine teams, and statistical methods for combining evaluations of independently tested modules.

In the spirit of desegregating the T&E communities and coordinating effort, we also recommend that these research efforts be shared through a centralized activity. There are

[9] NOTE: The following paragraphs are intended to be replaced with DOT&E priorities and relabeled as DOT&E. We are providing draft input so they have something, but ultimately this section will probably be rewritten by them.

numerous bubbles of researchers working on different aspects of these problems, but these groups often do not have strong connections with each other. A more thorough effort to find and connect these research efforts could have great value for knowledge sharing. The JAIC, the Autonomy Community of Interest, and the Defense and Aerospace Test Association Workshop (DATAWorks) could be useful places to begin.

Finally, there are numerous policy decisions regarding T&E that need to be made in the near future, and if there is any hope for adopting authors' recommendation to integrate test activities early and often in system lifecycles, these policies should probably be coordinated between the different stakeholders. For example, DoD Directive 3000.09 on Autonomy in Weapon Systems assigns OT test adequacy standards to DOT&E, and V&V standards to (the now defunct) AT&L. If DoD moves away from the waterfall approach to system development and testing, having different and possibly conflicting standards would make integrated testing particularly difficult. We recommend that DOT&E, DDT&E, and the JAIC T&E branch coordinate their policies on AI TEV&V.

Thinkers in the DoD have been writing about our autonomous horizons for over a decade now (Endsley, 2015; Office of the US Air Force Chief Scientist, 2011; Zacharias, 2019a), but these horizons are fast approaching. As the DoD accelerates the adoption of AI, the T&E community must also accelerate the development of assurance methods for AI-enabled systems. This document provides an overview of the thinking to date on the challenges these systems impose, as well as possible solutions. We hope our work will help our community be prepared when the DoD arrives at those AI horizons.

References

Achler, T. (2013). Neural networks that perform recognition using generative error may help fill the "Neuro-Symbolic Gap". *Biologically Inspired Cognitive Architectures, 3*, 6-12. doi:10.1016/j.bica.2012.10.001

Ahner, D. K., & Parson, C. R. (2016). *Workshop report: Test and evaluation of autonomous systems*. STAT Center of Excellence.

Ahner, D. K., Parson, C. R., Thompson, J. L., & Rowell, W. F. (2018). Overcoming the challenges in test and evaluation of autonomous robotic systems. *The ITEA Journal of Test and Evaluation, 39*, 86-94.

Air Force Scientific Advisory Board. (2017). *Adapting Air Force test and evaluation to emerging system needs*. Retrieved from United States Air Force Scientific Advisory Board

Alaa, A. M., & Schaar, M. v. d. (2019). *Demystifying black-box models with symbolic metamodels*. Paper presented at the 33rd Conference on Neural Information Processing Systems (NeurIPS 2019), Vancouver, Canada.

Anandkumar, A. (2020). How to create generalizable AI? *GTC 2020*.

Arnold, T., & Scheutz, M. (2018). The "big red button" is too late: an alternative model for the ethical evaluation of AI systems. *Ethics and Information Technology, 20*(1), 59-69. doi:10.1007/s10676-018-9447-7

Atherton, K. D. (2019). Can the Army perfect an AI strategy for a fast and deadly future? *AUSA*. Retrieved from https://www.c4isrnet.com/artificial-intelligence/2019/10/15/can-the-army-perfect-an-ai-strategy-for-a-fast-and-deadly-future/

Avery, M. R., & Simpson, J. R. (2019). *"How much testing is enough?" 25 years later*. Retrieved from individual release request

Baker, B., Kanitscheider, I., Markov, T., Wu, Y., Powell, G., McGrew, B., & Mordatch, I. (2019). Emergent tool use from multi-agent autocurricula. *ArXiv e-prints*.

Box, G. E. P. (1976). Science and statistics. *Journal of the American Statistical Association, 71*(356), 791-799.

Box, G. E. P. (1979). Robustness in the strategy of scientific model building. In R. L. Launer & G. N. Wilkinson (Eds.), *Robustness in statistics*. New York: Academic Press.

Brabbs, J., Lohrer, S., Kwashnak, P., Bounker, P., & Brudnak, M. (2019). *M&S as the key enabler for autonomy development, acquisition and testing*. Paper presented at the 2019 NDIA Ground Vehicle Systems Engineering and Technology Symposium.

Caseley, P. (2018). *Human-machine trust: Risk-based assurance and licensing of autonomous systems.* Paper presented at the SCI-313 Specialist Meeting Report.

Chella, A., Cossentino, M., Gaglio, S., & Seidita, V. (2012). A general theoretical framework for designing cognitive architectures: Hybrid and meta-level architectures for BICA. *Biologically Inspired Cognitive Architectures, 2*, 100-108. doi:10.1016/j.bica.2012.07.002

Cook, A. (2019). *Taming killer robots*. The JAG School Papers: Air University Press.

Defense Innovation Board. (2018). *DIB guide: Detecting agile BS*. Retrieved from https://media.defense.gov/2018/Oct/09/2002049591/-1/-1/0/DIB_DETECTING_AGILE_BS_2018.10.05.PDF.

Defense Innovation Board. (2019). *AI principles: Recommendations on the ethical use of artificial intelligence by the Department of Defense*.

Defense Science Board. (2012). *The Role of Autonomy in DoD Systems*. Washington, DC.

Defense Science Board. (2016). *Summer Study on Autonomy*. Washington, D.C.

Deonandan, I., Valerdi, R., Lane, J. A., & Macias, F. (2010). *Cost and risk considerations for test and evaluation of unmanned and autonomous systems of systems*. Paper presented at the 5th International Conference on System of Systems Engineering, Loughborough, UK.

Autonomy in Weapons Systems, 3000.09 C.F.R. (2012).

Durst, P. J. (2019). *The Reference Autonomous Mobility Model: A framework for predicting autonomous unmanned ground vehicle performance.* (Degree of Doctorate of Philosophy in Computational Engineering), Mississippi State, ProQuest LLC.

Durst, P. J., & Gray, W. (2014). *Levels of autonomy and autonomous system performance assessment for intelligent unmanned systems*. (ERDC/GSL SR-14-1). US Army Engineer Research and Development Center (ERDC).

Eaton, C. M., Chong, E. K. P., & Maciejewski, A. A. (2017). Services-Based Testing of Autonomy (SBTA). *The ITEA Journal of Test and Evaluation, 38*, 40-48.

Endsley, M. R. (2015). *Autonomous horizons: System autonomy in the Air Force – A path to the future*. Maxwell AFB, AL: Air University Press.

Endsley, M. R. (2019). *BOHSI Panel: Explainable AI, System Transparency, and Human Machine Teaming*. Paper presented at the 63rd International Annual of the Human Factors & Ergonomics Society, Seattle, Washington.

Ferreira, S., Faezipour, M., & Corley, H. W. (2013). *Defining and addressing the risk of undesirable emergent properties*. Paper presented at the 2013 IEEE International Systems Conference (SysCon), Orlando, FL.

Funahashi, K. I. (1989). On the approximate realization of continuous mappings by neural networks. *Neural Networks, 2*(3), 183-192.

Ghassemi, M. (2020). *The false hope of explainable machine learning in healthcare*. Paper presented at the Joint Statistical Meeting 2020, COVID Virtual. https://amstat-jsm.conferencecontent.net/session/219272

Giampapa, J. A. (2013). Test and evaluation of autonomous multi-robot systems. Pittsburgh, PA: Software Engineering Institute.

Gil, Y., & Selman, B. (2019). A 20-year community roadmap for artificial intelligence research in the US. *Computing Community Consortium (CCC) and Association for the Advancement of Artificial Intelligence (AAAI)*.

Goerger, S. R. (2004). *Validating human behavioral models for combat simulations using techniques for the evaluation of human performance*. Paper presented at the SCSC '03.

Goodfellow, I. J., Shlens, J., & Szegedy, C. (2015). Explaining and harnessing adversarial examples. *ArXiv e-prints*.

Gray, C. (2015, May 19, 2015). When the only winning move is not to play. Retrieved from https://conradthegray.com/blog/when-the-only-winning-move-is-not-to-play/

Greer, K. (2013). A metric for modelling and measuring complex behavioural systems. *IOSR Journal of Engineering (IOSRJEN), 3*(11).

Gunning, D. (2017). Explainable Artificial Intelligence program update: DARPA.

Gunning, D. (2019). *Explainable Artificial Intelligence*. Paper presented at the Board of Human-Systems Integration: Explainable AI Frontiers: Human-Systems Integration Challenges and Opportunities, Washington, D.C.

Gunning, D., & Aha, D. W. (2019). DARPA's Explainable Artificial Intelligence (XAI) Program. *AI Magazine, 40*(2), 44-58. doi:https://doi.org/10.1609/aimag.v40i2.2850

Haman, J. T. (2020, September 08, 2020).

Harikumar, J., & Chan, P. (2019). *Developing knowledge and understanding for autonomous systems for analysis and assessment events and campaigns*. (ARL-TR-8649).

Haugh, B. A., Sparrow, D. A., & Tate, D. M. (2018). *The status of test, evaluation, verification, and validation (TEV&V) of autonomous systems*. Retrieved from Alexandria, VA:

Heinrich, K., Zschech, P., Skouti, T., Griebenow, J., & Riechert, S. (2019). *Demystifying the black box: A classification scheme for interpretation and visualization of deep intelligent systems.* Paper presented at the AMCIS.

Hernández-Orallo, J. (2016). Evaluation in artificial intelligence: from task-oriented to ability-oriented measurement. *Artificial Intelligence Review, 48*(3), 397-447. doi:10.1007/s10462-016-9505-7

Hess, J. T., & Valerdi, R. (2010). *Test and evaluation of a SoS using a prescriptive and adaptive testing framework.* Paper presented at the 2010 5th International Conference on System of Systems Engineering, Loughborough, UK.

Hill, A., & Thompson, G. (2016). Five giant leaps for robotkind: Expanding the possible in autonomous weapons. *War on the Rocks*. Retrieved from War on the Rocks website: https://warontherocks.com/2016/12/five-giant-leaps-for-robotkind-expanding-the-possible-in-autonomous-weapons/

Hoff, K. A., & Bashir, M. (2014). Trust in automation: Integrating empirical evidence on factors that influence trust. *Human Factors: The Journal of the Human Factors and Ergonomics Society*, 0018720814547570.

Horowitz, M., & Scharre, P. (2015). *Meaningful human control in weapon systems: A primer*: Center for a New American Security.

Ilachinski, A. (2017). *AI, robots, and swarms issues: Questions and recommended studies*. Retrieved from Arlington, VA:

Johnson, G. (1984). Eurisko, the computer with a mind of its own. *the APF Reporter*. Retrieved from https://aliciapatterson.org/stories/eurisko-computer-mind-its-own

Kapinski, J., Deshmukh, J., Jin, X., Ito, H., & Butts, K. (2016). Simulation-based approaches for the verification of embedded control systems. *IEEE Control Systems Magazine*, 45-64.

Kirkpatrick, J., Pascanu, R., Rabinowitz, N., Veness, J., Desjardins, G., Rusu, A., . . . Hadsell, R. (2016). Overcoming catastrophic forgetting in neural networks. *Proceedings of the National Academy of Sciences, 114*. doi:10.1073/pnas.1611835114

Krichmar, J. L. (2012). Design principles for biologically inspired cognitive robotics. *Biologically Inspired Cognitive Architectures, 1*, 73-81. doi:10.1016/j.bica.2012.04.003

Kurup, U., & Lebiere, C. (2012). What can cognitive architectures do for robotics? *Biologically Inspired Cognitive Architectures, 2*, 88-99. doi:10.1016/j.bica.2012.07.004

Kwashnak, P. (2019). *Autonomous Systems Test Capability (ASTC) overview*. Paper presented at the Workshop on Test and Evaluation of Artificial Intelligence Enabled Systems, Aberdeen Proving Grounds, Maryland.

Laird, J. E. (2012). *The SOAR cognitive architecture*. Cambridge, Massachusetts: The MIT Press.

Laverghetta, T. J., Leathrum, J. F., & Gonda, N. (2018). *Integrating virtual and augmented reality based testing into the development of autonomous vehicles*. Paper presented at the MODSIM World 2018, Norfolk, VA.

Lede, J. (2019). *Autonomy overview*. Paper presented at the US-Japan Service to Service Dialogue.

Lee, J. D., & See, K. A. (2004). Trust in automation: Designing for appropriate reliance. *HUMAN FACTORS, 46*(1), 50-80. doi:10.1518/hfes.46.1.50_30392

Lenat, D. B. (1983). EURISKO: a program that learns new heuristics and domain concepts: the nature of heuristics III: program design and results. *Artificial intelligence, 21*(1-2), 61-98.

Lennon, C., & Davis, E. (2018). Autonomy Community of Interest: Test & Evaluation, Verification & Validation Working Group *US-UK TEM*.

Lenzi, N., Bachrach, B., & Manikonda, V. (2010). *DCF® a JAUS and TENA compliant agent-based framework for UAS performance evaluation.* Paper presented at the PerMIS '10 Proceedings of the 10th Performance Metrics for Intelligent Systems Workshop, Baltimore, MD.

Lopez, C. T. (2020). DOD adopts 5 principles of artificial intelligence ethics. *DOD News*. Retrieved from https://www.defense.gov/Explore/News/Article/Article/2094085/dod-adopts-5-principles-of-artificial-intelligence-ethics/

Luna, S., Lopes, A., Tao, H. Y. S., Zapata, F., & Pineda, R. (2013). Integration, verification, validation, test, and evaluation (IVVT&E) framework for system of systems (SoS). *Procedia Computer Science, 20*, 298-305. doi:10.1016/j.procs.2013.09.276

Macias, F. (2008). The Test and Evaluation of Unmanned and Autonomous Systems. *ITEA Journal, 29*, 388-395.

McLean, A. L., Bertram, J. R., Hoke, J. A., Rediger, S. S., & Skarphol, J. C. (2016). *LVC-enabled testbed for autonomous system testing*. Rockwell Collins. Retrieved from https://insights.rockwellcollins.com/2016/10/31/lvc-enabled-testbed-for-autonomous-system-testing/

Menzies, T., & Pecheur, C. (2005). *Verification and validation of artificial intelligence*. Paper presented at the Advances in Computers, Amsterdam.

Micskei, Z., Szatmári, Z., Oláh, J., & Majzik, I. (2012). *A concept for testing robustness and safety of the context-aware behaviour of autonomous systems*, Berlin, Heidelberg.

Miller, H. (2019). *Report on test infrastructure for emerging technology*.

Miller, M. J., McGuire, K. M., & Feigh, K. M. (2017). Decision support system requirements definition for human extravehicular activity based on cognitive work analysis. *Journal of Cognitive Engineering and Decision Making, 11*(2), 136-165. doi:https://doi.org/10.1177/1555343416672112

Mitchell, B. (2019). Google's departure from Project Maven was a 'little bit of a canary in a coal mine'. *fedscoop*. Retrieved from https://www.fedscoop.com/google-project-maven-canary-coal-mine/

Mueller, S. T., Hoffman, R. R., Clancey, W., Emrey, A., & Klein, G. (2019). *Explanation in human-AI systems: A literature meta-review synopsis of key ideas and publications and bibliography for explainable AI*.

Mullins, G. E., Stankiewicz, P. G., Hawthorne, R. C., Appler, J. D., Biggins, M. H., Chiou, K., . . . Watkins, A. S. (2017). Delivering test and evaluation tools for autonomus unmanned vehicles to the fleet. *JOHNS HOPKINS APL TECHNICAL DIGEST, 33*(4), 279-288.

Narla, A., Kuprel, B., Sarin, K., Novoa, R., & Ko, J. (2018). Automated classification of skin lesions: From pixels to practice. *Journal of Investigative Dermatology, 138*(10), 2108-2110. doi:https://doi.org/10.1016/j.jid.2018.06.175

National Transportation Safety Board. (2019). *Collision between vehicle controlled by developmental automated driving system and pedestrian, Tempe, Arizona, March 18, 2018*. (NTSB/HAR-19/03 or PB2019-101402).

Nisbett, R. E., & Wilson, T. D. (1977). Telling more than we can know: Verbal reports on mental processes. *Psychological Review, 84*(3), 231-259.

O'Connor, T., & Wong, H. Y. (2012). Emergent properties. *Stanford Encyclopedia of Philosophy.* Spring 2012. Retrieved from https://plato.stanford.edu/archives/spr2012/entries/properties-emergent/

Office of the US Air Force Chief Scientist. (2011). *Technology horizons: A vision for Air Force science and technology 2010-2030*. Maxwell AFB, AL: Air University Press.

OpenAI. (2018). OpenAI Five. Retrieved from https://openai.com/blog/openai-five/

Oxford English Dictionary. (Ed.) (2020) The Oxford English Dictionary.

Pinelis, Y. K. (2020). *AI test and evaluation framework*. Paper presented at the Joint Statistical Meeting 2020, COVID Virtual.

Polk, T., & Seifert, C. M. (2002). *Cognitive modeling* Cambridge, Massachusetts: The MIT Press.

Porter, D. J. (2019). *Demystifying the black box - A test strategy for autonomy*. Paper presented at the DATAWorks 2019, Springfield, VA.

Porter, D. J. (2020a). *Briefing to the URSA Legal, Moral, & Ethical Working Group: Assuring ethical behavior with AI-enhanced capabilities*. Retrieved from individual release request

Porter, D. J. (2020b). *A HellerVVA problem: The Catch-22 for simulated testing of fully autonomous systems*. Paper presented at the DATAWorks 2020, COVID Virtual.

Porter, D. J., McAnally, M., Bieber, C., & Wojton, H. M. (2020). *Trustworthy autonomy: A roadmap to assurance - Part 1: System effectiveness*. Retrieved from https://www.ida.org/research-and-publications/publications/all/t/tr/trustworthy-autonomy-a-roadmap-to-assurance-part-1-system-effectiveness

Porter, D. J., Pinelis, Y. K., Bieber, C. M., Wojton, H. M., McAnally, M. O., & Freeman, L. J. (2018). *Operational testing of systems with autonomy*. Retrieved from individual release request

Porter, D. J., & Wojton, H. M. (2020). *Briefing to the Air Force Scientific Advisory Board: T&E contributions to avoiding unintended behaviors in autonomous systems*. Retrieved from individual release request

Qiu, S., Liu, Q., Zhou, S., & Wu, C. (2019). Review of artificial intelligence adversarial attack and defense technologies. *Applied Sciences, 9*(5), 909.

R&E, A. (2015). *Technology Investment Strategy*.

Ring, J. (2009). Evolving an autonomous test and evaluation enterprise.

Roske, V. P., Kohlberg, I., & Wagner, R. (2012). *Autonomous systems: Challenges to test and evaluation.* Paper presented at the National Defense Industrial Association Test & Evaluation Conference.

Samsonovich, A. V. (2012). On a roadmap for the BICA Challenge. *Biologically Inspired Cognitive Architectures, 1*, 100-107. doi:10.1016/j.bica.2012.05.002

Sandamirskaya, Y., & Burtsev, M. (2015). NARLE: Neurocognitive architecture for the autonomous task recognition, learning, and execution. *Biologically Inspired Cognitive Architectures, 13*, 91-104. doi:10.1016/j.bica.2015.06.007

Santoni de Sio, F., & van den Hoven, J. (2018). Meaningful human control over autonomous systems: A philosophical account. *Frontiers in Robotics and AI, 5*. doi:10.3389/frobt.2018.00015

Scheidt, D. (2017). NAVAIR Autonomy TEVV Study: Weather Gage Technologies, LLC.

Schultz, A. C., Grefenstette, J. J., & Jong, K. A. D. (1993). Test and evaluation by genetic algorithms. *IEEE Expert, 8*, 9-14.

Silver, D., Huang, A., Maddison, C. J., Guez, A., Sifre, L., van den Driessche, G., . . . Hassabis, D. (2016). Mastering the game of Go with deep neural networks and tree search. *Nature, 529*(7587), 484-489. doi:10.1038/nature16961

Simen, P., & Polk, T. (2010). A symbolic/subsymbolic interface protocol for cognitive modeling. *Log J IGPL, 18*(5), 705-761. doi:10.1093/jigpal/jzp046

Simpson, J. R. (2020). *Sequential testing and simulation validation for autonomous systems*. Paper presented at the DATAWorks 2020, COVID Virtual.

Sparrow, D. A. (2020).

Sparrow, D. A., Tate, D. M., Biddle, J. C., Kaminski, N. J., & Madhavan, P. (2018). *Assessing the quality of decision-making by autonomous systems*. Retrieved from

Stracuzzi, D. J., Chen, M., Darling, M., Jones, T., Peterson, M., & Vollmer, C. (2020). *The role of uncertainty quantification in machine learning*. Paper presented at the DATAWorks 2020, COVID Virtual.

Streilein, W., Thornton, J., Malyska, N., Bernays, J., Mersereau, B., Roeser, C., . . . Zipkin, J. (2019). Future NMI analysis study: Counter-AI: Massachusetts Institute of Technology.

Subbu, R., Visnevski, N. A., & Djang, P. (2009). *Evolutionary framework for test of autonomous systems*. Paper presented at the PerMIS'09, Gaithersburg, MD, USA.

Tate, D. (2020, July 07, 2020).

Tate, D., Grier, R. A., Martin, C. A., Moses, F. L., & Sparrow, D. (2016). *A framework for evidence-based licensure of adaptive autonomous systems*. Retrieved from

Tate, D., & Sparrow, D. (2018). *Acquisition challenges of autonomous systems*. Paper presented at the Proceedings of the 15th Annual Acquisition Research Symposium, Monterey, CA.

Tate, D., & Sparrow, D. (2019). Lesson 2: How do autonomous capabilities affect T&E? *Automous Systems Test & Evaluation - CLE 002*: Defense Acquisition University.

Templeton, B. (2019). Tesla's "Shadow" testing offers a useful advantage on the biggest problem in robocars. *Forbes*. Retrieved from https://www.forbes.com/sites/bradtempleton/2019/04/29/teslas-shadow-testing-offers-a-useful-advantage-on-the-biggest-problem-in-robocars/#2349d1943c06

Thuloweit, K. (2019). Emerging Technologies CTF conducts first autonomous flight test. Retrieved from https://www.af.mil/News/Article-Display/Article/1778358/emerging-technologies-ctf-conducts-first-autonomous-flight-test/

Trent, S. (2019). *The Joint Artificial Intelligence Center: Transforming the DoD with human-centered technology*. Paper presented at the 63rd International Annual of the Human Factors & Ergonomics Society, Seattle, Washington.

US Department of Defense. (2019). *Summary of the 2018 Department of Defense Artificial Intelligence Strategy: Harnessing AI to Advance Our Security and Prosperity*.

Visnevski, N. A. (2008). *Embedded Instrumentation Systems Architecture*. Paper presented at the IEEE International Instrumentation and Measurement Technology Conference, Victoria, Vancouver Island, Canada.

Visnevski, N. A., & Castillo-Effen, M. (2010). *Evolutionary computing for mission-based test and evaluation of unmanned autonomous systems*. Paper presented at the IEEE Aerospace Conference.

Voulodimos, A., Doulamis, N., Doulamis, A., & Protopapadakis, E. (2018). Deep learning for computer vision: A brief review. *Comput Intell Neurosci, 2018*, 7068349. doi:10.1155/2018/7068349

Wegener, J., & Bühler, O. (2004). *Evaluation of different fitness functions for the evolutionary testing of an autonomous parking system.* Paper presented at the Genetic and Evolutionary Computation Conference.

Wojton, H. M., Porter, D. J., & Lane, S. T. (2020). Initial validation of the Trust of Automated Systems Test (TOAST). *Social Psychology*.

Woods, D., & Dekker, S. (2000). Anticipating the effects of technological change: A new era of dynamics for human factors. *Theoretical Issues in Ergonomics Science, 1*(3), 272-282. doi:10.1080/14639220110037452

Zacharias, G. L. (2019a). *Autonomous horizons: The way forward*. Maxwell AFB, Alabama: Air University Press.

Zacharias, G. L. (2019b). *Emerging technologies: Test and evaluation implications*. Paper presented at the DATAWorks 2019, Springfield, Virginia.

Zhou, Z., & Sun, L. (2019). Metamorphic testing of driverless cars. *Communications of the ACM, 63*(3), 61-67.

Appendix A: Summaries of Individual Studies

US Air Force Scientific Advisory Board (2017). Adapting Air Force test and evaluation to emerging system needs, SAB-TR-17-03.

This study focuses on five operationally-oriented challenge problems, all of which are associated with AS discovery, improvisation, and "initiative", a particular problem area for T&E:

- ❖ Challenges
  - "Testing and Evaluation methods are needed to quantify trust in autonomous and self-learning systems
    - ⇒ "Bounding system behavior is insufficient to quantify trust
    - ⇒ "Traditional test methods will overwhelm T&E capacity
    - ⇒ "No current methods exist for understanding decision logic
    - ⇒ "Range of system autonomy … will likely require application-dependent trust metrics
  - "Implementation of Deep-learning systems
    - ⇒ "Test methods do not yet exist for deep-learning systems
    - ⇒ "Limits AF ability to field systems"

- ❖ Recommendations
  - [R]equire an aggressive M&S-based approach
    - ⇒ To fully integrate early program, formal development and operational test planning to guide all virtual, ground, and [Open Air] testing.
  - [A]dhere to rigorous principles and best practices that enable software design-for-test.
    - ⇒ Strategies include developing and enforcing standards that promote software modularity (e.g., between critical and non-critical components) and the use of open architectures; formal and stochastic software V&V including correct-by-design software; and agile/DevOps software development strategies, where appropriate.
  - [A]dopt a policy to conduct cyber T&E throughout a program's lifecycle….
    - ⇒ The lifelong testing would be greatly aided by virtual testing using the National Cyber Range (NCR) or a similar capability, but that requires an increase in the investment of the effort to purchase the avionics and then develop models that accurately reflect the possible cyber vulnerabilities. The NCR is more focused at the enterprise level, and virtualization would enable full-scale cyber testing at the avionics/embedded software levels and for systems-of-systems.

- [Apply a] comprehensive data analytics approach to T&E analysis and planning needs.
    - ⇒ Specifically, next-generation DOE techniques should be developed and implemented for efficient test design in the case of autonomous and networked-autonomous systems.
    - ⇒ [E]xtending current automatic software test tools into the realm of autonomous test tools that decide when they have accomplished enough testing in a certain area and turn their attention to other high-need regions offer great potential to increase the accuracy and agility, while decreasing cycle times, of our current software practices.
- [C]ontinue to partner with academia, industry, and operators
    - ⇒ To investigate and develop the T&E methods needed across the gamut of autonomous systems (i.e., autonomous, networked autonomous, learning systems).

Ahner, D., & Parson, C. (2016). Workshop Report: Test and Evaluation of Autonomous Systems. *STAT Center of Excellence*, Wright-Patterson AFB, OH.

The primary challenge in testing autonomous systems is the broad scale and complexity of the systems, missions, and conditions. This is best addressed by breaking down the requirement, system, and/or mission into smaller pieces, which can then be readily translated into rigorously quantifiable statistical designs.

❖ Challenges

- Requirements and Measures
    - ⇒ "When quantifying the performance of autonomous systems, many difficulties arise. T&E must address the inputs, internal processes and states, and outputs for all autonomous functions employed by the system (e.g., perception, reasoning, learning, decisions, and behavior) as well as overall system performance to inform acquisition decisions and operational risk."
    - ⇒ "Each of these elements must be understood in order to quantify performance of the decision engine, to inform acquisition decisions and operational risk, and ultimately to build trust from the warfighter that the system will perform as intended. This situation presents challenges to defining requirements in ways that are clear and testable, including in a well-defined and comprehensive operational mission environment. It also presents challenges to developing metrics to adequately test and evaluate these systems."
    - ◊ How to quantify success of decision making
    - ◊ How to measure perception, reasoning, and learning
    - ◊ Metrics for: Trust, Intent, System learning, Perception, Reasoning, Distributed perception, Distributed decision-making
- Personnel, Test Infrastructure, Knowledge/Methods
    - ◊ Developing Skills and Recruiting
        - ⇒ "A need to identify and develop the necessary skill sets within existing personnel, and recruit qualified candidates to become experts in test and evaluation of autonomous systems."
        - ⇒ "This challenge has two key elements: training the existing and future workforce and developing new T&E methodologies relevant to autonomous systems."
    - ◊ "Identification and development of correct range requirements and instrumentation needed to perform valid test and evaluation of autonomous systems."
    - ◊ Knowledge/Methods
        - ⇒ Sequential progressive testing will likely be critical, and developing methods which allow for real-time test planning and conduct are needed.
        - ⇒ The need for range world models. These models will likely need to be both

physical and simulated, and should account for the unique aspects that autonomous systems will bring to the test community.

- ⇒ Issues associated with the areas of repeatability and the desire to quantifiably bound the performance envelope while testing to the edges of those boundaries, when dealing with an on-line learning AS.

- Design for Test[ability]
  - ◊ The test API should be
    - ⇒ Domain agnostic, with sufficient extensibility to accommodate domain specific attributes
    - ⇒ Able to precisely stimulate the system to simulate specific input conditions
    - ⇒ Able to extract sufficient data to provide insight or introspection of internal states during specified trigger events or time stamps
    - ⇒ Able to perform the stimulation and data extraction during M&S events and during live test events
  - ◊ The system's world models and experiences will need to be
    - ⇒ discoverable prior to testing in order to baseline the system's level of intelligence.
- Test Adequacy & Integration
  - ◊ How is the acquisition process affected, and how do we adapt it?
  - ◊ What new scientific approaches for test design need to be developed to ensure adequacy
  - ◊ What safety and security considerations need to be adequately tested?
  - ◊ What safety considerations need to be adequately addressed in design of the test environment?
  - ◊ How will agile or elastic test planning and test conduct be performed to accommodate dynamic test conditions?
- Testing Continuum
  - ◊ Resource conflicts (ranges, personnel, funding)
  - ◊ Need flexible concept of 'required' performance
  - ◊ Next test depends on result of previous
  - ◊ Not aligned with current requirements and processes
  - ◊ Need to embed testers from day 1, with test support at development
  - ◊ Test to evaluate utility, not rigid requirements
- Safety / Cyber Security for Autonomous Systems
  - ◊ Inability to test all cases – infinite factor space further complicated by a potentially infinite decision space
  - ◊ System may be changing continuously as knowledge is gained and decisions are made
  - ◊ How to tell "good" perception/reasoning from bad
  - ◊ Awkward time scales: Cyber may be too fast, but long endurance tests will likely be too slow
  - ◊ Potentially exploitable algorithms (decision processes) by adversaries

  - ◊ Manage, mitigate, and define risks
  - ◊ Real-time prediction
  - ◊ “Negative learning” may not surface during any feasible test or acquisition cycles
- Testing of Human System Teaming
  - ◊ Characterize and measure performance of human-machine partnership
  - ◊ Measure shared situational awareness
  - ◊ Testing human-machine trade space
  - ◊ Testing heterogeneous autonomous systems
  - ◊ Measuring difference between human intent and machine actions
  - ◊ Creating a normalized model of the human
- Post Acceptance Testing
  - ◊ Recurring, periodic assessment of compliance with existing, newly introduced, or learned capabilities, rules, or constraints
  - ◊ Assess value, robustness of learning (guard against negative learning or brittleness)
  - ◊ Will require some level of self-monitoring
    - ⇒ Prevent negative responses from developing
    - ⇒ Last acceptable certified level reversion capability
    - ⇒ What is trigger that requires more testing?
  - ◊ Assess autonomous system adaptation to an aging physical system
  - ◊ Feedback & assessment of updates

❖ Recommendations

  - ◊ Improved statistical engineering methods
    - ⇒ Improved statistical engineering methods are needed to support both developmental and operational testing of autonomous systems to address systems interacting with a dynamic environment in a non-deterministic manner. Improvement of these methods is a component of a larger need; these adaptive autonomous systems will require more stringent adherence to systems engineering principles throughout development.
  - ◊ Processes and methods need to be developed to address Inputs – Process – Outputs of autonomous systems and human-machine element interaction and roles within the process.
  - ◊ A test and evaluation continuum paradigm
    - ⇒ A test and evaluation continuum paradigm must be developed and adopted that requires testing start early and a more sequential progressive approach is taken that includes development and implementation of a comprehensive M&S strategy across the life cycle. There is no “test phase” with a beginning or end; it extends throughout the life cycle of the system.
  - ◊ Measures must be developed to address state space [coverage] adequacy, trust, and human-machine interaction.

◊ Design of experiment methods must be developed for defining test cases and expected results that overcome the difficulty of enumerating all conditions and non-deterministic responses that autonomy will generate in response to complex environments.

◊ Models and live virtual constructive (LVC) test beds are needed that support robust testing while minimizing risk and cost.

◊ Development of techniques that capture learning growth, possibly similar to reliability growth models is needed.

Arnold, T., & Scheutz, M. (2018). “The ‘big red button’ is too late: an alternative model for the ethical evaluation of AI systems.” *Ethics and Information Technology, 20,* 59-69. https://doi.org/10.1007/s10676-018-9447-7

> “As a way to address both ominous and ordinary threats of artificial intelligence (AI), researchers have started proposing ways to stop an AI system before it has a chance to escape outside control and cause harm. A so-called “big red button” would enable human operators to interrupt or divert a system while preventing the system from learning that such an intervention is a threat. … In this paper, [the authors] describe the demands that recent big red button proposals have not addressed, and [they] offer a preliminary model of an approach that could better meet them.”
>
> “The twofold aim of this paper is (1) to show how the big red button fails to address larger practical questions about AI safety, and (2) to sketch a technical approach based on an “ethical core” in the system that addresses several important aspects of such broader safety considerations.”

❖ Challenges

- 1. Testing an AI system for ethical concerns and shutting down the system as a result of ethical concerns
    - ◊ Use of a ‘big red button’ could come too late.
        - ⇒ “Though an emergency shutdown button for AI makes some intuitive sense, the general premise for such a device is to intervene at the point when a system has, in effect, already “gone rogue.””
    - ◊ The system may be able to prevent itself from being shutdown
        - ⇒ “Aside from needing not to be late, this type of intervention is supposed to fend off a worry with reward-based learning systems: what happens when such a system finds a way to manipulate its reward function or figures out a way to maximize rewards by preventing itself from being shut off?”
    - ◊ Anticipating and projecting actions in view of ethical principles
        - ⇒ “If the moment of crisis for a system that has gone rogue gives too short notice for its violation, some type of anticipatory and pre-emptive test is needed.”
    - ◊ Keeping Actions and principles free from manipulation
        - ⇒ “the fidelity of a test hinges on the system not being adjusted (by itself or others) for the specific purpose of meeting test requirements. Thus, the means of examination will have to be opaque to the system itself.” Also, “a system’s internal means of deceit may be the problem.”
    - ◊ Continuing diagnostics and “opaque” self-examination
        - ⇒ “Given uncertainty around the scope and impact of a system’s actions, especially ones far from obviously fraught with danger, a test must be ongoing and vigilant, and must be kept out of the system’s purview.”
- 2. Testing a system in a simulated environment

- ◊ Synchronizing the simulation used for testing with the real world
- ◊ The system may need to act in the real world while it is being tested in simulation
- ◊ Preventing the system from learning that it is being tested

❖ Recommendations

- An ethical core (EC) built into the AI system’s architecture that is opaque to the AI and forms a buffer between the AI’s actions and the real world
  - ⇒ “The overall idea is to utilize a system architecture where an “ethical core/testing layer” is introduced below a virtual machine layer where different operating systems can run in virtual machines, which, in turn, run the AI system’s control architecture (see Fig. 1 in the source document). The ethical layer runs on top of a minimal operating system and is comprised of all functionality necessary to perform the ethical testing. By situating this layer outside the VM on which the AI system is running, the AI system has no way of knowing whether there is such a layer, nor can it access any of that layer’s states. The idea here is to remove any possibility for the AI system to directly manipulate the testing routine, regardless of how advanced its learning algorithms and its potentially malicious intent might be. In addition, the proposed architectural configuration utilizing virtual machines does not require a redesign of the AI system being analyzed; rather, the AI system can run unaltered in the VM on top of the ethical core.”
- ◊ Continual, ongoing self-evaluation and testing
  - ⇒ “A better approach would be to make ongoing self-evaluation and testing an integral part of a system’s operation, diagnose how the system is in error and to prevent chaos and risk before they start.” “A continual testing approach will be far more effective, responsive, and vigilant toward a system’s learning and action in the world than an emergency button which one might not get to push in time.”
- ◊ “This EC would be kept opaque to the system itself.”
- ◊ A simulation environment
  - ⇒ “… realistic enough for the agent to be unable to tell whether its sensory information is coming from the real world or the simulated environment”
- ◊ An ethical scenario generator
  - ⇒ “… that can generate a large number of ethically challenged scenarios together with solutions using the real-world scenarios the agent is familiar with”
- ◊ A minimally operational shut-down system
  - ⇒ “…that can read sensory information and operate the actuators in a way that will bring the physical system into an orderly state where the system can be turned off”
- ◊ Connections the system’s sensors and actuators
  - ⇒ “that can be intercepted and re-routed in the VM layer and interfaces to the VM layer that allow for saving, cloning, starting and stopping virtual machines”

◊ Ethical Scenario-generation mechanism paired with a simulation environment for testing

⇒ "We argue for an ethical core (EC) that consists of a scenario-generation mechanism and a simulation environment that are used to test a system's decisions in simulated worlds, rather than the real world. This EC would be kept opaque to the system itself."

◊ Testing a cloned version of the system in a parallel environment

⇒ "the EC instructs the VM layer to checkpoint the AI/architecture layer and build a VM image of the system. The image is then saved and cloned, and the cloned image SV is then started in a new VM with sensors and actuators of connect to the simulated environment. System S continues to operate normally in the meantime without having been able to notice that it was cloned."

Baker, B., Kanitscheider, I., Markov, T., Wu, Y., Powell, G., McGrew, B., & Mordatch, I. (2019). Emergent tool use from multi-agent curricula. *OpenAI white paper*.

This study focused on novel methods of reinforcement learning to develop game playing agents (hide-and-seek), but in doing so created methods for quantitatively evaluating agent performance of potential use to T&E. To overcome the problem of metric selection, they recommend decomposing missions into domain-specific sub-skills and developing quantitative metrics for those skills, as well as measuring speed of relearning in related tasks or environments to help quantify generalizability across potential operational environments.

- ❖ Challenges
  - Task Complexity and Evaluation Scalability
    - ⇒ “However, as environments increase in scale and multi-agent autocurricula become more open-ended, evaluating progress by qualitative observation will become intractable.”
  - Emergent Behavior
    - ⇒ “Co-adaptation between agents and environments can also give rise to emergent complexity”
  - Metric Selection
    - ⇒ “…the objective being optimized does not directly incentivize the learned behavior, making evaluation of those behaviors nontrivial. Tracking reward is an insufficient evaluation metric in multi-agent settings, as it can be ambiguous in indicating whether agents are improving evenly or have stagnated. Metrics like ELO or Trueskill can more reliably measure whether performance is improving relative to previous policy versions or other policies in a population; however, these metrics still do not give insight into whether improved performance stems from new adaptations or improving previously learned skills.”

- ❖ Recommendations
  - Select operationally relevant skills (low- and high-level) to quantitatively test performance against
    - ⇒ “We therefore propose a suite of targeted intelligence tests to measure capabilities in our environment that we believe our agents may eventually learn, e.g., object permanence, navigation, and construction.”
    - ⇒ “proposed framework for evaluating agents in open-ended environments as well as a suite of targeted intelligence tests for our domain”
    - ⇒ “…we then propose a suite of domain-specific intelligence tests to quantitatively measure and compare agent capabilities.”
  - Evaluate transfer to related tasks or environments

⇒ “We propose to use transfer to a suite of domain-specific tasks in order to assess agent capabilities. To this end, we have created 5 benchmark intelligence tests that include both supervised and reinforcement learning tasks. The tests use the same action space, observation space, and types of objects as in the hide-and-seek environment.”

Caseley, P. (2018, December). Human-machine trust: Risk-based assurance and licensing of autonomous systems. *SCI-313 Specialist Meeting Report*.

> This meeting report contains sparse summaries of many presenters and ideas. From the lack of detail and diversity of sources, it is difficult to identify a unified theme. However, the studies focus on the problem of human-machine interactions, and recommend a licensing-based assurance case for specific autonomous capabilities. Whether this is meant to apply at the process-level (e.g., a licensing test for self-driving vehicles vs. test for face recognizers vs. etc.) or at the system-level (e.g., Expedient Leader-Follower has its license and can be given/sold to Britain), or both is unclear.

❖ Challenges

- Complexity
- Simulation Fidelity
  - ⇒ "The assurance from simulating autonomous behaviors and the requisite fidelity of simulations needs further research."
- Human supervision does not fully mitigate autonomy errors
  - ⇒ "In other words, humans are poor supervisory controllers."

❖ Recommendations

- License autonomous capabilities
  - ⇒ "Licensing is a means of introducing a new capability; it could be specific to the autonomy-only element."
  - ⇒ "Licensing would need an assurance process that provides evidence that the autonomy element is sufficient."
  - ⇒ "Licensing would need an authority to issue the license and "police" the assurance processes that gives confidence in the autonomy."
  - ⇒ "The assurance process does not need to be common between licensing agencies but would be specific to the autonomy-human."
  - ⇒ "The license would define the capability of the necessary assurance of the autonomy-human team elements, the license would be the limitation of the capability (not the whole system) – a bit like the weapon and the platform."
  - ⇒ "Licenses may be time limited to allow for changes in the environment and then re-issue of license indicating that the authority is satisfied that the human-autonomy can cope with the evolved environment."
- Reuse existing methods where possible
  - ⇒ "Similarity between Unmanned and Manned should not be ignored when considering assurance challenges."
  - ⇒ "The structure and forms of assurance argument and evidence may differ but the principles of assurance cases still hold."

- ⇒ “Challenges relating to processes, methods, tools, etc. for assuring dependable systems are largely solved. AS are not necessarily more complex than traditional dependable systems. We already have state-space explosion, unpredictable environments, emergent behavior, HMIs, etc.”

- Have independent specialists
  - ⇒ “SCI-313 reviewed whether autonomy or systems containing autonomy have a route to military capability through use of licensing – the result was licensing looks feasible and has some benefits of establishing trust but would need the support of a specialist board/authority to assess and determine and endorse the licensing bounds of the HMT.”

Office of the Assistant Secretary of Defense For Research & Engineering. (2015). Autonomy Community of Interest (COI) Test and Evaluation, Verification and Validation (TEVV) Working Group Technology Investment Strategy 2015-2018.

The number of gaps in TEVV for autonomous systems stem from a number of challenges with which testing has not had to traditionally contend, including the state-space explosion, systems that exhibit emergent behaviors, and the complexities of human-machine teaming. Autonomous TEVV will require new methods and tools that account for the complexities of engineering machine learning systems with non-deterministic behaviors and allow for new information learned through test and evaluation to shape requirements analysis. The paper presents this concept as a flattening of the "Classic V" systems engineering development cycle model.

❖ Challenges

- State space explosion
    - ◊ Increasing levels of autonomy make full testing of autonomous systems increasingly infeasible; testing cannot account for all possible decision spaces and states
        - ⇒ “Autonomous systems are characteristically adaptive, intelligent, and/or may incorporate learning. For this reason, the algorithmic decision space is either non-deterministic, i.e., the output cannot be predicted due to multiple possible outcomes for each input, or is intractably complex. Because of its size, this space cannot be exhaustively searched, examined, or tested; it grows exponentially as all known conditions, factors, and interactions expand. Therefore there are currently no established metrics to determine various aspects of success or comparison to a baseline state enumerated.”
- Unpredictable environments
    - ◊ The unpredictability of environments exacerbates the state space problem
        - ⇒ “The power of autonomous agents is the ability to perform in unknown, untested environmental conditions. Examples of environmental “stimuli” include actors capable of making their own decisions in response to autonomous system actions; producing a cognitive feedback loop that explodes the state space. Additionally, autonomous decisions are not necessarily memoryless and the state space is not just the intractably complex in the current situation, but also in the multiplicity of situations over time. Currently fielded systems have very limited robustness to dynamic / changing environmental conditions. Adaptive autonomous algorithms have the potential to overcome current automated system brittleness in future dynamic, complex, and/or contested environments. However, this performance increase comes with the price of assuring correct behavior in a countless number of environmental conditions. This exacerbates the state-space explosion problem.”
- Emergent behaviors

  - ◊ “Interactions between systems and system factors may induce unintended consequences.”
    - ⇒ “Interactions between systems and system factors may induce unintended consequences. With complex, adaptive systems, how can all interactions between systems are captured sufficiently to understand all intended and unintended consequences? How can autonomous design approaches identify or constrain potentially harmful emergent behavior both at design time and at run time? What limitations are there with the current Design of Experiments approach to test vector generation when considering adaptive decision-making in both discrete decision logic and continuous variables in an unpredictable environment? Since emergent behavior can be produced by interactions between small, seemingly insignificant factors how can we provide test environments or test oracles that are of sufficient fidelity to examine and capture emergent behavior (in M&S, T&E, and continuous operations or run time testing)?”
- Human-machine teaming/communication
  - ◊ It is difficult to verify trust in autonomous systems through M&S and T&E as is possible in other older systems
    - ⇒ “Handoff, communication, and interplay between operator and autonomy become a critical component to the trust and effectiveness of an autonomous system. Current certification processes eliminate the need for “trust” through exhaustive Modeling and Simulation (M&S) and T&E to exercise all possible operational vignettes. When this is not possible at design time, how can trust in the system be ensured, what factors need to be addressed, and how can transparency and human-machine system requirements for the autonomy be defined?
- Gaps in ATEVV
  - ◊ Lack of verifiable autonomous system requirements (around CONOPs, MOEs, performance measures, metrics in general)
    - ⇒ “Currently, there is a lack of common, clear, and consistent requirements for systems that include autonomous requirements, especially with respect to environmental assumptions, Concept of Operation (CONOPS), interoperability, and communication. There is also a lack of clearly defined Measures of Effectiveness (MOEs), performance measures, and other metrics.”
  - ◊ Lack of modeling, design, interface standards
    - ⇒ “Currently, no standardized modeling frameworks exist for autonomous systems that span the whole system lifecycle (R&D through T&E). Therefore, a gap exists in traceability between capabilities implemented in conventional systems as well as adaptive, nonlinear, stochastic, and/or learning systems and the requirements they are designed to meet. This results in a need to integrate models that are both heterogeneous and composable in nature and existing at different levels of abstraction, including requirements, architecture models, physics-based models, cognitive models, test range/environment models, etc.”

    - ◊ Lack of AT&E capabilities (test beds, skillsets, ranges – rework and redesign because of technology are also problematic)
        - ⇒ "As stated before, there is a current gap in T&E ranges, test beds, and skillsets for handling dynamic learning/adaptive systems. The complexity of these systems results in an inability to test under all known conditions, difficulties in objectively measuring risk, and an ever-increasing cost of rework/redesign due to errors found in late developmental and operational testing. Furthermore, the lack of formalized requirements and system models makes test-based instrumentation of model abstractions increasingly difficult. This limits design-for-test capabilities, including tests to evaluate human-autonomy interactions."
    - ◊ Lack of human operator reliance to compensate for brittleness
        - ⇒ "Currently, the burden of decision making under uncertainty is placed solely on human operators. Certification, acceptance, and risk mitigation often assume the human operator can compensate for the brittleness currently found in manned, remotely piloted, or tele-operated systems. However, as systems move from relatively predictable automated behaviors to more unpredictable and complex autonomous behaviors, and as autonomous systems operate in denied environments in which they interact with human intermittently, it will become increasingly difficult for human operators to understand and respond appropriately to decisions made by the system."
    - ◊ Lack of runtime V&V during deployed autonomy operations
        - ⇒ As stated earlier, current automated systems rely on human oversight to guarantee safe and correct operation, with the human operator acting as the ultimate monitor, kill switch, and recovery mechanism for brittle automation. However, as systems incorporate higher levels of autonomy, it will no longer be feasible or safe to rely solely on human operators for system monitoring and recovery.
    - ◊ Lack of evidence re-use for V&V

- ❖ Recommendations
    - Adopt software engineering "V" development model
        - ◊ Use V model to incorporate V&V throughout the design process
            - ⇒ "AFRL researchers have proposed that the classic "V" used to describe the software development process be modified to incorporate verification activities throughout the development cycle (see Figures 1 & 2). Figure 2 shows a conceptual Autonomy Community of Interest TEVV Process Model that integrates development and V&V, with V&V activities occur during and between each major development activity. With this process model, we endeavor to "flatten" the systems engineering "V" through the incremental and compositional assurances (or arguments) of safety, security, performance, and risk."
    - Aim to meet ATEVV goals laid out by ASD (R&E) Autonomy COI

◊ TEVV Goal 1: Methods & Tools Assisting in Requirements Development and Analysis

⇒ “This goal focuses on increasing the fidelity and correctness of autonomous system requirements by developing methods and tools to enable the generation of requirements that are, where possible, mathematically expressible, analyzable, and automatically traceable to different levels (or abstractions) of autonomous system design.

⇒ “Formalized requirements enable automatic test generation and traceability to low-level designs, but note that TEVV representatives must be involved early in the requirements development process. Specific requirements and requirement templates must be constructed that articulate how autonomous vehicles need to perform in unknown and untested environmental conditions that can induce unintended consequences.”

◊ TEVV Goal 2: Evidence-Based Design and Implementation

⇒ “Methods and tools need to be developed at every level of design from architecture definition to modeling abstractions to software generation / hardware fabrication, enabling the compositional verification of the progressive design process, thereby increasing test and evaluation efficiency.”

◊ TEVV Goal 3: Cumulative Evidence through RDT&E, DT, & OT

⇒ “Methods must be developed to record, aggregate, leverage, and reuse M&S and T&E results throughout the system’s engineering lifecycle; from requirements to model-based designs, to live virtual construction experimentation, to open-range testing.”

◊ TEVV Goal 4: Run Time Behavior Prediction and Recovery

⇒ “For highly complex autonomous systems, an alternate method leveraging a run-time architecture must be developed that can provably constrain the system to a set of allowable, predictable, and recoverable behaviors, shifting the analysis/test burden to a simpler, more deterministic run-time assurance mechanism.”

◊ TEVV Goal 5: Assurance Arguments for Autonomous Systems

⇒ “Not only do multiple new TEVV methods need to be employed to enable the fielding of autonomous systems, a new research area needs to be investigated in formally articulating and verifying that the assurance argument itself is valid.”

⇒ “Additionally, standard autonomy argument templates must be developed, enabling the reuse of explicit arguments of risk, performance, and safety, closely tied to autonomy requirements and TEVV practices which, if performed, provide an acceptable collection of evidence for an autonomous system.”

Deonandan, I., Lane, R., & Macias, J. (2010). Cost and risk considerations for test and evaluation of unmanned and autonomous system of systems. *2010 5th International Conference on System of Systems Engineering*, Loughborough, 2010, 1-6.

This article focuses on how the challenges of testing AI/AS will impact budgeting challenges, and propose a risk-based, parametric cost estimation procedure to help with choosing, scoping, and planning test events.

- ❖ Challenges
  - System of System Level Risks
    - ◊ System Complexity
      - ⇒ "Many factors can increase the integration complexity of the SoS including the number of systems to be integrated, number of interfaces involved and technology maturity of the SoS. Many UASoS have never even existed in the past making it very difficult to predict any emergent properties. A UASoS requires the ability for manned and unmanned systems to co-operate with each other to fulfill its purpose. In addition, the number of requirements of the SoS is a key driver of risk, as well as changes in requirements throughout SoS development and operation."
    - ◊ Lack of CONOPs & Metrics
      - ⇒ "Many times it is unclear what the SoS needs to do in order to fulfill its mission and without the appropriate metrics to evaluate the performance of the UASoS, it is difficult to determine whether the mission is successful or not."
    - ◊ Assuring Interoperability
      - ⇒ "individual systems within a SoS may have varying levels of maturity and may enter the SoS at different stages of the SoS lifecycle. Ensuring that these systems can still work together and merging newer more advanced technologies with more traditional technologies can present a significant challenge to development and validation of the SoS."
  - Testing and Network Risks
    - ◊ State-space
      - ⇒ "Unmanageable combinatorial problems can result when a large number of tests need to be performed on a large number of systems, and especially in the DoD, there is need to prioritize tests to ensure the systems meet schedule requirements."
    - ◊ Prioritizing test events/points
      - ⇒ "The type of test and amount of each type of test to be performed will also be a driver of costs. For example, field tests require considerable resources, labor, and scheduling, and is significantly more costly than a simulated test which can be done in a virtual environment."
    - ◊ Coordinating T&E efforts

- ⇒ “Multisite coordination for testing also becomes an issue especially when multiple stakeholders are involved and individual systems are located in many different places.”
- ◊ Emergent Properties
  - ⇒ “When systems are integrated, it is difficult to predict how the test process needs to adapt to account for emergent properties…”
  - ⇒ “T&E contributes significantly to the cost of the system especially given the risks and uncertainties associated with UASoS.
- Budget planning challenges
  - ◊ AS will need more testing than traditional systems
    - ⇒ “previous studies on systems engineering cost models have shown that developers are so convinced that T&E is such a small proportion of the total life cycle cost, … However, further analysis of T&E in the SoS environment … are leading experts to re-evaluate these ideas.”
  - ◊ AS haven’t been built before
    - ⇒ “The budget, both in terms of cost and effort, is currently determined based on similar projects that have been conducted in the past, coupled with extrapolations to account for the new system under test. However, UASoS do not have a significant history, but are in such high demand that there is the need to understand how much effort is required for testing.”

❖ Recommendations

- Move to a continuum of testing
  - ⇒ “There needs to be a move away from traditional boundary lines placed between developmental testing (DT) and operational testing (OT). Currently, developmental testing entails identifying the technical capabilities and limitations, assessing the safety, identifying and describing technical risk, assessing maturity, and testing under controlled conditions. Operational testing is a fielding activity, measuring the capability of the system to perform in the field. On a SoS level, especially with a UASoS, both these spectrums of testing are required simultaneously and even existing programs currently do joint integration testing emphasizing the need for the involvement of both operational and developmental testing throughout the life cycle of the UASoS.”
- Simulate where possible
  - ⇒ “While it is impossible to eliminate all risks through computer simulations; the more failure scenarios that can be predicted and tested in a simulated environment, the less costly it will be during the fielding process, especially in the case of communication failures and loss of equipment.”
- Plan tests by estimating costs and risk through modeling
  - ⇒ By using a risk based testing approach, we identified the risks that need to be mitigated, and what priorities need to be made in the testing process based on

these risks. We will combine these results into a cost model, which we will use to estimate the amount of test effort required for a given level of confidence.

- ◊ Create a common language for AI cost/risk drivers to enable cross-program comparison
  - ⇒ "One of the important elements is ensuring that the definitions of the drivers are consistent so they can be rated from similar perspectives. We presented the prioritization of both technical and organizational cost drivers."

Department of Defense (2019). DoD Digital Modernization Strategy: DoD Information Resource Management Strategic Plan FY19-23.

This document covers the DoD Digital Strategy, which focuses on a range of issues that go beyond artificial intelligence into areas like cloud computing, digital modernization, innovation, cybersecurity and infrastructure modernization. AI test and evaluation is not specifically referenced in the document, which focuses on AI (specifically through the JAIC) as one aspect of the broader strategy. Objective 1 of the strategy's first goal ("Innovate for Competitive Advantage") is to "Establish the Joint Artificial Intelligence Center (JAIC) to Accelerate Adoption and Integration of AI-Enabled Capabilities to Achieve Mission Impact at Scale." While AI T&E is not explicitly referenced, a challenge can be inferred from the Strategy Elements for Objective 1 (establishing the JAIC).

- ❖ Challenges
  - Incorporate oversight, ethics, and safety
    - ◊ "Strategy Element #4: Lead DoD in AI Planning, Policy, Oversight, Ethics, and Safety"
      - ⇒ "[The JAIC] will operationalize AI capabilities by overcoming policy, technical, and financial roadblocks that prevent enterprise-level deployment, and by leveraging the capabilities established through DISN modernization (as described in Goal 1, Objective 8)."
  - Assure/certify AI Algorithms, Data, and Models
    - ◊ "Strategy Element #6: Assure /Certify the AI Algorithms, Data, and Models Developed for JAIC Implementations"
      - ⇒ "[The JAIC] will operationalize AI capabilities by overcoming policy, technical, and financial roadblocks that prevent enterprise-level deployment, and by leveraging the capabilities established through DISN modernization (as described in Goal 1, Objective 8)."

Defense Science Board (2012). *The Role of Autonomy in DoD Systems*. 2012 DSB Autonomy Study: Office of the Under Secretary of Defense for Acquisition, Technology and Logistics, Washington, DC, 2012.

“The Under Secretary of Defense for Acquisition, Technology and Logistics (USD (AT&L)) should create developmental and operational test and evaluation (T&E) techniques that focus on the unique challenges of autonomy (to include developing operational training techniques that explicitly build trust in autonomous systems).”

- ❖ Challenges
  - ◊ Nondeterministic performance and decision making by the T&E community
  - ◊ Highly dynamic, unstructured operating environments and computational tractability
  - ◊ Input/state space size and tests do not provide the needed levels of assurance
    - ⇒ “[T]here is a need for acceptance of nondeterministic performance and decision making by the test and evaluation community. Unmanned systems will operate in highly dynamic, unstructured environments, for which there are not computationally tractable approaches to comprehensively validate performance. Formal methods for finite-state systems based on abstraction and model-based checking do not extend to such systems, probabilistic or statistical tests do not provide the needed levels of assurance, and the set of possible inputs is far too large. Both run-time and quantum [?] verification and validation (V&V) approaches may prove to be viable alternatives. Run-time approaches insert a monitor/checker and simpler verifiable backup controller in the loop to monitor system state during run time and check against acceptable limits, and then switch to a simpler backup controller (verifiable by traditional finite-state methods) if the state exceeds limits.”

- ❖ Recommendations
  - ◊ “Creating techniques for coping with the difficulty of defining test cases and expected results for systems that operate in complex environments and do not generate deterministic responses.”
  - ◊ “Measuring trust that an autonomous system will interact with its human supervisor as intended.”
  - ◊ “Developing approaches that make the basis of autonomous system decisions more apparent to its users.”
  - ◊ “Advancing technologies for creating and characterizing realistic operational test environments for autonomous systems.”
  - ◊ “Leveraging the benefits of robust simulation to create meaningful test environments.”

- ◊ “[E]stablish a research program to create the technologies needed for developmental and operational test and evaluation (T&E) that address the unique challenges of autonomy. Among the topics that this research should address are:
  - ⇒ “Techniques for defining test cases and expected results that overcome the difficulty of enumerating all conditions and non-deterministic responses that autonomy will generate in response to complex environments,
  - ⇒ “Methods and metrics for confirming that an autonomous system will perform or interact with its human supervisor as intended and for measuring the user’s trust in the system,
  - ⇒ “Interfaces that make the basis of autonomous system decisions more apparent to its users,
  - ⇒ “Test environments that include direct and indirect users at all echelons, as appropriate for an intended capability and
  - ⇒ “Robust simulation to create meaningful test environments.”
- ◊ “[S]tructure autonomous systems acquisition programs to separate the autonomy software from the vehicle platform.”

Defense Science Board (2016). Office of the Under Secretary of Defense for Acquisition, Technology and Logistics, Report of the Defense Science Board Summer Study on Autonomy, Washington DC, 2016.

Recommendations fall into two broad categories: operations-oriented and development process-oriented. Process oriented recommendations include challenges in mitigating increasing cyber vulnerabilities, and creating new T&E and M&S paradigms applicable to ASs. Issues and recommendations in AS T&E include:

- ❖ Challenges
  - Learning Systems
    - ⇒ Dealing with learning systems is more difficult. It necessitates continuous testing, both via M&S and live testing, throughout the life cycle. Records need to be kept of the environment under which the behavior was captured, as well as, if possible, the reasoning the system went through to arrive at that behavior.
  - Emergent Behavior
    - ⇒ Dealing with "emergent behavior" is also difficult. Conduct research on how to handle it.

- ❖ Recommendations
  - [E]stablish a new T&E paradigm for testing software that learns and adapts
    - ⇒ "The Director of Operational Test and Evaluation (DOT&E), in conjunction with the Office of Developmental Test and Evaluation (DT&E), should establish a new T&E paradigm for testing software that learns and adapts. Considerations include:
    - ⇒ "Opportunities to adopt or adapt commercial best practices in T&E of learning systems
    - ⇒ "Experimentation and development of new methods and means for testing software that learns, such that the "correct answer" changes with context, experience, and new data"
  - [E]stablish a new paradigm for T&E of autonomous systems that encompasses the entire system lifecycle.
    - ⇒ "The DoD test and evaluation community should establish a new paradigm for T&E of autonomous systems that encompasses the entire system lifecycle. Considerations include:
    - ⇒ "Make extensive use of live and synthetic environments for evaluating and qualifying transition from development to fielded systems
    - ⇒ "Establish standards and guidelines for continuous verification and validation (V&V) for autonomous systems
  - USD(AT&L) should require the acquisition community to establish and implement a consistent and comprehensive M&S strategy throughout the lifecycle of the system

- Expand the use of M&S to do automated testing of “thousands of test cases”, then do follow up “real world” testing to ensure that the M&S results match the real world cases.
- Do iterative modeling and testing early (build, model, test, modify); keep the data and models throughout development. Embrace DevSecOps software practices of the commercial world.

Durst. P. J. (2019). The Reference Autonomous Mobility Model: A framework for predicting autonomous unmanned ground vehicle performance. *Dissertation completed at Mississippi State University Department of Computational Engineering* (2019).

This thesis presents a framework and model for assessing autonomous vehicles through the lens of unmanned ground vehicles (UGVs), building a framework called the Reference Autonomous Mobility Model (RAMM), in a search for a methodology to validate and verify, “simulations of complex, intelligent, and autonomous systems”. The RAMM attempts to be a “mission-level” mobility model to autonomous ground vehicles. The model is physics-based, building on previously-developed physics-based models for ground systems. By integrating all systems into a single physics-based model for autonomous systems, the validation and verification (V&V) of autonomous systems as a whole may be possible. This comes after a “layered” approach of testing models of individual sensors, the environment, and comparisons of sensors to the “real world” and manage differences of the simulation to the environment.

Further, the author tests GPS algorithms, lane detection algorithms, and a convolutional neural network (CNN) using the author’s proposed RAMM and route, while incorporating LIDAR, GPS, and IMU sensor data into a “fuzzy” path planning simulation.

❖ Challenges
- Practicality of validation & verification of autonomous
  - ◊ “Lack of a theoretical framework for V&V of simulations for predicting the behaviors of autonomous robotic systems.”
    - ⇒ A variety of frameworks and methods exist, and some attempt to grapple with quantifying the uncertainty in a simulation that could come along with autonomous systems
    - ⇒ Open-ended physics models are more difficult to control and produce outputs which are too open-ended
    - ⇒ Trust in autonomous systems is difficult to measure with “no firmly established formal methodology for obtaining trust in the outputs of these simulations or even for defining what ‘trust’ means.”
- V&V of full systems is difficult in separate tests
  - ◊ Each model depends on different rules and assumptions, and multiple models usually work together
  - ◊ Potential for already-validated models to produce invalid results
  - ◊ Current simulation models are difficult to validate, for example USARSim

- ❖ Recommendations
  - I – Employ “layered” V&V
    - ⇒ Test components in separate models, analyze outputs in their unique context in extreme conditions; V&V the environment model as well
    - ⇒ Check each model’s ability to make judgements about its environment
    - ⇒ The author presents an example using the Virtual Autonomous Navigation Environment (VANE) tool with a case of a sign detection algorithm detecting a stop sign in a VANE image and a similar setup with a real-world image. The author compares the rates of detection in both images each to validate the algorithm
  - II – Combine V&V’d models into a single, physics based model
    - ⇒ Must produce “closed-loop” simulation results that may take thousands of iterations
    - ⇒ Evaluate all components on a “gross level” once combined
    - ⇒ Visual framework on pages 75 and 76
  - III –Use previously established frameworks as a guide
    - ⇒ Suggests use of Sargent’s definitions of V&V, Balci’s comparisons of graphical outputs of a model, and extreme condition testing
    - ⇒ The author’s RAMM builds on the “conceptual design” of the NATO Reference Mobility Model (NRMM), a tool built for ground vehicle acquisitions but is not built for autonomous vehicle testing, and the VANE tool, used for testing sensor algorithms
  - IV – Suggests looking to fields of human behavioral modeling for models to develop autonomous systems models
    - ⇒ Fields such as economics and tools such as fuzzy neural nets and finite state machines can aide in crafting decision spaces

Eaton, C., Chong, E., & Maciejewski, A. (2017). Services-based testing of autonomy (SBTA). *The ITEA Journal of Test and Evaluation*, 38, 40-48.

The test and evaluation (T&E) of autonomous systems, that adequately supports the verification and validation (V&V) process, is a significant challenge facing the test community. The ability to quickly and reliably test autonomy is necessary to provide a consistent T&E, V&V (TEVV) capability. A safe, efficient, and cost effective test capability, regardless of autonomy or sensor capability, is required. Autonomy and sensor capabilities, referred to as services, can be integrated easily into small Unmanned Aircraft Systems (sUAS) of differing capabilities and complexities. An integrated open-source architecture, for both software and hardware, implemented on multiple sUAS of varying capabilities can provide a robust test capability for emerging autonomous behaviors. The inclusion of a run time assurance (RTA) common safety watchdog and a Live-Virtual-Constructive (LVC) capability provides a consistent, robust, and safe test capability/environment. The use of an open software and hardware architecture ensures cross-platform viability. These features will allow test teams to focus on the newly incorporated autonomy and sensor services, not on other ancillary capabilities and systems on the test vehicle. Testing of the services in this manner will enable a common TEVV approach, regardless of final platform integration while decreasing risk and accelerating the availability of autonomy services. Services-Based Testing of Autonomy (SBTA) provides a cost-effective and focused capability to test autonomous services, whether software, hardware, or both.

❖ Challenges
- Integrating autonomy into existing platforms:
  - ⇒ The cost to integrate autonomy into existing platforms can be expensive and time-consuming. Providing a means to perform early testing of autonomy services is, therefore, critical.
- Potential for spoofing sensors
  - ⇒ The intersection between the cyber and autonomy worlds raises yet another source of TEVV difficulty: can the perception sensors feeding the autonomy engine be spoofed (GPS, signature imagery, etc.) and can the autonomy engine understand that the “perceived world view” has been hacked?
- Testing complex cyber-physical systems
  - ⇒ Complex cyber-physical systems, such as autonomous UASs, are difficult to test with traditional methods currently used for standard UASs and manned aircraft.

❖ Recommendations
- Service-Based Testing of Autonomy
  - ⇒ Providing a robust, reliable, and reusable approach to testing autonomy is key; this is why the SBTA method has been developed. This approach will provide

a heterogeneous fleet of UASs that are simple and cost effective to modify and operate. These vehicles will be modified to have an open system architecture that will enable easy reconfiguration and installation of autonomy or other services that will enable testing. The approach is adaptable to different vehicles across a large operational envelope.

Ferreira, S., Faezipour, M., & Corley, H. W. (2013). Defining and addressing the risk of undesirable emergent properties. *2013 IEEE International Systems Conference (SysCon)*, Orlando, FL, 836-840.

The authors discuss the difference, in their view, between Desirable Emergent Properties (DEPs) and Undesirable Emergent Properties (UEPs) in an autonomous system, and the possibility and process of managing UEPs as systems engineering risks. DEPs are "are engineered into human-made systems by understanding and developing systems that meet an identified set of stakeholder requirements. DEPs are wanted by stakeholders," while UEPs, "encompass behaviors or conditions that can cause unwelcome, detrimental or counterproductive functions or circumstances. UEPs can impair the ability of a complex system to meet its stakeholders' desired system objectives."

- ❖ Challenges
  - Managing emergent properties/UEPs in system designs
    - ◊ Various sources and causes of UEPs
      - ⇒ "EPs can develop as a result of internal system element interfaces and the system's external interfaces with other systems throughout the system's lifecycle from its development through its disposal. EPs can arise in a system due to the environmental conditions in which the system is developed, tested, operated, and supported. The human interface with the system can also introduce EPs."
      - ⇒ "If the system is used in ways not originally intended, EPs may also result."
      - ⇒ "EPs may be expected, where the property is deliberately engineered into a system or where there is already awareness that the property will be present given the selection and arrangement of particular components. EPs may also be unexpected, where the property is unforeseen and has not been forecast to occur."
      - ⇒ "While it may be possible to predict UEPs in systems, in other cases, forecasting all UEPs may not be possible. Many UEPs may be surprises due to the lack of knowledge and pre-existing awareness of patterns that exists with interactions between specific elements of a system, other systems, humans, and environments, especially if the system is new."
    - ◊ Systems engineering does not often consider UEPs in the development lifecycle
      - ⇒ "Limited attention is normally paid to considering UEPs during the risk management process."
      - ⇒ "One would expect that the more complex a system and its interfaces with various other systems, humans, and environments, the greater the quantity of

interactions with various elements that could cause a greater potential for emergent properties."

- ❖ Recommendations
  - I - Create a taxonomy of DEPs and UEPs
    - ⇒ "The taxonomy was updated to use 'expected' and 'unexpected' as opposed to the use of the words 'planned' and 'unplanned'. This taxonomy is helpful in understanding how to characterize Eps".
    - ⇒ Separates "Emergent Properties" into "Expected" and "Unexpected", then each into "Desirable" and "Undesirable".
  - II - Observe steps in development process to analyze and plan for risks
    - ⇒ "Risk Identification", "Risk Analysis", "Risk Mitigation Planning", "Risk Mitigation Plan Implementation", and "Risk Tracking."
  - III - View UEPs as systems engineering risks
    - ⇒ "Given the potential for negative consequences, it is imperative to consider UEPs as risks. The potential for UEPs should be considered over the entire system lifecycle if they cannot be eliminated without compromising the DEPs."
    - ⇒ "The benefit of adding [UEPs as a risk in current frameworks like INCOSE] is that it can act as a trigger to systems engineers that the risks of UEPs may be associated to a system and its interactions with humans, other systems, and environments, especially where these may not have been considered as risks before."
    - ⇒ "System engineering may include trading off and balancing the expected DEPs and UEPs in a system in order to achieve an optimum solution considering particular architecture choices."
    - ⇒ "If particular UEPs are expected, a risk manager can prioritize the potential for various UEPs and develop a plan to manage them. They can be added to the system risk checklist if not already there. On the other hand, the possibility of unexpected UEPs requires vigilance on the part of the risk manager and contingent emergency planning when these EPs occur."
    - ⇒ "In this latter case of "unknown unknowns", it is still possible to identify that risk does exist because the system or modifications to the system may be novel, where it is new development and there is currently a lack of experience and a lack of knowledge of potential UEPs."
  - IV - Delineate between predicting emergent properties
    - ⇒ "…make a distinction between predicting and detecting emergent properties. Prediction, or forecasting, that an emergent property may exist, is performed before an emergent property occurs. Detection is the observation that an emergent property has occurred."
    - ⇒ "…it should become easier to predict emergent properties as a system evolves over its lifecycle. This is primarily due to the additional information available about the system as well as information about its interfaces with other systems,

humans, and the environments that is defined as development progresses and as the system is physically implemented."

⇒ "Existing literature defines factors which can be used to assess emergence. Collier and Muller discuss cohesion as an important characteristic related to emergence. Gore and Reynolds present reproducibility of behaviors, predictability, and temporal aspects in an emergence taxonomy. Vinerbi, Bondavalli, and Lollini present sources of emergence to include components, interactions, and context. Ferreira and Tejeda propose additional factors which could be used to predict emergent properties."

Giampapa, J. (2013, October 30). Test and evaluation of autonomous multi-robot systems. *Software Engineering Institute, Carnegie Mellon University*. Presentation.

This is a presentation focusing on assuring/testing and evaluation of autonomous agents that are considered entities in a cyber-physical system and members of a socio-technical system. The author claims that "cost-effective quantifiable assurance techniques for individual and coordinated robots are possible." In particular, the author discusses two complementary techniques: probabilistic model checking and reliability analysis. He states that "the preliminary results are encouraging," and "more research is required to evaluate potential for reuse and shortened assurance processes [and] to account for more coordination phenomena."

- ❖ Challenges
  - 1. Assuring behavior of autonomous multi-robot systems
    - ◊ Too difficult, time-consuming, labor-intensive
    - ◊ Idiosyncratic to the robot, mission and operating context
    - ◊ Not generalizable, reusable or cost-effective
  - 2. Evaluation of a property when the state space is large and the state and/or property are stochastic
    - ⇒ Classical Model Checking has the disadvantages of state space explosion and atomistic representation of the state space, and when both the state space and/or property are stochastic, it is difficult to enumerate all states and properties without abstraction.
  - 3. Understanding Behavioral Performance
    - ◊ Understand the atomistic behavioral performance characteristics
    - ◊ Relate these characteristics to each other
    - ◊ Characterize effects of using multiple robots
    - ◊ Predict and validate

- ❖ Recommendations
  - Probabilistic Model Checking
    - ⇒ Provides ways for expressing the state and property for stochastic systems and properties, and accommodates "rules of combination to reduce [the] state space."
  - Behavioral Reliability Analysis
    - ⇒ The technique that best predicts multi-robot coordinated performance
    - ⇒ Analyzing the data allowed [them] to hypothesize root causes for misbehaviors
    - ⇒ Performance expectations were revised once [they] understood how the metrics behave

Gil, Y. & Selman, B. (2019). *A 20-year community roadmap for artificial intelligence research in the US.* Computing Community Consortium (CCC) and Association for the Advancement of Artificial Intelligence (AAAI). Released August 6, 2019.

This material does not focus on the challenges of testing AI or autonomy, but the research, development, and cultural hurdles to achieving meaningful and useful AI. However, they make two cross-cutting recommendations that apply to the T&E community as well: invest in developing our infrastructure and workforce, as neither is sufficient yet for the challenges of AI.

- ❖ Challenges
  - Task Relevance/Fidelity
    - ⇒ "AI testbeds must strive to balance tractability and real-world relevance. Many researchers choose to study simplified tasks in closed domains, rather than open-ended real-world problems, because toy tasks are more tractable for today's methods."
  - Lack of Research and Testing Infrastructure
    - ⇒ "There are no existing examples of mission-centered AI laboratories. Outside of AI, there are many such experimental laboratories in other sciences that could serve as models. The SLAC National Accelerator Laboratory, founded in 1962 as the Stanford Linear Accelerator Center, has led to four Nobel-winning results in particle physics to date."

- ❖ Recommendations
  - Develop testbeds with care
    - ⇒ "In order to strike this balance between research tractability and real-world relevance, testbeds need to be carefully and iteratively crafted in a tight feedback loop between testbed designers and AI researchers."
  - Invest in testbeds and research centers
    - ⇒ "Each MAIL would fund AI research in the range of $100M/year for several decades, with a separate budget for operations and support of the center. This would be a reasonable period of time to see significant returns on investment and transformative research in the target areas. With this level of funding, a MAIL could support an ecosystem of roughly 50 permanent AI researchers, 50 visitors from the National AI Research Centers at any given time, 100-200 AI engineers, and 100 domain experts and staff focused on supporting AI research. MAILs should be run by people with substantial AI experience and credentials in order to bring together the research community and ensure that research quality remains a priority."
  - **"I — Create and Operate a National AI Infrastructure** to serve academia, industry, and government through four interlocking capabilities:

- ◊ "Open AI platforms and resources:
  - ⇒ "a vast interlinked distributed collection of "AI-ready" resources (such as curated high quality datasets, software, knowledge repositories, testbeds for personal assistants and robotics environments) contributed by and available to the academic research community, as well as to industry and government."
- ◊ "Sustained community-driven AI challenges:
  - ⇒ "organized sequences of challenges that build on one another, posed by AI and domain experts to drive research in key areas, building upon—and adding to—the shared resources in the Open AI Platforms and Facilities."
- ◊ "National AI Research Centers:
  - ⇒ "multi-university centers with affiliated institutions, focused on pivotal areas of long-term AI research (e.g., integrated intelligence, trust, and responsibility), with decade-long funding to support on the order of 100 faculty, 200 AI engineers, 500 students, and necessary computing infrastructure. These centers would offer rich training for students at all levels. Visiting fellows from academia, industry, and government will enable cross-cutting research and technology transition."
- ◊ "Mission-Driven AI Laboratories:
  - ⇒ "living laboratories for AI development in targeted areas of great potential for societal impact. These would be "AI-ready" facilities, designed to allow AI researchers to access unique data and expertise, such as AI-ready hospitals, AI-ready homes, or AI-ready schools. They would work closely with the National AI Research Centers to provide requirements, facilitate applied research, and transition research results. These laboratories would be crucial for R&D, dissemination, and workforce training. They would have decade-long funding to support on the order of 50 permanent AI researchers, 50 visitors from AI Research Centers, 100-200 AI engineers and technicians, and 100 domain experts and staff."

- "II — Re-conceptualize and Train an All-Encompassing AI Workforce, building upon the National AI Infrastructure listed above to:
  - ◊ "Develop AI Curricula at All Levels:
    - ⇒ "guidelines should be developed for curricula that encourage early and ongoing interest in and understanding of AI, beginning in K-12 and extending through graduate courses and professional programs."
  - ◊ "Create Recruitment and Retention Programs for Advanced AI Degrees:
    - ⇒ "including grants for talented students to obtain advanced graduate degrees, retention programs for doctoral-level researchers, and additional resources to support and enfranchise AI teaching faculty."
  - ◊ "Engage Underrepresented and Underprivileged Groups:
    - ⇒ "programs to bring the best talent into the AI research effort."
  - ◊ "Incentivize Emerging Interdisciplinary AI Areas:

- ⇒ **“**initiatives to encourage students and the research community to work in interdisciplinary AI studies—e.g., AI safety engineering, as well as analysis of the impact of AI on society—will ensure a workforce and a research ecosystem that understands the full context for AI solutions.”

◊ “Highlight AI Ethics and Policy:

- ⇒ “including the importance of the area of AI ethics and policy, and the imperative of incorporating ethics and related responsibility principles as central elements in the design and operation of AI systems.”

◊ “Address AI and the Future of Work:

- ⇒ “these challenges are at the intersection of AI with other disciplines such as economics, public policy, and education. It is important to teach students how to think through the ethical and social implications of their work.”

◊ “Train Highly Skilled AI Engineers and Technicians:

- ⇒ “support and build upon the National AI Infrastructure to grow the AI pipeline through community colleges, workforce retraining programs, certificate programs, and online degrees.”

Goerger, S. (2004). Validating human behavioral models for combat simulations using techniques for the evaluation of human performance. Technical Report. *Naval Postgraduate School*, Moves Institute. Monterey, CA.

> "Prior to their use in simulations and analytical studies, DoD models are required to undergo the verification, validation, and accreditation (VV&A) process in an attempt to establish an acceptable level of credibility. In general, the human behavioral model validation process, as outlined by the Defense Modeling and Simulation Office (DMSO), is not extendable to meet requirements for validating the varied and complex behavioral models in use or under development for DoD simulations. This paper reviews several issues with validating human behavior representation (HBR) and identifies potential practices for enhancing the validation process for current and future human behavioral models for use in or application to combat simulations."

❖ Challenges

- 1. Complexity of Human Behavioral Models
    - ⇒ "With physics based models, there are established procedures for performing VV&A that allow developers and users to understand the strengths and limitations of a model. For cognitive models, the procedures are not as well established and are often limited in their execution and in the information they provide. Understanding the human thought and decision making processes is complex and evolving."
- 2. DMSO Identified Validation Issues:
    - ◊ Large number of possible actions
        - ⇒ "The first is that for even simple human behaviors, the set of possible actions is normally very large. This makes it difficult to ensure examination of all viable solutions."
    - ◊ Non-linearity of the constrained space of consideration
        - ⇒ "The nonlinearity of the space prevents a simple causal relationship to be drawn between situational parameters and resulting actions."
    - ◊ Stochastic features in models
        - ⇒ "This "unpredictable" characteristic, unless it can be forced to be deterministic, often makes repeatability impossible for a model therefore making model validation more difficult, and frequently impossible."
    - ◊ Chaotic behavior
        - ⇒ "Chaotic behavior exhibited by behavior models that are sensitive to initial and boundary conditions. Models with such issues are limited to the breadth of their validation and to the set of scenarios where they exhibit stable behavior."
- 3. Referent bias
    - ⇒ "Because of the vast spectrum of potential situations and human responses, the identification and collection of referent are often limited."

        - ⇒ "The validation process is inconsistently applied because it is performed by multiple V&V agencies with non-standard criteria or non-uniform referent."
        - ⇒ "In essence, one must find results from other valid HBR models or build and validate another cognitive model to provide referent for validation of a new cognitive model. This dependence on other models makes validation using psychological and physiological correspondences tenuous at best."
- 4. Model Representation
    - ⇒ "Problems occur when a model is fed unique/new inputs for which real world outputs have not been recorded. In these situations, it is not clear if the model's results adequately represent probable or possible actions."
- 5. Use of Subject Matter Experts
    - ⇒ "The use of SMEs to evaluate the results of a simulation is analogous to the use of introspection."
    - ⇒ "According to a meeting of validation experts at Foundations '02, there are at least three major issues with the use of SME: perspective, performance, and perception."
  - ◊ Perspective
  - ◊ Performance
  - ◊ Perception
- 6. Limitations with face validation of overt behaviors
    - ⇒ "using results based, overt behavior validation of HBR systems often fails to capture the flexibility of the model. This method of validation also falls short of covering the dynamic problem space in which such a model could be asked to operate."
- 7. Cost
    - ⇒ "validation is routinely left to the end of the model development process and limited to the remaining funds and time available."

❖ Recommendations

- Improve Quality of Subject Matter Experts
  - ◊ "A set of standards for identifying and accrediting SMEs"
    - ⇒ "Selecting and certifying SMEs would ensure a minimum set of standards for SMEs, provide validation agents with a pool of potential SMEs, and increase the credibility of SMEs."
  - ◊ "Training SMEs to help provide them with a set of skills to help them focus their validation efforts"
    - ⇒ "Along with certification is the requirement for a training program to ensure potential SMEs can gain the necessary knowledge of models and simulations and the validation process so they can prepare to complete a certification process. … To help limit the bias of SMEs, they should 1) be familiar with the validation process and different validation techniques, 2) have at least a basic understanding of the different types of simulations and their purposes, and 3)

be exposed to different types of data displays to help them prepare for the potential systems to which they could be exposed and help reduce misconceptions of simulation capabilities and intent.”

- Cognitive Task Analysis
    - ⇒ “Cognitive Task Analysis (CTA) is an extensive/detailed look at tasks and subtasks performed by a person to achieve a goal. … Such an analysis could be used as bases for collecting the referent used for the development and validation of HBR.”
- Human Performance Evaluation
    - ⇒ “Based on the model representation used and the level of validation one attempts to accomplish, HBR models processes and results could be categorized and evaluated based on one of three domains: psychomotor, cognitive, and affective. Within these categories, there are levels of complexity that can be discovered and evaluated based on the types of actions and responses a model portrays.”
    - ⇒ “Using CTA and human performance evaluation techniques would help model developers collect referent and validation agents develop questions to focus SME efforts.”

Goodfellow, I., Shlens, J., & Szegedy, C. (2015). Explaining and harnessing adversarial examples. *ICLR*.

While Goodfellow et al. (2015) is not a paper directly about T&E, the techniques described will be critical for providing assurance for sub-symbolic-AI systems, and the test recommendations are implicit. The authors describe both a technique for more efficiently generating adversarial examples, and use this to better explore the underlying nature of adversarial perturbations. They recommend that this technique be used to train networks until improvement against adversarial examples asymptotes—not just until real task performance levels off. The implicit test recommendation is that adversarial and real examples should be provided to the system to test its robustness against these techniques.

❖ Challenges

- Sub-symbolic machine learning systems are vulnerable to adversarial disruption
  - ⇒ "These results suggest that classifiers based on modern machine learning techniques, even those that obtain excellent performance on the test set, are not learning the true underlying concepts that determine the correct output label. Instead, these algorithms have built a Potemkin village that works well on naturally occurring data, but is exposed as a fake when one visits points in space that do not have high probability in the data distribution."
  - ⇒ "…adversarial perturbations generalize across different clean examples."

❖ Recommendations

- Deep nets should be trained and tested against adversarial examples
  - ⇒ "Linear models lack the capacity to resist adversarial perturbation; only structures with a hidden layer (where the universal approximator theorem applies) should be trained to resist adversarial perturbation."

Greer, K. (2013). A metric for modeling and measuring complex behavioral systems. *IOSR Journal of Engineering, 3(11),* 19-28.

This paper proposes a method for providing what is essentially a first-pass evaluation of multi-agent stigmergic systems—systems that are individually simple but collectively can solve complex problems, e.g., swarms. The author proposes using a series of equations and behavioral scripts to estimate—prior to simulation—whether a multi-agent system will produce desired behavior. The proposal calls for manually crafting the behavioral script and assuming individual agent capabilities. These are activities done already for agent-based modeling (ABS), which is a common recommendation for space coverage prior to higher-fidelity multi-agent simulation (MAS). The gap this method fills is providing a rough estimate of an ABS prior to expending computational resources on executing the simulation.

- ❖ Challenges
  - Complex environments
    - ⇒ "…an autonomous environment, where the agents can perform a number of different, but relatively simple acts, without the help of a guiding system. In this environment it is difficult to control or fully predict the outcome of the agent activities…"
  - Achieving the sufficient level of modeling fidelity
    - ⇒ "Co-adaptation between agents and environments can also give rise to emergent complexity"
    - ⇒ "The problem is if a system … contains a large number of agents … it would be helpful to know … if the system will in fact be able to carry out the required tasks. This would save time modelling the problem, if the agents are discovered to be too simple and also with evaluating the result of any simulation."
  - Assigning credit or blame in multi-agent system of systems
    - ⇒ "If something goes wrong with the process, then it might be difficult to identify the source of the problem, is it agent1 and behavior *x* or agent2 and behavior *y*?"

- ❖ Recommendations
  - Use equations and behavioral scripts to estimate agent success
    - ⇒ "…this metric will help give an initial indication as to how suitable the agents would be for solving the problem. The system is modelled as a script, or behavioral ontology with a number of variables to represent each of the behavior attributes. The set of equations can be used both for modeling and as part of the simulation evaluation."

Gunning, D. (2017, November). DARPA XAI Program Update.

This briefing is used as a stand-in for multiple program updates given on the XAI program since 2017. Relevant to T&E, the methods to measure explanation effectiveness and discover models from black boxes are not well-developed, and the XAI program is setting out to create them. However, while progress is being made, researchers are reporting that by using model induction, they are discovering that black boxes have often developed idiosyncratic/overfit/ungeneralizable solutions.

❖ Challenges
- Opaque Decision Making
    - ⇒ "Explainable AI will be essential if users are to understand, appropriately trust, and effectively manage this incoming generation of artificially intelligent partners."

❖ Recommendations
- Model Induction
    - ⇒ "[Develop] techniques to infer an explainable model from any model as a black box."
- Measure Explanation Effectiveness
    - ◊ Borrow from psychology
        - ⇒ Explanation Framework that involves tracking decisions based on explanation feedback to the user.
        - ⇒ Categorization of measures include user satisfaction, mental models, task performance, trust assessment, and correct-ability.

Harikumar, J., and Chan, P. (2019). Developing knowledge and understanding for autonomous systems for analysis and assessment events and campaigns. *CCDC Data & Analysis Center, Army Research Lab*.

The authors discuss the "analysis and assessment" (A&A) as part of the T&E of autonomous systems. The authors present challenges that autonomous systems pose for analysis and assessment and pose a list of 13 questions to aid in the analysis and assessment of autonomous systems. Within the definition of autonomous that is used for this article, systems are differentiated as automated systems, autonomous systems, unmanned systems, adaptive systems, and intelligent systems. Challenges in this area are classified as direct, indirect, distributed across multiple systems based on hierarchy and organization, and human-autonomy. A method of viewing autonomous systems through the same 3 basic subsystems for testing is proposed (sensors, logic circuits/AI, and actuators), suggesting testing each separately before testing the whole system. Finally, the authors present exemplar measures and metrics to analyze performance while reviewing previous efforts at building frameworks for measures and metrics in autonomous systems.

- ❖ Challenges
  - Direct challenges
    - ◊ Emergent Behavior and Recovery
      - ⇒ "If the autonomous system exhibits emergent behavior that is within 'normal' acceptable operational range for the given mission, adapts to the 'right' (local, global) parameter value sets to complete mission goals, has an acceptable self-determination of normal state, and can recover to normal state within a reasonable amount of time."
  - Indirect Challenges
    - ◊ "Autonomous system use in new or scaled missions
    - ◊ Autonomous system integration with other homogeneous systems in larger missions."
  - Distributed Challenges, in a cluster of multiple autonomous systems
    - ◊ "'Social' adaptation
    - ◊ Hierarchical behavior
    - ◊ 'Friend or foe' determination
    - ◊ Detection of abnormal behavior of other elements"
  - Human-Autonomy Challenges
    - ◊ Trust and Evaluation of Trust
    - ◊ Human Response to autonomous system behavior
    - ◊ Complexity
      - ⇒ These challenges involve, "human trust of the system decision making, human response to the autonomous system behavior, and challenges with evaluation of this trust," and the increasing complexity of these challenges with multiple, varied systems.

❖ Recommendations

- I – Answer questions about the framework to be used for A&A and the environment/uncertainty within which an autonomous system will operate
  - ◊ Questions listed serve to guide the development of autonomous systems A&A, for example:
    - ⇒ “Q3: How does one quantify the success of decision making?”
    - ⇒ “Q8: How do we quantify the level of distraction to the Warfighter using the autonomous system?”
    - ⇒ “Q9: What is the differential advantage in using an autonomous system (comparative analysis of survivability benefits gained about new vulnerabilities introduced by the autonomous system)?”
    - ⇒ “Q13: What is the expected frequency and magnitude of abnormal, destabilizing, or ‘outlier’ events?”
- II – Separate subsystems and test them individually before a combined test
  - ◊ Separate systems into sensors, logic/AI, actuators
    - ⇒ Individual bench testing of sensors; seeing their adequacy for an autonomous system’s “mission needs.”
    - ⇒ “Ontological testing” for “discrete” systems, and “continuous systems” during training.
    - ⇒ Bench testing of actuators.
  - ◊ Test each system’s inputs into each other
- III – Develop measures and metrics of evaluation to address aforementioned challenges
  - ◊ Find measures to analyze a system’s change in environment
    - ⇒ E.g., “Number of times the system changed its behavior because of a change in environment”, “Number of times the system behavior change was restricted by outside control.”
  - ◊ Find measures of “Autonomous System Intelligence”
    - ⇒ E.g., “Could the system perform its tasks in unstructured environment (Yes/No)”, “Number of steps the autonomous system took to complete the task.”
  - ◊ Find measures of system capability
    - ⇒ E.g., “Analyze the choice set made by the autonomous system prior to the autonomous system response.”
  - ◊ Other examples, e.g. “Learning rate” with a metric being the “Weighted sum of learning from saved historic data and temporal update rate”.
- IV – Recommends building on previous autonomous system framework development efforts
  - ◊ Examples include Autonomy Levels for Unmanned Systems (ALFUS) and Performance Measures Framework for Unmanned Systems (PerMFUS).

Haugh, B., Sparrow, D., & Tate, D. (2018). The status of Test, Evaluation, Verification, and Validation (TEV&V) of autonomous systems, P-9292. *Institute for Defense Analysis*, Alexandria, VA.

"This memorandum enumerates TEV&V challenges that have been identified by the commercial, academic, and government autonomy research communities; describes the focus of current academic research programs; and highlights areas where current research leaves unaddressed gaps in capability."

- ❖ Challenges
  - Instrumenting machine thinking:
    - ⇒ Providing adequate internal "instrumentation" to identify where errors are: at the conventional locations in cyber-physical systems (sensor, software/hardware, effector), or at more subtle functional areas of perception/cognition/effectuation.
  - Linking system performance to autonomous behaviors:
    - ⇒ Understanding how component functions (or failures) contribute to overall behavior (or misbehavior).
  - Comparing AI models to reality:
    - ⇒ Understanding how closely the AS's internal "mental model(s)" correspond with reality, for model-based Ass.
  - CONOPS and training as design features:
    - ⇒ Recognizing that ASs working with other ASs or humans will, because of their flexibility and potential initiative, likely serve to drive team behaviors and overall CONOPS, in manners unforeseen at the time of design. "This will pose organizational and personnel challenges to T&E, as well as methodological challenges."
  - Human trust:
    - ⇒ Ensuring appropriate multi-dimensional levels of human trust of the system [and by the system of the human].
  - Elevated safety concerns and asymmetric hazard:
    - ⇒ Because the cost of AS errors is potentially high (unlike Go-playing algorithms), and because the potential for errors will increase with system complexity and degrees of decisional freedom not normally given to conventional military systems, expected utility/loss (product of likelihood and cost), extra concern will need to be taken during DT and OT.
  - Exploitable vulnerabilities:
    - ⇒ ASs may have expanded "attack surfaces" not normally present in conventional cyber-physical systems, for example perceptual misapprehensions caused by "poisoned" training data, or simple but unanticipated CCD activities taken by a

target. Active counter-AI activities are currently underway to identify such vulnerabilities.

- Emergent behavior:
    - ⇒ Unanticipated behaviors can result with a single AS encountering a complex or untested-for environment, or from the interactions with one or more humans (friendly or adversary) or with one or more ASs. The breadth of these potential encounters is a challenge in itself.
    - ⇒ ‘US DoD Directive 3000.09 specifically warns against the possibility of “unanticipated emergent behavior resulting from the effects of complex operational environments on autonomous or semi-autonomous systems.”’
- Post-fielding changes:
    - ⇒ Systems that learn over time during post-fielding operations may change their behaviors over time, calling for a need for either predictive models of how those behaviors will be instantiated, or for calling back from the field for on-going intermittent testing.
- Verification and validation of training data:
    - ⇒ If data is used to train the AS, then the data needs to undergo a VV&A process as well, to ensure that the system is not “miss-trained” inadvertently, or by adversary design.

❖ Recommendations

- Use formal methods for AS verification:
    - ⇒ Formal methods have been used for expensive one-time development efforts in fairly predictable environments (e.g., space probes; see T. Menzies and C Pecheur, 2004), but it is unlikely they will be scalable to the potential environment (natural and adversarial) a military system is likely to encounter (“state space explosion”). “Use these methods where possible.”
- Cognitive instrumentation:
    - ◊ Software Instrumentation
        - ⇒ *Design the architecture and the code to provide “software instrumentation” of the AS functions and operations*. This will not only serve the purpose of providing the AS with SA of its own functioning and system health, but it can provide a basis for explaining the basis of its perceptions/decisions/actions to teammates (both machines and humans).
- Explainable AI:
    - ◊ Ability to explain perceptions/decisions/actions
        - ⇒ Provide an *ability to explain perceptions/decisions/actions* to developers, testers, and operators, to support diagnosis/debugging, effective deployment options (CONOPS development), and determining limits of dependable performance within a multi-dimensional envelope.
    - ◊ Trust:

- ⇒ Support appropriate levels of trust by developers/testers/operators, via explaining an ASs behaviors.

- Adversarial testing:
  - ⇒ Red team the subject AS using humans and/or other ASs, to look for “corner cases” where AS behaviors break down. Use to augment conventional DOE methods.
- Run-time monitoring:
  - ⇒ Incorporate a simpler run-time supervisor of AS behaviors so that when unacceptable behavior is identified, the AS can switch to a “back-up” mode which is more deterministic, and less likely to violate behavioral boundaries. Of course, building the monitor may entail its own issues, with regard to continuous monitoring, detection of impending or ongoing unacceptable behavior, and graceful switchover or takeover of ongoing AS functions.
- Resources and Tool Development:
- Data:
  - ⇒ Because the most spectacular recent advances in machine perception have been with deep learning over massive datasets, we need to pay attention to the veracity of those datasets. Data V&V is called for.

Hernandez-Orallo, J. (2017). Evaluation in artificial intelligence: From task-oriented to ability-oriented measurement. *Artificial Intelligence Review, 48,* 397-447.

This article presents a lengthy review of evaluation methods for AI. Broadly, he breaks down methods into two categories: task-driven and ability-driven evaluation. Task-driven has been the dominant approach in the field for decades, with three mains types: human discrimination, benchmarking, and peer competition/relative performance. While useful so far, the author argues these are hitting their limit, as they apply best to constrained, well-defined tasks. As we move towards systems that must perform many different and more complex tasks, trying to evaluate performance on all tasks across the varied conditions they will encounter quickly becomes intractable. Instead, he advocates for using ability-oriented evaluation: try to quantify general or holistic skill capability rather than specific task performance. The authors acknowledge this approach is much less mature, but identify it as the necessary solution to the challenge of massively growing problem spaces.

- ❖ Challenges
  - Conflation of evaluating weak vs. soft AI
    - ⇒ "McCarthy's pristine definition of AI sets this unambiguously: "[AI is] the science and engineering of making intelligent machines" (McCarthy 2007). As a consequence, AI evaluation should focus on evaluating the intelligence of the artefacts it builds. However, as we will further discuss below, 'intelligence tests' (of whatever kind) are not the everyday evaluation approach for AI. The explanation for this is that most AI research is better identified by Minsky's more pragmatic definition: "[AI is] the science of making machines capable of performing tasks that would require intelligence if done by [humans]" (Minsky (1968), p. v). As a result, AI evaluation focuses on checking whether machines do these tasks well."
    - ⇒ "The intention of stressing this duality is that this should necessarily pervade the evaluation procedures in AI. Specialised AI systems should require a task-oriented evaluation, while general-purpose AI systems (also known as AGI systems, from the term 'artificial general intelligence') should require an ability-oriented evaluation. In practice, however, we see that some general-purpose AI systems are evaluated with a narrow set of tasks."
    - ⇒ "[For task-based evaluation], we will identify several problems, most of them derived from the confusion of a task definition with its evaluation."
  - Inconsistency in evaluation approach
    - ⇒ "However, there is still a great deal of disaggregation, many ad-hoc procedures, bad habits and loopholes about what is being measured and how it is being measured."

        - ⇒ “evaluation efforts are extremely scattered across AI disciplines, and it is quite common to find duplicated efforts, for which solutions have to be found again and again. Apparently, there is limited exchange of experiences between them, just because the domains are different.”
- Task-based Evaluation of AI Challenges
    - ◊ Selecting Test Cases
        - ⇒ “An appropriate sampling procedure from the class of problems defining the task is not always easy.”
    - ◊ Non-determinism
        - ⇒ “As AI systems become more sophisticated, white-box assessment becomes more difficult, if not impossible, because the unpredictability of complex systems. Many AI systems incorporate many different techniques and have stochastic behaviours.”
    - ◊ Evaluation by Human Discrimination
        - ⇒ “If done properly, it may take too much time.”
    - ◊ Evaluation through problem benchmarks
        - ⇒ “However, as the public access to the benchmark before the evaluation can lead to “evaluation overfitting”, there have also been some occasional evaluations in AI following a “secret generalized methodology.”
    - ◊ Evaluation through peer confrontation
        - ⇒ “…we evaluate a system by letting it compete against another system. … The results of each match (possibly repeated with the same peer) may serve as an estimation of which of the two systems is best (and how much). Nonetheless, the main problem about this approach is that the results are relative to the opponents.”
- Ability-Based Evaluation of AI Challenges
    - ◊ Inapplicability of current popular methods
        - ⇒ “The three types of evaluation seen for task-oriented evaluation are not directly applicable, as we now do not want to evaluate systems for what they do but for what they are able to (learn to) do.”

❖ Recommendations

- Switch to behavioral evaluation from algorithmic evaluation.
    - ⇒ “…we will argue that white-box evaluation (by algorithm inspection) is becoming less predominant in AI, and we will devote the rest of the paper to blackbox evaluation (by behaviour).We will distinguish three types of behavioural evaluation: by human discrimination (performing a comparison against or by humans), problem benchmarks (a repository or generator of problems) and by peer confrontation (1-vs-1 or multi-agent ‘matches’).”
- Select representative metrics and test cases
    - ◊ Use large and/or unknown metrics to prevent overfitting

- ⇒ "To avoid or reduce this problem, it is much better if M is very large or infinite, or at least the problems are not disclosed until evaluation time"
- ◊ Use problem generators for where possible
  - ⇒ "Generators can be based on the use of some prototypes with parameter variations or distortions. These prototypes can be "based on reality", so that the generator "takes as input a real domain, analyses it automatically and generates deformations […] that follow certain high-level characteristics"
  - ⇒ "However, it is not always easy to generate a large M of realistic problems (e.g., in a car driving domain)."
- ◊ Diversely and randomly sample from possible test cases
  - ⇒ "Random sampling using p seems to be a more reasonable alternative. … The idea is to sample in such a way that the diversity of the selection is increased."
  - ⇒ Although there are many ways of obtaining a 'diverse' sample, we just highlight two main approaches that can be useful for AI evaluation … *Information-driven sampling*: assume that we have a similarity function sim(μ1,μ2), which indicates how similar (or correlated) exercises μ1 and μ2 in M are. In this case, we need to sample on M such that the accumulated mass on p is high and that diversity is also high. … *Difficulty-driven sampling* … The idea to optimize the evaluation is to choose a range of difficulties for which the evaluation results may be informative."
  - ⇒ "Item response theory (IRT) (Embretson and Reise 2000) estimates mathematical models to infer the associated probability and informativeness estimations for each item."

- Increase Cooperation/Centralize Knowledge Sharing
  - ⇒ "It is then very useful to look at some organisations that can serve to centralise and exchange insights and lessons-learnt across several domains, apart from identifying new needs for benchmarks and competitions."
- Overall Recs
  - ⇒ "Given the caveats mentioned above about AI evaluation, we now enumerate a series of generic guidelines for AI practitioners willing to create or improve a benchmark or competition:
  - ⇒ "The definition of Ω, the set of possible systems that can be evaluated (or that can be opponents in peer confrontation evaluation), must be clarified from the beginning, as well as whether the AI systems are fully autonomous or require the integration and fine tuning of AI researchers. If humans are considered, the way in which they are admitted and how they are instructed must be defined. The more general Ω is the less we can assume about the evaluation process. If Ω is heterogeneous (e.g., a universal test), different interfaces must be considered.
  - ⇒ "The definition of M, the set of possible tasks, and its associated distribution p configure what we are measuring. This can be built from a set of problems or using a generator. This pair _M, p_ has to be representative of a task (in task-

oriented evaluation) or an ability (in ability-oriented evaluation). If it is a peer confrontation evaluation, M will be enlarged with as many combinations between game (environment) and agents in Ω are possible. The distribution p will be updated accordingly.

⇒ “The definition of R and its aggregation Φ must ensure that the values $R(\mu)$ for all $\mu \in M$ are going to be commensurate and that the aggregation is bounded. An analysis about expected measurement error is useful at this point. The robustness of R depending on the length of time left for each episode will indicate whether repetitions are needed to reduce the measurement error given by $R(\mu)$.

⇒ “As much as possible, the similarity between tasks or a set of features describing them should be identified. An intrinsic difficulty function (even if approximate) is always very useful. Showing the distribution of difficulty for M can be highly informative. If difficulty is available, item response curves could be prepared.

⇒ “The sampling method must be as efficient as possible, by using, e.g., an information driven sampling or a range of difficulties if we have a non-adaptive evaluation. For the peer-confrontation evaluation, the arrangement of matches can be designed beforehand if the evaluation is not adaptive. Similarly, the procedure for an adaptive evaluation must also be carefully designed to ensure measurement robustness. Simulations can be useful to estimate this.

⇒ “Information about how the evaluation is performed (including R, Φ and some illustrative problems) can be disclosed to the systems that are being evaluated (or to their designers). However, Ω, M and p should not be disclosed. If possible, the problems should not be disclosed after the evaluation either, as keeping them secret makes it possible to compare with the same problems for different subjects or at different times (e.g., we can evaluate progress of a system or a discipline during a period).

⇒ “After the evaluation, results must be analysed beyond the mere calculation of the aggregated results. Item response functions and agent response functions (Hernández-Orallo et al. 2014) can be constructed empirically from the results and compared with the theoretical functions or any other information about Ω and M. Discrepancies or anomalies may suggest that the evaluation scheme has to be revised. Results of the evaluation must become public at the highest possible detail, so they can be analysed and compared by other researchers and participants (following, e.g., the notion of ‘experiment database’ (Vanschoren et al. 2012), such as in the machine learning community).

Hess, J., & Valerdi, R. (2010). Test and evaluation of a SoS using a prescriptive and adaptive testing framework. *2010 5th International Conference on System of Systems Engineering*, Loughborough, 2010, pp. 1-6.

> “Testers need the ability to adapt test planning on the order of days and weeks. PATFrame will use its reasoning engine to prescribe the most effective strategies for the situation at hand. Strategies in this context include methods of experimental designs, test schedules and resource allocation. By facilitating rapid planning and replanning, the PATFrame reasoning engine will enable users to use information learned during the test process to improve the effectiveness of their own testing rather than simply follow a preset schedule. This capability is particularly attractive in the domain of Systems of Systems testing because the complexity of test planning and scheduling make frequent re-planning by hand infeasible.”
>
> “Our proposed method, [PATFrame], addresses a number of important SoS testing challenges by enabling users to adapt their test plans in light of newly learned information. The ability to re-plan test programs has already shown benefits through exploratory testing, and we believe that our algorithm-driven method for adaptation may provide similar results.”

- ❖ Challenges
  - 1. Time required for testing
    - ◊ How to test effectively in a compressed schedule?
      - ⇒ “Testers need the ability to adapt test planning on the order of days and weeks.”
  - 2. Complexity, Uncertainty in planning, and Adaptability of Testing
      - ⇒ “test organizations and testers in general are asked to make decisions under a great deal of uncertainty.”
      - ⇒ “A well planned test schedule for a major system can be “overcome by events” (OBE) at any time, and we have no way of knowing when or how this will be.”
    - ◊ How much testing is enough?
    - ◊ How to prioritize tests?
  - 3. Formal testing criteria for SoS testing
      - ⇒ “there are virtually no formal SOS requirements to test against in Developmental Testing, meaning we often rely on Stewart-esque "I know it when I see it" criteria to define SOS failures.”

- ❖ Recommendations
  - Prescriptive and Adaptive Test Framework (PATFrame)
      - ⇒ “By facilitating rapid planning and replanning, the PATFrame reasoning engine will enable users to use information learned during the test process to improve the effectiveness of their own testing rather than simply follow a preset schedule.”
    - ◊ Design of Experiments

- ⇒ “Design of Experiments is a set of methods for efficiently gathering information. The simplest incarnations of DOE facilitate planning tests of a system by taking into account information already known about dependent variables and their effects (including interactions with one another) on the independent variables being measured.”

- ◊ Defect Modeling
  - ⇒ IBM proposed using information regarding tracked defects “to project the number of remaining defects, and the rate at which they will be discovered. This method is used to answer the question … “When am I done testing?”.”
- ◊ Exploratory Testing
  - ⇒ “ET, a strategy used in software testing, leverages human judgment to improve a test program “on-the-fly”. As opposed to scripted testing, ET encourages the tester to use information learned from their previous tests to select the next test case.”

Hill, & Thompson, G. (2016, December 28). Five giant leaps for robotkind: Expanding the possible in autonomous weapons. *War on the Rocks*. https://warontherocks.com/2016/12/five-giant-leaps-for-robotkind-expanding-the-possible-in-autonomous-weapons/

This study focuses on five operationally-oriented challenge problems, all of which are associated with AS discovery, improvisation, and "initiative", a particular problem area for T&E:

- ❖ Challenges
  - Hostage rescue using discriminating lethality
  - Deployment under disrupted/degraded comms
  - On the spot improvisation of materiel solutions
  - Adaptation for discovery of new tactics/TTPs
  - Evoking "disciplined initiative" when original plan is failing/illegal/immoral

Ilachinski, A. (2017). AI, robots, and swarms: Issues, questions, and recommended studies. *Center for Naval Analyses*, white paper.

This white paper presents a high-level overview of the challenges of acquiring AI-enabled systems, with a strong focus on certification/T&E issues. The author argues that the DoD acquisition apparatus is completely unprepared for the complexity, fast development, and non-static nature of these systems, and massive change is needed. However, the author provides a mix of mid-level recommendations (e.g., use non-traditional modeling techniques like agent-based modeling) and calling for research areas (e.g., figure out what cyber for AI/AS will look like and how to do it).

❖ Challenges

- "Devil is in the details" research hurdles
  - ⇒ "Developers of autonomous systems must confront many of the same fundamental problems that the academic and commercial AI and robotic research communities have struggled for decades to "solve." To survive and successfully perform missions, autonomous systems must be able to sense, perceive, detect, identify, classify, plan for, decide on, and respond to a diverse set of threats in complex and uncertain environments. While aspects of all these "problems" have been solved to varying degrees, there is, as yet, no system that fully encompasses all of these features."
- Complex and uncertain environments
  - ⇒ "Autonomous systems must be able to operate in complex possibly, a priori unknown environments that possess a large number of potential states that cannot all be pre-specified or be exhaustively examined or tested. Systems must be able to assimilate, respond to, and adapt to dynamic conditions that were not considered during their design. This "scaling" problem i.e., being able to design systems that are developed and tested in static and structured environments, and then have them perform as required in dynamic and unstructured environments is highly nontrivial."
- Emergent Behavior
  - ⇒ "Emergent behavior: For an autonomous system to be able to adapt to changing environmental conditions, it must have a built-in capacity to learn, and to do so without human supervision. It may be difficult to predict, and be able to account for a priori unanticipated, emergent behavior (a virtual certainty in sufficiently "complex" systems-of-systems dynamical systems).
- Human-machine interactions
  - ⇒ "Human-machine interactions/I: The operational effectiveness of autonomous systems will depend on the dynamic interplay between the human operator and the machine(s) in a given environment, and on how the system responds, in real time, to changing operational objectives, in concert with the human's own

adaptation to dynamic contexts. The innate unpredictability of the human component in human-machine collaborative performance only exacerbates the other challenges identified on this list.

⇒ "Human-machine interactions/II: The interface between human operators and autonomous systems will likely include a diverse space of tools that include visual, aural, and tactile components. In all cases, there is the challenge of translating human goals into computer instructions (e.g., "solving" a longstanding "AI problem" of natural language processing), as well as that of depicting the machine's "decision space" in a form that is understandable by the human operator (e.g., allowing the operator to answer the question, "Why did the system choose to take action X?")."

- Non-agility of DoD Acquisition

⇒ "By way of comparison, note that within roughly this same interval of time, the commercial AI research community has gone from just experimenting with (prototypes of dedicated hardware-assisted) deep learning techniques, to beating the world champion in Go (along with achieving many other major breakthroughs)."

⇒ "Of course, DoD acquisition challenges, particularly for weapons systems that include a heavy coupling between hardware and software, have been known for decades. However, despite numerous attempts by various stakeholders to address these challenges, the generic acquisition process (at least on the traditional institutional level) remains effectively unchanged."

- Complexity of the state space

⇒ "…it is impossible to conduct an exhaustive search of the vast space of possible system "states" for autonomous systems…"

- Complexity of the environment

⇒ "the behavior of an autonomous system cannot be specified—much less tested and certified—in situ, but must be tested in concert with interaction with a dynamic environment … rendering [the problem] combinatorially intractable"

- Unpredictability

⇒ "to the extent that autonomous systems are inherently complex adaptive systems, novel or unexpected behavior can be expected to arise naturally and unpredictably in certain dynamic situations; existing T&E/V&V practices do not have the requisite fidelity to deal with emergent behavior."

⇒ "Gap 2: An underappreciation of the unpredictable nature of autonomous systems, particularly when operating in dynamic environment, and in concert with other autonomous systems. Existing T&E/V&V practices accommodate neither the basic properties of autonomous systems, as expected by AI and indicated by decades of deep fundamental research into the behavior of complex adaptive systems, nor the requirements they must meet, as weapon systems (as spelled out by DoD Directive 3000.09)."

- Trust

  - ⇒ "existing T&E/VV&A practice is limited to testing systems in closed, scripted environments, since "trust" is not an innate trait of a system"
- Experience and/or learning
  - ⇒ "to be more effective, autonomous systems may be endowed with the ability to accrue information and learn from experience. But such a capability cannot be certified monolithically, during one "check the box" period of time. Rather, it requires periodic retesting and recertification, the periodicity of which is necessarily a function of the system's history and mission experience. Existing T&E/V&V practices are wholly inadequate to address these issues."
- Lack of a coherent theoretical framework for AI/autonomy
  - ⇒ "Gap 3: A lack of a universally agreed upon conceptual framework for autonomy that can be used both to anchor theoretical discussions and to serve as a frame-of-reference for understanding how theory, design, implementation, testing, and operations are all interrelated."

❖ Recommendations

- Incorporate T&E throughout acquisition cycle
  - ⇒ "Step 3: Moving design, development, testing, and accreditation through the DoD acquisition process (and accommodating autonomy's unique set of technical challenges while doing so)."
- Develop an operationally meaningful conceptual framework for autonomy.
  - ⇒ "For example, build on lessons learned from the National Institute of Standards and Technology's (NIST's) stalled evolution of its ALFUS (Autonomy Levels for Unmanned Systems) framework, and develop the skeleton of an idea proposed by DoD's Defense Science Board's 2012 report on autonomy."
- Develop measures of effectiveness (MOEs) and measures of performance (MoP) for autonomous systems.
  - ⇒ "Develop a methodology by which the effectiveness of autonomous systems can be measured at all levels (e.g., developers, program managers, decision-makers, and warfighters) and across all required functions, missions, and tasks (e.g., coordination, mission tasking, training, survivability, situation awareness, and workload)."
- Use nontraditional modeling and simulation (M&S) techniques to help mitigate AI/autonomy-related dimensions of uncertainty.
  - ⇒ "…M&S is moving away from "simulations as distillations" of real systems … to "simulation-based rules and algorithms as descriptions" of real (i.e., engineered) robots and behaviors. … For example, while "swarm engineering" methods exist to facilitate the unique design requirements of robotic swarms, no general method exists that maps individual rules to (desired) group behavior. Multi-agent based modeling techniques are particularly well suited for developing these rules, and, more generally, for studying the kinds of self-

organized emergent behaviors expected to arise in coupled autonomous systems."

- Develop new T&E/V&V standards and practices appropriate for the unique challenges of accrediting autonomous systems.
  - ⇒ "For example, help ameliorate basic gaps in testing in terms of accommodating complexity, uncertainty, and subjective decision environments, by appealing to and exploiting lessons learned from the development and accreditation practices established by the complex system theory and multi-agent-based modeling research communities."
- Explore basic human-machine collaboration and interaction issues.
  - ⇒ "As autonomy increases, human operators will be concerned less with the manual control of a vehicle, and more with controlling swarms and directing the overall mission: "What are the operator's informational needs (and workload limitations) for controlling multiple autonomous vehicles?" "How do humans keep pace with an accelerating pace of autonomy-driven operations?" "What kinds of command-and-control relationships are best for human-machine collaboration?" "How are human and autonomous-system decision-making practices optimally integrated?" and "What data practices are key to developing shared situation awareness?"
- Explore the cyber implications of autonomous systems.
  - ⇒ "Explore what new features increased AI-driven autonomy brings to the general risk assessment of increasingly autonomous unmanned systems. On one hand, autonomy may potentially reduce a force's overall vulnerability to jamming or cyber hacking. … On the other hand, autonomy itself may also be more, not less, vulnerable to a cyber intrusion."

Kapinski, J., Deshmukh, J., Jin, X., Ito, H., & Butts, K. (2016). Simulation-based approaches for the verification of embedded control systems. *IEEE Control Systems Magazine*, 45-64.

This paper reviews methods for providing assurance about complex (but symbolically coded) software controlling physically embedded systems. They identify the need to move through the trade-space of providing more approximate answers with more scalable techniques as a key problem in complex autonomous software. They recommend model-based development, where developers and/or testers create models at different levels of approximation depending on system maturity. The paper reviews a number of concrete techniques and how to apply them to different stages of development.

- ❖ Challenges
  - Software Complexity
    - ⇒ "Increased autonomy is often achieved by using advanced algorithms that increase the complexity of the control software."
    - ⇒ "…manually generating code in a monolithic manner and then validating the system design with experimental tests. This approach is expensive and difficult to manage for complex systems…"
  - Scalability vs. Approximation
    - ⇒ "Approaches can broadly be classified in terms of how well they account for the possible behaviors of the model and how well they scale. Some techniques, such as model checking, can provide formal guarantees of correctness for all behaviors of software systems, but these do not scale well for many industrial embedded control systems. Simulation, on the other hand, can be applied to models of any scale but only provides an approximation of behavior for a discrete set of operating conditions."
  - System Requirements/Specification
    - ⇒ "All testing and verification approaches rely on some form of requirements, either formal or informal, but the process of creating correct and useful requirements is an often underappreciated activity. Care should be taken to create requirements that accurately reflect the intended behavior of the system."

- ❖ Recommendations
  - Used model-based development (MDB) with levels of modeling fidelity/specificity
    - ⇒ The model lays out the path requirements -> control design model -> specification model -> code -> platform hardware -> platform + plant in a "V" shape with the first 3 topics on the left hand side corresponding to the earlier phase, and the later 3 on the right hand side corresponding to the later phase.
    - ⇒ "Focus on control algorithms, high-level requirements, easier and cheaper to debug and repair code" in the earlier phase. "Focus on control implementations,

real-time/platform-aware requirements, harder and more expensive to debug and repair” In the later phase.

⇒ Additionally, requirements inform platform + plant directly, control design model informs platform hardware directly, and specification model informs code directly.

⇒ “The earlier stages of the development are associated with the left side of the design V. Identifying a problem with the control design at these early stages results in less expensive rework than if the problem is identified later in the development process.”

Laverghetta, T., Leathrum, J., & Gonda, N. (2019). Integrating virtual and augmented reality based testing into the development of autonomous vehicles. Old Dominion University, Norfolk VA, *MODSIM World 2019*.

There are multiple challenges to the development of autonomous systems. These make testing difficult and raise the requirement bar for testing a system to ensure safety. To test systems more efficiently, we should integrate virtual reality and augmented reality. We can augment real-world tests with virtual features, and we can augment a virtual test with real stimuli. This way we can safely test the system early on in a virtual environment, then slowly add features of reality to augment the fidelity of the virtual environment, concluding with real world testing. Using a software framework, the autonomous system experiences a seamless transition between virtual and real-world testing, ignorant to whether it is being tested in a virtual or real environment.

- ❖ Challenges
    - Traditional testing paradigms cannot be relied on for autonomous systems
        - ◊ It may not be possible to understand the reasoning behind an AI's decision
            - ⇒ "… it may not be possible to determine why the software made a decision."
        - ◊ It may be difficult to test subsystems in isolation from the greater system and the environment
            - ⇒ "But it may be difficult to test subcomponents of the system in the absence of the complete system and the ability to perceive the environment"
        - ◊ Fully autonomous systems do not have humans to recognize and address unintended behavior
            - ⇒ "First, in fully autonomous vehicles, there is no human backup to address faults, malfunctions, and unexpected operating conditions."
            - ⇒ "Thus, the autonomy software must have significant additional complexity to address all potential contingencies, making testing more difficult"
    - Decision making by Autonomous Systems are often non-deterministic
        - ◊ Probabilistic sampling and distributions are used in decision making
            - ⇒ "Second, autonomous software often utilizes non-deterministic components and statistical algorithms"
            - ⇒ "This makes it difficult to evaluate the results of testing because there is no uniquely correct result for a given test scenario and the tests are non-repeatable"
    - High standards for Safety
        - ◊ Autonomous systems need high degrees of confidence in safety to be operated fully autonomously.
            - ⇒ "A failure of the software could result in the destruction of property and loss of life."

        - ⇒ “Such vehicle testing is time consuming and expensive; often it simply is not feasible to conduct enough tests with the physical vehicle to ensure desired safety levels”

❖ Recommendations

- Test subsystems in isolation using a black box simulation/model
    - ⇒ “[Difficulty in testing isolated subsystems] presents the opportunity for system and environment simulations to drive the black box testing, providing the stimuli to the subcomponents to allow the observation of their behavior”
- Transitioning Testing from VR and AR into Real World Testing
    - ◊ Tests early into the development lifecycle, gradually increasing in resolution
        - ⇒ “It also presents the opportunity to test early in the design and development system, known to reduce development costs.”
    - ◊ Use of a software framework/virtualization to seamlessly transition between simulated and real world.
        - ⇒ “The software under test should be oblivious of whether operating in a virtual/test environment or in physical operating conditions.”
    - ◊ Gradually introduce real sensor input/output into the virtual test.
        - ⇒ “… slowly integrating actual computational, sensing, and motion capabilities as the hardware becomes available.”

Lede, J. (2019, April 5). Autonomy overview. *US-Japan Service to Service Dialogue.*

This briefing is a short recap of the challenges and recommendations made by the Autonomy CoI groups, e.g., Ahner and/or Lennon. The author calls for testable requirements, new analysis tools for compositional verification, re-use and relicensing of previously approved capabilities, run-time monitors as a prime driver of assurance, and assurance cases.

❖ Challenges

- “Requirements that are mathematically expressible, analyzable, and automatically traceable to different levels of autonomous systems”
  - ⇒ “Dynamic requirements generation & feedback”
  - ⇒ “Design time and run time transparency”
- “Methods and tools enabling the compositional verification of the progressive design process”
  - ⇒ “Trust/transparency in design”
  - ⇒ “’Correct by construction’ synthesis”
- “Systems that are ‘licensed’ to perform functions in limited cases”
  - ⇒ “Applicability of Learning Algorithms”
  - ⇒ “Pedigree-Based Licensure”
- System constrained by set of allowable, predictable, and recoverable behaviors, shifting analysis/test burden to more deterministic run-time assurance mechanism”
  - ⇒ “Run time analysis prediction”
  - ⇒ “Transparency models for past performance and future behaviors”
- “Argument based notations, structures and semantics of arguments, implicitly tied to requirements”

❖ Recommendations

- “Methods, Metrics, and Tools Assisting in Requirements Development and Analysis”
  - ⇒ “Precise, structured standards to automate requirement evaluation for testability, traceability, and consistency”
- “Evidence-Based Design and Implementation”
  - ⇒ “Assurance of appropriate decisions with traceable evidence at every level to reduce the T&E burden”
- “Cumulative Evidence through Research, Development, and Operational Testing”
  - ⇒ Progressive sequential modeling, simulation, test, and evaluation to record, aggregate, leverage, and reuse M&S/T&E results throughout engineering lifecycle”
- “Run-time Behavior Prediction and Recovery”

    - ⇒ “Real time monitoring, just-in-time prediction, and mitigation of undesired decisions and behaviors”
- “Assurance Arguments for Autonomous Systems”
    - ⇒ “Reusable assurance case-based on previously evidenced ‘building blocks’”

Lennon, C., & Davis, E. (2018, September). Autonomy Community of Interest: Test & Evaluation, Verification & Validation Working Group. *US-UK Test & Evaluation Meeting*.

This briefing is a short recap of the challenges and recommendations made by the Autonomy CoI groups, e.g., Ahner and/or Lennon (see also Lede, 2019). The authors call for testable requirements, new analysis tools for compositional verification, re-use and relicensing of previously approved capabilities, run-time monitors as a prime driver of assurance, and assurance cases.

❖ Challenges

- “State-Space Explosion”
    - ⇒ “Algorithm state space cannot be exhaustively searched, and lack of understanding of that space limits simplifying assumptions”
- “Unpredictable Environments”
    - ⇒ “Lack of understanding of how system will react to its environment”
- “Emergent Behavior”
    - ⇒ “Interactions between complex adaptive systems; how to constrain behavior at design time & run time”
- “Human-Machine Communication”
    - ⇒ “Designing transparency and human-system requirements”
- Gaps for ATEVV
    - ◊ “Lack of Verifiable Autonomous System Requirements”
        - ⇒ “Requirements with assumptions; measures of effectiveness”
    - ◊ “Lack of Modeling, Design & Interface Standards”
        - ⇒ “No standardized modeling framework spanning system lifecycle creates lack of traceability between capabilities and requirements”
    - ◊ “Lack of Autonomy T&E capabilities”
        - ⇒ “Ranges, test beds, skill sets”
    - ◊ “Lack of human operator reliance to compensate for brittleness”
        - ⇒ “Human machine interface; human performance & training”
    - ◊ “Lack of Run Time V&V for Deployed Autonomy”
        - ⇒ “Cannot always rely on human supervision; need bounding of behavior”
    - ◊ “Lack of Evidence Re-use for V&V”
        - ⇒ Unsustainable T&E for complex autonomy”

❖ Recommendations

- Accumulate evidence to provide assurance across system lifecycle
    - ◊ Methods & Tools for Requirement Development & Analysis
        - ⇒ Precise, structured standards to automate requirement evaluation for testability, traceability & de-confliction
    - ◊ Evidence-Based Design and Implementation
        - ⇒ Assurance of appropriate decisions with traceable evidence
    - ◊ Cumulative Evidence through RDT&E, DT, OT
        - ⇒ Progressive modeling, simulation, test & evaluation
    - ◊ Run Time Behavior Prediction & Recovery
        - ⇒ Just in time prediction & mitigation of undesired behaviors
    - ◊ Assurance Arguments for Autonomous Systems
        - ⇒ Reusable assurance case based on previous evidence

Lenzi, N., Bachrach, B., & Manikonda, V. (2010, September). DCF – A JAUS and TENA compliant agent-based framework for UAS performance evaluation. *PerMIS '10: Proceedings of the 10th Performance Metrics for Intelligent Systems Workshop,* September 2010, 119–126.

This paper describes an implemented solution to challenges of inadequate simulations of UAS that are not modeled with sufficient attention to either simulation fidelity, interoperability concerns, or the existence of other agents. They propose using common architectures (e.g., LVC) to simulate performance testing of these systems, and discuss their specific implementation of this.

- ❖ Challenges
  - Need a framework
    - ⇒ "To reduce life cycle costs and improve T&E, there is increasing need for a generalized framework that can support the design and development of T&E approaches for multi-UAS teams and validate the feasibility of the concepts, architectures and algorithms.
  - Emergent behavior/non-deterministic
    - ⇒ "This challenge is most significant in the cognitive/social domains, where the development of test approaches and methodologies are difficult because of the emergent nature of behaviors in response to dynamic changes in the battlespace."
  - Inadequacy of current simulations
    - ⇒ "Current simulations rarely capture the complexity of real world effects"
    - ⇒ "…very often high fidelity simulations do not scale as the number of UAS increases. On the other extreme, directly implementing hardware platforms without high resolution simulations to help refine the design induces significant risk"

- ❖ Recommendations
  - Create scalable, human integrative simulation environments using common architectures
    - ⇒ "…To address this need, IAI has developed a JAUS and TENA compliant Integrated Agent-based T&E Framework for Teams of Unmanned Autonomous Systems developed. IAI has also developed the Vignette Editor which fulfills a variety of functions from visualizing the state of the UAS team, creating T&E scenarios and monitoring the UAS team performance."

Luna, S., Lopes, A., Yan See Tao, H., Zapata, F., & Paneta, R. (2013). Integration, Verification, Validation, Test, and Evaluation (IVVT&E) framework for system of systems (SoS). *Procedia Computer Science, 20*, 298-305.

> "Current IVVT&E methodologies focus on the constituent-system levels rather than the testing strategies at the SoS level. This paper proposes an IVVT&E framework for a SoS and utilizes the communications platform of an UAS as an example. The proposed framework allows the early identification of possible systems evolution and knowledge emergence during the system design phase. In addition, the proposed IVVT&E framework is the result of the conjunction of several well-known methodologies, such as DSM, graph theory, and combinatorial/pair-wise testing, which are systematically integrated with a proposed methodology to optimize the IVVT&E activities of a SoS."

❖ Challenges

- Complexity
  - ⇒ "Typically, as the number of systems, interconnections, and interface protocols grows over time, the system complexity increases and the resulting SoS becomes difficult to maintain."
- Constituent level-testing vs. SoS level testing
  - ⇒ "In the current IVVT&E phases, constituent system-level regression testing needs to be expanded to the SoS level to ensure that SoS is not affected by the constituent system upgrades or changes. Integration may happen asynchronously in constituent systems but networked facilities for integration are now commonly utilized for SoS IVVT&E activities [ODUSD, 2008]. Current research concentrates on methodologies for the SoS capabilities IVVT&E after the constituent systems are upgraded and delivered to integrate with the overall SoS."
- Scalability of the testing methodologies for the SoS
  - ⇒ "The scalability of the testing methodologies for the SoS is a major concern, in particular when large numbers of systems are involved and IVVT&E may be too costly or time-consuming to implement within a limited period of time."
- Uncertainty
  - ⇒ "Macias (2008) acknowledged that T&E must be prepared to handle both certain and uncertain test requirements and would require new tools and methods to address SoS in action-based environments for uncertain operating scenarios."
  - ⇒ "Uncertainty is ubiquitous in all the stages of the SoS development and life cycle if any emergence is allowed in known scenarios and/or if the SoS is operating in unknown scenarios. But, design and development in uncertain/unknown/unexpected operational environments present tremendous

challenges since information about uncertainty will never be complete or accurate.”

❖ Recommendations

⇒ As an example, the authors apply the recommended framework below to UAS

- Systems of Systems (SoS)
  - ◊ Architectural Frameworks
    - ⇒ “The evolutionary IVVT&E framework for SoS would require adapting current framework to enable early identification of possible evolution and knowledge emergence during system design, to allow space for improvement on the requirements to respond to evolution and test for uncertainty. Formal IVVT&E models benefit system design in two ways: (1) Concepts and algorithms can facilitate the communication of ideas in a methodical way, and (2) Formal methodologies allow system developers to analyze properties of a design by building design logic into requirements and using formal models for synthesis of architecture-level representations [Selberg and Austin, 2008].”
  - ◊ Department of Defense Architecture Framework (DoDAF)
    - ⇒ “DoDAF is an architecture framework based on well-understood requirements, ensuring that all requirements are taken into consideration, and making the requirements traceable. This implies that all system variables are well known and with the expected system behavior. However, modern SoS may generate new knowledge or behave in unexpected ways (knowledge emergence, behavior emergence) that renders the SoS unpredictable for untested scenarios and thus, there is a need for IVVT&E methodologies [that deal] with knowledge and behavior emergence.”
  - ◊ Enablers, Controllers, and Inputs/Outputs
    - ⇒ “Enablers allow the realization of required SoS processes. Controllers assist in the appropriate process identification to ensure that all required processes perform their capabilities. Inputs/outputs monitoring keep track of the SoS capability and helps in understanding the system behavior, both in the constituent system and the SoS as a whole.”
  - ◊ SoS interfaces
    - ⇒ “Architecting the SoS around a set of standards will enable constituent systems to be replaced with minimal rearchitecting of system interfaces”
    - ⇒ “The presence of interface layers in the architectural framework greatly assists in facilitating the integration of newly evolved systems or modeling the effects of knowledge emergence and operational uncertainties within existing SoS. Any SoS can be decomposed into architectural layers and can assist in identifying the different interface layers required for facilitating the test and evaluation of SoS.”
- Design Structure Matrix (DSM)

  - ⇒ "The DSM implementation re-organizes iteration loops based on sequencing and clustering of the processes. This improved method shows the impact of the reduced time for IVVT&E activities with early detection of design failures [Levardy and Browning, 2005]."

- Network Graphs
  - ⇒ "Graph theory allows the pictorial representation of the minimum links necessary to test the SoS. … This methodology will provide the optimized testing path through a determined region."
- Testing Strategies
  - ⇒ "Addressing [uncertainty] challenges requires an efficient testing strategy, where once the formal model has been developed, an optimal set of combinatorial test scenarios is designed for the formal model to be evaluated [Zapata et al., 2013]."
  - ◊ Combinatorial/Pair-wise Testing
    - ⇒ "Pair-wise testing is a combinatorial technique that has an exponential nature for building test suites."

Macias, F. (2008). The test and evaluation of unmanned and autonomous systems. *ITEA Journal*, *29*, 388-395.

> "Current Department of Defense test and evaluation capabilities and methodologies are insufficient to address testing of weapon systems operating in non-deterministic and unscripted modes characteristic of the unmanned and autonomous system. Task complexity and adaptability to the environment are critical for evaluation of unmanned and autonomous performance. This represents a new challenge for the test and evaluation community. Verification of system performance and interactions will require the tester to understand the nuances of multiple technical domains…"
>
> "Current T&E techniques are suitable for systems with tightly coupled tethered operations. As we approach infinitesimally close to fully autonomous systems over a 30-year horizon, testing becomes enormously more difficult. Therefore, to address these and other testing limitations, the Test Resource Management Center established the Unmanned and Autonomous System Test (UAST) focus group…"

- ❖ Challenges
  - Complexity
    - ⇒ "In the ordered world of T&E, we have well developed guidance for simple and complicated systems but not for complex systems. As UASs trend to more complex and chaotic dynamics we must relax our strategies to enable us to establish emergent practices for complex systems and novel practices for chaotic systems deployment."
  - Speed of testing turn-around
    - ⇒ "Traditional T&E is limited to single system focus with life cycle development numbered in years."
  - Cost/Overhead associated with testing
    - ⇒ **"**Test, as it is practiced today, has huge overhead and is highly optimized for yesterday's problems."

- ❖ Recommendations
  - Development of a UAST framework that supports
    - ◊ Early tester participation,
    - ◊ Multi-level assessment
      - ⇒ "Monitoring, assessment, and response occur at multiple levels."
    - ◊ Plan-based assessment
      - ⇒ "Monitoring is triggered by an assessment of dependencies and constraints on plan execution."
    - ◊ Capability-based assessment
      - ⇒ "Ongoing assessment of vehicle mission-related capabilities is based on subsystem and environment status."

- ◊ Predictive assessment
  - ⇒ “Monitoring and assessment anticipate future events or conditions.”
- ◊ Team-based assessment
  - ⇒ “Assessment occurs not just of individual vehicles, but at the team level as well.”

McLean, A., Bertram, J., Holke, J., Rediger, S., & Skarphol, J. (2018). LVC enabled testbed for autonomous system testing. *Rockwell Collins & Advanced Technology Center, white paper*.

This white paper identifies the challenge of safely testing autonomous systems in light of emerging regulation, lack of control/safety in live testing, and increasing complex systems with ever evolving hardware and software requirements and capabilities. The authors recommend the use of simulated LVC environments to help make the jump from early bench testing to acquiring safety releases for live test.

- ❖ Challenges
  - “Regulatory Restrictions”
    - ⇒ “Depending on the nature of the autonomous operation and the particular aircraft, it may be simply impossible to obtain permission to perform the flight from the regulatory body, or the process to obtain the approval may approach or exceed the duration of the project.”
  - “Flight Test Complexity”
    - ⇒ “Although remote vehicle operators, through backup control systems, may be able to mitigate certain risks during test flights, the lack of continuous human-in-the-loop control adds new wrinkles to an already complex test environment.”
    - ⇒ “Exceedingly long hours put in by engineering to deal with last minute issues may result in significant budget overruns.”
  - “Computational Resources”
    - ⇒ “…hardware target will change through the development and integration phases”
    - ⇒ “It may be months before this hardware is completed, pushing risk into the integration and flight test phases”
    - ⇒ "Any T&E approach for autonomous systems must be tolerant of late arrival of hardware without completely disrupting the development and integration cycle”
  - “Lack of Human-in-the-loop”
    - ⇒ “In autonomous systems, the human may not have the needed situational awareness to assume control quickly enough, making it much harder to safely recover the vehicle should the autonomous behavior not function correctly.”
  - “Iterative Development”

⇒ "Traditional processes bring a set of integrated capabilities through test and evaluation together. The natural outcome of this approach is that modifications to the system, except to correct deficiencies, is discouraged. For autonomous systems, innovation is occurring at a rate that precludes a waterfall T&E approach. A more appropriate T&E approach for autonomous systems would allow iterative integration as innovations occur, and would avoid the overhead typically imposed by the need for an exhaustive set of regression tests."

❖ Recommendations

- "Use LVC as a risk-reducing transition element from early dev to OT"
    - ⇒ "Autonomous aircraft development demands a testbed capable of providing a smooth transition through two critical phases: (1) moving from the development environment to the integration environment, and (2) moving from the integration environment to the operational test or demonstration platform. The testbed environment we developed is a single integrated set of reconfigurable LVC-enabled resources that host the entire spectrum of testing, from development and integration tests through human-factors evaluations and final operational flight tests."
    - ⇒ "This can be extended to larger "systems of systems" by using a mixture of live and simulation pieces."

Menzies, T., & Pecheur, C. (2004, July 12). Verification and Validation and Artificial Intelligence. Preprint submitted to Elsevier Science.

The authors describe their experience with the NASA Remote Agent Experiment (RAX) system that autonomously controlled the Deep Space One Probe while it was 60M miles from earth, over a two-day period. On that basis they concluded that AI systems can be:

- ❖ Challenges
  - • Highly complex software systems
    - ◊ Traditional testing
      - ⇒ This is inadequate for AS's because of the large number of "situations" encountered (state space explosion)
    - ◊ Run-time monitoring
      - ⇒ Using much interstitial code to detect "error" conditions (out of bounds,…) and provide a recovery mechanism.
      - ⇒ But it requires anticipating a myriad of potential error situations
    - ◊ Static analysis
      - ⇒ Checking out the source code statically without actually executing it. Great for finding basic coding errors
      - ⇒ But clearly suffers from the large problem space imposed by varying environmental variables outside of the agent
    - ◊ Model checking
      - ⇒ Provides checking by exhaustively exploring all reachable states of the software, or model of the software. They have used it on parts of the RAX system
      - ⇒ While running model checking software may be fast, building the model can be time-consuming (and introduce its own errors)
      - ⇒ But it "requires that this state space be finite and tractable", not feasible with a state space explosion imposed by the environment
    - ◊ Theorem proving
  - • [Challenges] of a declarative and knowledge based nature
      - ⇒ They contrast procedural approaches (e.g., a C program, which specifies an order to code actions) with declarative or rule based approaches (e.g., an expert system which specifies relations and if-then rules)
      - ⇒ There are additional tools for procedural systems V&V
  - • [Challenges] of non-deterministic and adaptive nature
    - ◊ External non-determinism
      - ⇒ Again, this is due to uncertainties in the situations that may be imposed by the external world of the agent (to which it is reacting and upon which it acts)

- ◊ Internal non-determinism:
  - ⇒ Here the system may be making random choices in certain situations; or there may be changes in internal behaviors to timing issues due to concurrency issues (multiple threads racing, for instance); or due to system adaptation or learning over time

Micskei, Z., Szatmari, Z., Olah, J., & Majzik, I. (2012). A concept for testing robustness and safety of the context-aware behaviour of autonomous systems. *KES-AMSTA 2012,* 504-513.

Micskei et al. (2012) notes that due to the complexity of even relatively unsophisticated autonomous tasks, formally defining requirements and selecting test cases related to those requirements will be daunting. They argue that practical solutions to the search space demand that testing be at least partially automated. Their recommendation can be summarized as formally defining critical environmental inputs and features, then selecting techniques that allow scenarios to be automatically generated and evaluated based on these features, and ensuring that the system's representations and processing of these can be captured during test (typically simulated).

❖ Challenges

- Even simple tasks are complex
  - ⇒ "Even if the task of a robot is relatively simple, e.g., to pick up garbage from the ground, it should be able to differentiate and recognize numerous types of objects and be prepared to take into account the unexpected movements of humans. Thus, it shall be robust in order to be capable of handling unforeseen situations and safe to avoid harmful effects with respect to humans."
  - ⇒ "First, the behaviour is *highly context-aware*: the actual behaviour of an AS depends not only on the events it receives, but also on the perceived state of the environment (that is typically stored as a context model in the AS). Second, the context is complex and there are a *large number of possible situations*: in real physical world the number and types of potential context objects, attributes and interactions that need to be specified can be large. Third, *adaptation to evolving context* is required: as most of the autonomous systems contain some kind of learning and reasoning capabilities, their behaviour can change in time based on feedback from the evolving environment."
  - ⇒ *"Lack of easy-to-use mechanisms to express and formalize context-aware behaviour* (although these mechanisms are a prerequisite of requirements-based automated test generation and test evaluation). Most noticeably, in existing standard test description languages there is no support to express changes in the context."
- Requirements Specification
  - ⇒ "These characteristics have also consequences on the specification of the requirements to be tested. Full behaviour specification can be impractical due to the complexity of the behaviour and the diversity of the system environments, and requirements should include the evolution of the environment."
- Test Case Selection
  - ⇒ "Ad-hoc testing of stressful conditions and extreme situations: Previous research focused first of all on producing high fidelity simulators or executing

excessive field testing for the verification of AS. There exist methods for testing the physical aspects; however, not all behavioural aspects are well-covered."

- Lack of Metrics
  - ⇒ Lack of precise and objective test coverage metrics that can characterize the thoroughness of the testing process. On the basis of context models we defined precise coverage metrics, especially robustness related metrics that refer to constraints and conditions (to be violated), and combinations of context elements and fragments.

❖ Recommendations

⇒ Figure 1 provides an overview of the automated testing approach described below. Testing scenarios are informed from context and action models. The context model together with the scenario generate test data which are used in execution of the test. A test oracle, described below, is generated from the scenario and the action model and is used to evaluate the test.

- Capture System-level Data During Test
  - ◊ Formally define test data requirements
    - ⇒ "*Test data.* The input to the system under test should be specified. As described previously, in case of autonomous systems this should include (i) the initial state of the context of the system, (ii) its evolution in time, and (iii) the messages and commands received by the SUT. These concepts are captured in a context model. Outputs of the SUT are included in a separate action model."
  - ◊ Automate generation and recording of test data
    - ⇒ "We proposed an automated test data generation approach, which uses search-based techniques. The key ingredients for the application of a search-based technique are the representation of the potential solutions and the definition of a fitness function."
    - ⇒ "First, so called abstract test data are generated and then a postprocessing step produces concrete test data in a format dependent on the simulator. This way the formalized requirements use only general, abstract concepts and relationships (e.g., the robot is near to something). These are replaced in the postprocessing step with compatible types defined in the simulator and the exact parameters (e.g., physical coordinates) are assigned."
  - ◊ Automate evaluation of test cases
    - ⇒ "After test data are generated, the following steps have to be executed. The simulator is fed with each generated test data (which describe the environment of the SUT) and then it processes the dynamic part of the test data (i.e., the evolution of the context, sending and receiving of messages etc.)."
- Create a "test oracle" tool that can automatically code desired outcomes for test events
  - ⇒ "The responsibility of the test oracle is to evaluate the test outcome, the actions and output messages of the system. As specifying the exact outcome of every

situation could be infeasible, a lightweight approach is used. The requirements are expressed as scenarios and are checked for every executed test (to detect potential safety and robustness failures)."

- Have a "context model"—a structured process for describing test data
  - ⇒ "It consists of two parts. The static part represents the environment objects and their attributes in a type hierarchy (in the vacuum cleaner example it includes concepts like room, furniture inside a room, humans or animals). The dynamic part contains events as distinguished elements to represent changes with regard to objects (i.e., an object appears, disappears) and their relations and properties (e.g., a relation is formed or a property is transformed). The events have attributes and specific relations to the static objects depending on the type of the event."
  - ⇒ "Several modelling languages exist to express such models" (pg. 508)

Mueller, S., Hoffman, R., Clancey, W., & Emrey, A. (2019, February). Explanation in Human-AI systems: A literature meta review, synopsis of key ideas and publications, and bibliography for Explainable AI. *DARPA XAI Program Task Area 2, deliverable February 2019.*

This review paper was a deliverable for the DARPA XAI program. It recounts the history and current state-of-the-art techniques in providing explainability for AI-enabled systems. Though it does not directly address T&E, it was included as it will be helpful for discussing gaps in explainability evaluation and for general background as well.

❖ Challenges

- Existing AI/ML benchmarks won't help with XAI
  - ⇒ "It is essential that we as a community respect the time and effort to conduct evaluations." The context of this statement is that the computer science community has developed many benchmarks that can be relatively easily used to determine whether an algorithm is better than another, but since explanations are intended for human, they need behavioral science concepts to evaluate properly.
- Achieving the sufficient level of modeling fidelity
  - ⇒ "Co-adaptation between agents and environments can also give rise to emergent complexity"
  - ⇒ "The problem is if a system … contains a large number of agents … it would be helpful to know … if the system will in fact be able to carry out the required tasks. This would save time modelling the problem, if the agents are discovered to be too simple and also with evaluating the result of any simulation."
- Assigning credit or blame in multi-agent system of systems
  - ⇒ "If something goes wrong with the process, then it might be difficult to identify the source of the problem, is it agent1 and behavior *x* or agent2 and behavior *y*?"

❖ Recommendations

- Ensure evaluators consider the following
  - ⇒ global-versus local explanations
  - ⇒ the need to evaluate the performance of the human-machine work system (and not just the performance of the AI or the performance of the users).
  - ⇒ that the experiment procedures tacitly impose on the user the burden of self-explanation.
- Evaluate explainability with a general model/framework across systems
  - ⇒ This framework can be described by a process and measures.
  - ⇒ The process involves a user receiving an explanation that is generated from the XAI System. The explanation is revised by the User's Mental Model which enables better performance.

⇒ The measures describe the assessment mechanisms at each stage of the process. The explanation is assessed by “goodness” criteria and tests of satisfaction, the user’s mental model is assessed by a test of comprehension, and “better” performance is measured by a test of performance.

Office of the US Air Force Chief Scientist (2011). Technology horizons: A vision for Air Force science and technology 2010-2030. Air University Press, Maxwell AFB, AL.

- Challenges
  - V&V Issues
    - Need for effective V&V that enables “trust in autonomy”.
    - Verifiability and Certifiability
      - Because of the astronomically large state space of these systems, “[development] of such systems is thus inherently unverifiable by today’s methods, and as a result their operation – in all but comparatively trivial applications – is uncertifiable.”
      - Or, equivalently: “It is possible to develop systems having high levels of autonomy, but it is the lack of suitable V&V methods that prevents all but relatively low levels of autonomy from being certified for use.”
    - Highly adaptable, autonomous control systems
      - “Emphasis [should be] on composability via system architectures based on fractionation and redundancy. This involves advancing methods for collaborative control and adaptive autonomous mission planning, as well as V&V of highly adaptable, autonomous control systems.”

Durst, P. J., & Gray, W. (2014). Levels of autonomy and autonomous system performance assessment for intelligent unmanned systems. *U.S. Army Corps of Engineers Engineer Research and Development Center Geotechnical and Structures Laboratory*.

A review of attempts to develop a framework for levels of autonomy, seeing where standards have been formed for unmanned systems (UMSs) and presenting recommendations on where improvements need to be made through the lens of unmanned ground vehicles (UGVs) and ground vehicle T&E. The goal of the authors is to review attempts to quantify autonomy levels (ALs) and attempts at frameworks to move toward a more broadly-acceptable framework for describing levels of autonomy.

❖ Challenges

- Standards for unmanned systems (UMS) have yet to be accepted across the T&E community
  - ⇒ "Several standards have been proposed for UGVs, some of which have been adopted by the international community. These standards relate primarily to UGV software architecture and messaging formats with the goal of enabling interoperability."
- Component-level testing is "lacking"
  - ⇒ Testing sensors, the robotic platform's capability, human/machine interaction, and software are underdeveloped. Some modified legacy tests for components and for UMS algorithms do exist, and mission performance measurement proposed by NIST and the American Society for Testing and Materials (ASTM) have not been widely adopted.
- Even current frameworks do not lend themselves to mathematical rigor
  - ⇒ The Autonomy Levels for Unmanned Systems (ALFUS) capabilities for combining metrics into levels of autonomy are not able to do so mathematically or in a standardized way, nor do they account for mission performance, "too vague and too complex".

❖ Recommendations

- I - Use pre-existing frameworks as a guide
  - ⇒ Current frameworks for interoperability include:
  - ⇒ Joint Architecture for Unmanned Systems (JAUS)
  - ⇒ 4D/RCS reference architecture
  - ⇒ NATO STANAG for Unmanned Aerial Vehicles (UAVs)
  - ⇒ ASTM Committee F41 for Unmanned Maritime Vehicles (UMVs)
- II - Presents current methods of discussing levels of autonomy for consideration
  - ⇒ Autonomy Levels for Unmanned Systems (ALFUS) framework offers model for understanding potential metrics

- ⇒ Using the Contextual Autonomy Capability (CAC), metrics can be developed using the axes of Mission Complexity, Environmental Complexity, and Human Independence, but one level of autonomy cannot be produced.
- ⇒ ALFUS has a summary 0-10 scale of autonomy levels on a model that displays an “autonomy trend” between “Remote control” at 0 and “full intelligent autonomy” at 10.
  - (i) At 0: ““Remote control of UMS wherein the human operator, without benefit of video or other sensory feedback, directly controls the actuators of the UMS on a continuous basis, from a location off the vehicle and via a tethered or radio linked control device using visual line-of-sight cues.” (Huang 2004)”
  - (ii) At 10: ““Completes all assigned missions with highest complexity; understands, adapts to, and maximizes benefit/value/efficiency while minimizing costs/risks on the broadest scope environmental and operational changes; capable of total independence from operator intervention.” (ALFUS Framework 2005)”
- ⇒ Not a method lending itself to standardization, as it changes metric-by-metric and does not “decompose tasks in a commonly agreed-upon, standard way”

- III – Proposes using generic, high-level UAS architectures, based on “context” and “non-context”
  - ◊ One contextual model developed by the US Army Research and Development Center (ERDC) splits UMS architecture into basic layers, “Information acquisition”, “Information analysis”, “Decision and action selection”, “Action implementation”
    - ⇒ Though it is not perfect. “There are, of course, many exceptions that do not fit perfectly within this framework. For most robots, there is not such a clear delineation between each level of the architecture. Often, perception, modeling, planning, and execution all happen simultaneously.”
  - ◊ The NCAP autonomy levels, non-contextual, ranging from 0 to 3
    - ⇒ “A UMS’s AL is defined within the context of the generic architecture model. A UMS that only contains perception, i.e., a teleoperated UGV with an on-board camera, has no autonomy. The UGV simply collects data about its surroundings but does nothing with these data; it has no intelligence. A UGV that generates some sort of world model or retains an internal knowledge base of its surroundings is considered semi-autonomous. At this level, the UGV is interpreting the raw sensor data on its own and has the beginnings of intelligence. A UGV that uses its world model to form a plan of action is considered autonomous.”
    - ⇒ “A UMS’s AL is defined by the architecture level at which a human interacts with the robot.”
    - ⇒ “bench testing of camera, LIDAR, and other sensors, performance testing of SLAM algorithms, or mobility testing of the UMS platform would be

performed, and the results of these component-level tests would then be combined to provide the final, single number autonomous potential."

⇒ Does not provide a mission or environment-specific analysis.

Qiu, S., Liu, Q., Zhou, S., & Wu, C. (2019). Review of artificial intelligence adversarial attack and defense technologies.” Applied Sciences, 9(909). Doi:10.3390/app9050909.

While this paper is not directly related to T&E, it does appear to have an implied relationship with T&E. Artificial intelligence systems are vulnerable to adversarial attacks, which limit the applications of artificial intelligence technologies in key security fields. Issues resulting from adversarial attacks include confidence reduction, misclassification, targeted misclassification, and source/target misclassification. In particular, adversarial attacks can be implemented in the training, testing, and deployment stages, and the resulting issues caused by the attacks manifest in the testing and deployment stages. Therefore, improving the robustness of AI systems against adversarial attacks has played an increasingly important role in the further development of AI. This paper aims to comprehensively summarize the latest research progress on adversarial attack and defense technologies in deep learning.

- Challenges
  - Training Stage Adversarial Attacks
    - Modifying Training Dataset
    - Label Manipulation
    - Input Feature Manipulation
  - Testing Stage Adversarial Attacks
    - White-Box Attacks
      - Adversaries have access to the parameters, algorithms, and structure of the target model and utilize this knowledge to construct adversarial samples to carry out attacks.
    - Black-Box Attacks
      - Adversaries do not have access to information regarding the target model, but they can train a substitute model by querying the target model and utilize the transferability of adversarial samples or by using a model inversion method.
  - Deployment Stage Attacks
      - The authors omit discussion of this class of attacks, as it is similar to the class of attacks occurring in the testing stage.

- Recommendations
  - Modifying Data
    - Adversarial Training
      - Introducing attack samples into training to improve the robustness of predictions.
    - Gradient Hiding
      - Hides information regarding the model gradient
    - Blocking Transferability

    - ⇒ Implemented by perturbing inputs or training on adversarial samples by providing the classifier a null label for such inputs.
  - ◊ Data Compression
    - ⇒ Compression of the input can reduce the effect of a disturbance, but can also reduce the accuracy of the model
  - ◊ Data Randomization
    - ⇒ "random resizing adversarial samples can reduce the effectiveness of adversarial samples. Similarly, adding some random textures to the adversarial samples can also reduce their deception to the network model. [sic.]"
- Modifying Models
  - ◊ Regularization
    - ⇒ Penalize the estimation in order to reduce the model complexity
  - ◊ Defensive Distillation
    - ⇒ "produces a model with a smoother output surface and less sensitivity to disturbance to improve the robustness of the model."
  - ◊ Feature Squeezing
    - ⇒ "reduce the complexity of the data representation, thereby reducing the adversarial interference due to low sensitivity. … Although this technique can effectively prevent adversarial attacks, it also reduces the accuracy of the classification of real samples."
  - ◊ Using a Deep Contractive Network/noise reduction
    - ⇒ "uses noise reduction automatic encoder to reduce the adversarial noise."
  - ◊ Mask Defense
    - ⇒ "insert a mask layer before processing the classified network model. This mask layer trained the original images and corresponding adversarial samples and encoded the differences between these images and the output features of the previous network model layer. It is generally believed that the most important weight in the additional layer corresponds to the most sensitive feature in the network. Therefore, in the final classification, these features are masked by forcing the additional layers with a primary weight of zero. In this way, the deviation of classification results caused by adversarial samples can be shielded."
  - ◊ Use of Parseval Networks/hierarchical regularization
    - ⇒ "Cisse et al. proposed a network called Parseval as a defensive method against adversarial attacks. This network adopts hierarchical regularization by controlling the global Lipschitz constant of the network. Considering the network can be viewed as a combination of functions at each layer, it is possible to have robust positive ions for small input perturbations by maintaining a small Lipschitz constant for these functions, they proposed to control the spectral norm of the network weight matrix by parameterizing the spectral norm of the network weight matrix through Parseval tight frames, so it was called "Parseval" network. [sic.]"

- Using Auxiliary Tools
  - ◊ Defense Generative Adversarial Networks
    - ⇒ Utilizes a generative adversarial network to project input images onto the range of the generator prior to feeding the image to the classifier.
  - ◊ MagNet Detector
    - ⇒ used to identify legal and adversarial sample
  - ◊ High-Level Representation Guided Denoiser
    - ⇒ Used to design a robust target model

Ring, J. (2009). Evolving an autonomous test and evaluation enterprise. *OntoPilot LLC*, 2009.

The presenter argues for an evolving test & evaluation (T&E) enterprise for unmanned aerial systems (UAS) while managing the expectations of stakeholders throughout the T&E process. Graphics and frameworks for observing systemic issues in the T&E enterprise are presented, how an intelligent system fits into the decision-making apparatus of a battlefield, and how the presenter has identified transfers of knowledge and sources of capabilities in a T&E enterprise. The presenter then proceeds to explain measures of effectiveness of a T&E enterprise, a frameworks for understanding parts of a UAS system, and how to best initiate a T&E enterprise to allow organic emergence of a structure to promote better stakeholder service and synergy.

- ❖ Challenges
  - Managing stakeholder expectations
    - ⇒ The T&E of UAS is high in complexity with ambiguous and a great variety of challenges, and stakeholders are not always in the proper place in the loop to have full knowledge of the process. There are many factors to consider in the highly complex picture of a battlefield where intelligent systems fit in.
  - Conceptualizing a framework to understand the components of a UAS
    - ⇒ Modeling of processes can be ambiguous, creating problems for development and communication to stakeholders
  - Creating the right conditions for a T&E enterprise to meet all expectations
    - ⇒ DoD processes are decentralized and non-standard

- ❖ Recommendations
  - I – Allow the enterprise structure to emerge organically
    - ⇒ A variety of scenarios can emerge in the development of systems, so the structure that fits the enterprise the most should be adapted to the system that is being developed
    - ⇒ Enterprise capabilities should be ensured through “Situation Assessments”, resource “Co-Alignment”, “Staff Capability Development”, and making sure others are in place
  - II – Develop a common, understandable systems framework

        - ⇒ Use standard labels to show problem spaces and the systems, actions, and dependencies that exist within them
        - ⇒ Approach problems through a model-based program for the system and problem space, as well as an enterprise architecture to show potential structural choices
    - III – Have metrics to evaluate enterprise success
        - ⇒ Be aware of enterprise measures, such as Productivity, Innovation, Cost of Quality, Learning Curve, Work Climate Surveys, and Benchmarking, as well as personnel measures such as ambiguity, Apathy, and Rumors
    - IV – Ensure the proper motivating factors to ensure the enterprise emerges as well as possible
        - ⇒ Consider factors of motivation and potential scenarios for the development of an enterprise

Roske, V., Kohlberg, I., & Wagner, R. (2012, March 15). Autonomous systems challenges to test and evaluation. *National Defense Industrial Association, Test and Evaluation Conference*, 12-15 March 2012.

This NDIA presentation looks at AS Testability, and identifies characteristics/metrics/standards. The authors recommend that testing be fractionated, testing on perception, reasoning, and execution (see/think/do).

- ❖ Challenges
    - Challenges for System Designers (and T&E):
        - ◊ Establishing which Characteristics to observe
            - ⇒ "Environmental characteristics germane to the system's objectives
            - ⇒ Includes characteristics of objectives, of threats, of location, of neutrals, of the of the system itself, of many other germane entities"
        - ◊ Establishing Metrics for each characteristic
            - ⇒ "What essentially describes (measures) the characteristics?
            - ⇒ Tilt or height of a wall, GPS coordinates, motion of a human
        - ◊ Establishing Standards for the Metrics
            - ⇒ How "collapsed" (short or leaning) does a wall need to be to be "destroyed"
            - ⇒ To stimulate action (coordinates of "here" VS of the "destination")
            - ⇒ To know when to STOP or not take action
    - T&E Challenges for perception function
        - ◊ Need to separate out evaluating sensor performance from evaluating system inferences from those sensors

    - **<u>T&E Challenges for reasoning function</u>**

- ◊ **Informing a confidence in an algorithm's decision making performance**
- ◊ **Demands (professional/ moral/ legal) for ensuring adequate T&E to avoid unacceptable consequences from system behavior**
  - ⇒ Establishing Certifications for Autonomous System T&E methods and practitioners
  - ⇒ T&E of Decision Making Algorithms in a system context

❖ Recommendations

- To ensure Testability: (what to measure to establish performance)
  - ◊ Requires a new System Design discipline and an early collaboration with T&E
    - ⇒ Establishing System Boundaries between Perception, Decision Making and Execution Functions
    - ⇒ Incorporating decision algorithm performance in system control design
    - ⇒ Producing Characteristics, Metrics and Standards for effective decision making
- To ensure adequate Testing (to inform confidence in the measured performance)
    - ⇒ Requires a new, scientifically rigorous foundation for planning T&E programs for autonomous systems, Merging: Control Theory, Complexity Science, Design of Experiments

Scheidt, D. (2017). “NAVAIR Autonomy TEVV Study.” Weather Gage Technologies, LLC.

As NAVAIR seeks to develop an in-house autonomy test, evaluation, verification and validation (ATEVV) capability, it needs to develop its own relevant ATEVV policies and practices, tool suites, and skills and expertise. This presentation presents an 8-step process for ATEVV that is intended to mitigate the verification challenges posed by multiple algorithms and subcomponents in an autonomous system of systems. A detailed breakdown of the process is presented below in “Recommendations.”

❖ Challenges

- Complexity in systems of systems
    - ◊ Multiple algorithms within autonomous systems carrying out different functions
        - ⇒ “Independent validation of these algorithms is insufficient. An assurance argument that guarantees that the properties that emerge from the interactions between the algorithms are satisfactory is necessary.”
- It’s unclear how different elements of testing and verification relate to each other
    - ◊ How do various tests of different components sum to satisfactory verification/proof of safety case?
        - ⇒ “How do these relate to each other? The answer is that we don’t really know, and we’re not really likely to know until somebody (NAVAIR) attempts an end-to-end ATEVV.”
- Future unknowns
    - ◊ Policy and legal ethical questions
        - ⇒ What are the limits of ATEVV as we now understand them? What are the ethical and legal ramifications associated with our ability to understand and assure the performance of autonomous systems?
        - ⇒ What should a policy that appropriately divides the authority and responsibility for autonomous unmanned vehicle operations between: Operator; Commander; Peers (i.e., combatants w/o control over UAS); Acquisition Community; Test Community; Development Contractor?

❖ Recommendations

- Create an 8 step process to develop ATEVV process for NAVAIR
    - ◊ 1. Scope the problem
        - ⇒ “What is the “autonomous” about the system? What kinds of decisions are made by the autonomous system: perception, localization, path planning, learning of target classification, learning target behaviors, learned control policies and so on

⇒ “What is the operating environment like? Static or Dynamic? Simple or complex? Includes other decision-makers, are they friendly, neutral or adversarial?

⇒ “What are the possible interactions between the autonomy and the outside world?

⇒ “Anticipated insight and product –Identifying gaps in terms, ontologies and languages that are required to define operational domain and conditions.”

◊ 2. Define the requirements

⇒ “The first product of the autonomy requirements process is a detailed specification that defines requirements for the entire system, which include the physical plant as well as the autonomous decision-making apparatus.”

◊ 3. Define the metrics, units of measure

⇒ “The second product of the autonomy requirements process is the development of the cognitive specifications which separates out the requirements that are specific to the decision-making process.

◊ 4. Cross matrix the scope and the requirements

⇒ “The third product of an autonomous system requirements analysis is the Adaptability Matrix, which identifies change that must be managed by the autonomous system.

⇒ “The fourth product of the autonomy requirements process is the Cognitive Decomposition which breaks down decision-making requirements into sub-requirements for each component within the cognitive architecture.

⇒ The final product of the requirements process is cognitive process model that illustrates the relationships between cognitive components and a dependency matrix that enumerates the required quality of cognitive inputs and the produced quality of cognitive outputs. By explicitly enumerating cognitive component dependencies we may produce assurance traceability from disparate sources.

◊ 5. Identify and access tools and methods

⇒ “For each tool/method identify – Scope of the tool, what decisions it can be used to test, class of vehicle; Assumptions/inputs required by the tool; Products/outputs by the tool and the assurance/invariants represented by those products”

◊ 6. Define the assurance process

⇒ “After defining the requirements in detail, and identifying suitable ATEVV tools, an Assurance Plan defines the assurance methods that will be used to engender trust of each cognitive component.”

◊ 7. Map existing tools to the process and identify gaps

⇒ “Select existing tools and techniques for use in ATEVV for our target system. Define a subset of requirements we expect to validate (note that a full validation is likely to be prohibitively expensive).

- ⇒ “From the sets of tools identified produce a 4D mapping of Scope x Requirements x Cognitive Element x Tool/method in which each cell is (roughly) formulated as a Hoare triplet;
- ⇒ “Preconditions method Postcondition
- ⇒ “This provides us with a formalism that can be used to combine TEVV product results into an assurance argument. Note that it is more important to create and end-to-end chain that spans the process than it is to comprehensively address all requirements.”

◊ 8. Go experimenting

- ⇒ “Conduct a deep experimentation program in which all phases of the ATEVV process are used to produce a limited assurance argument. Did the tools produce the expected products? If not, why? Were the products sufficient to support the argument envisioned? If not, why?”

Schultz, A., Grefenstette, J., & De Jong, K. (1993). *Test and evaluation by genetic algorithms*. Genetic Algorithms, 9-13.

The authors used a genetic algorithm to test the autonomous system simulation. The AS in this paper was a controller for a flight simulator. The goal of the genetic algorithm was to find as many fault scenarios as possible altering the combination of initial conditions. They discuss ideas of how to evaluate the fault scenarios found. One method is to measure the difference between the controllers actual performance in a scenario against a perfect response in the same scenario. The second approach is to measure the fitness based on the likelihood and severity of the fault conditions. The third approach is that the optimization function of the genetic algorithm is highly rewarded for fault scenarios on boundaries of the controller’s performance space. The fourth is a very fuzzy 'interesting scenarios' evaluation.

- ❖ Challenges
  - Traditional V&V of specifications is insufficient
    - ⇒ “Validation and verification are not enough: the controlled might perform as specified, but the specifications could be incorrect.”
    - ⇒ “traditional controller tests are labor intensive and time consuming.”

- ❖ Recommendations
  - Use machine learning to automate the process
    - ⇒ “…subject a controller to an adaptively chosen set of fault scenarios in a vehicle simulator, and then use a genetic algorithm to search for fault combinations that produce noteworthy actions in the controller.”
    - ⇒ “…we must explicitly define an evaluation function that can measure the fitness of each scenario. This can be difficult because evaluation criteria are often based on informal judgments.”
    - ⇒ “To search for fault scenarios, we use a class of learning systems called *genetic algorithms”*

Streilein, W., Thornton, J., Malyksa, N., Bernays, J., Mersereau, B., Roeser, C., Mohindra, S., Shah, D., & Zipkin, J. (2019, July 15). *Future NMI Study: Counter-AI*. Massachusetts Institute of Technology Lincoln Laboratory.

Machine learning AI capabilities are vulnerable to adversarial counter-AI attacks. These attacks can have a substantial effect on the performance of AI systems. These attacks may not need significant access to the AI model, as attacks can be developed on proxy models and deployed against AI systems. ML systems need to be made robust against adversarial attacks during development. Testing AI systems against a counter-AI red team would develop this robustness. Having an AI red team would help to mitigate these challenges. It would be a centralized location for subject-matter experts. The red team would work both on evaluating AI systems in development as well as supporting their own red team infrastructure. An important element of this infrastructure will be the testbed framework for evaluating AI systems. Integrating this AI red team into the AI T&E lifecycle would support the development of robust AI systems hardened to adversarial attack.

- Challenges
  - AI capabilities provide new and unintuitive vulnerability to attack
    - Computer vision works very differently from humans and can be attacked unintuitively
      - “For example, applying simple digital transformations to input images can drastically alter the resultant output of AI unequipped to handle deviations from the expected input.
    - Poisoning attacks target models training off of observations
      - “This type of attack is common in systems that must rely on observations in the operational domain for its training data.”
    - Evasion attacks push classifier algorithms into an erroneous class
      - “… evasion attack is the most common adversarial AI use case.”
      - “The most threatening evasion attacks utilize adversarial inputs that are imperceptible to casual human observers and resilient to pre-algorithm data conditioning.”
    - Model inversion attacks can extract private/sensitive information from the algorithm
      - “Training data privacy is especially important when the data involved are sensitive or strictly regulated…”
  - Adversarial attacks are not especially challenging
    - Packages/toolkits exist for adversarial attacks
      - “Numerous toolkits exist to produce adversarial input, and the brittleness and low explainability of most AI algorithms leads to regrets when modifying implementations for security purposes.”

- ◊ Securing algorithms can be detrimental to performance
  - ⇒ “Training-based approaches … have the clear disadvantage of reducing response accuracy.”

❖ Recommendations

- Defenses have been developed against adversarial attack
  - ◊ Algorithmic and non-algorithmic approaches are available
    - ⇒ “… measure prediction consistency across multiple trained classifiers … input transformations or feature squeezing to reduce input perturbations.”
  - ◊ Use of multiple defense approaches may be needed
    - ⇒ “The strongest defenses will use a combination of algorithmic and system-level techniques integrated throughout the AI-supported system to detect and repel adversarial attack.”
- Vulnerability of AI systems to adversarial countermeasures can be quantified through direct testing and experimentation
  - ◊ Use of metrics to quantify impact of adversarial attack and test over multiple datasets
    - ⇒ “… it would be helpful to assess the impact of different attack scenarios by running experiments over entire data sets and deriving quantitative metrics of effectiveness.“
- Use of red team group for countering adversarial attacks on AI systems
  - ◊ Use of AI red teaming can evaluate systems under development
    - ⇒ “… the Red Team assessments will provide a critical evaluation of systems under development…”
    - ⇒ “… leading to systems that are pre-hardened to be robust at deployment.”
  - ◊ Red team can be subject-matter experts
    - ⇒ “This component of the team will be able to stay abreast of advances in the AI community in order to ensure the work on the team is relevant.
  - ◊ Red team used for both evaluation and red team infrastructure support
    - ⇒ “… those that occur on an on-demand basis as developed AI capabilities are evaluated … and activities that support persistent Red Team infrastructure development.”
    - ⇒ “[Evaluation] will largely take place using software and rely upon blue team-provided code and mission data.”
    - ⇒ “… it will be essential that the Red Team monitors ongoing developments in both AI development and counter-AI capabilities”
  - ◊ Heterogeneous team of government personnel and FFRDC/UARC
    - ⇒ “… to use a combination of government personnel to guide the Red Team activities, and FFRDC or UARC support to supply the technical expertise”
- Assembly and use of testbed framework

⇒ “… one of the most important near-term actions is assembling the first version of the testbed infrastructure, … the AI Countermeasures Evaluation framework”

◊ Several options for physical infrastructure

⇒ “… including commercial cloud infrastructure, a managed configuration of cluster compute assets, or integration with the larger JAIC common infrastructure.”

⇒ “Note that many of these components will not have to be developed from scratch, but can make use of existing open source libraries and ongoing DoD and IC-sponsored efforts”

Rubbu, R., Visnevski, N., & Djang, P. (2009). Evolutionary framework for test of autonomous systems. *Association for Computing Machinery PerMIS'09*, September 21-23, 2009, Gaithersburg, MD. p. 93-98.

> "Autonomous systems of the future will need to be tested so their mission capabilities and robustness are predictable to the warfighter. The principal challenge therefore is the set of test strategies for these future autonomous systems. The goal of the test community is that these autonomous systems be broadly accepted to seamlessly operate either independently or as part of a human-in-the-loop system. [The authors'] goal is to develop an efficient intelligent test process that will enable the rapid introduction of autonomous systems on the battlefield. [They] propose a novel war game simulation-based multi-objective evolutionary test framework that combines the elements of testing an autonomous system's mission execution capabilities as a function of its innate capabilities and evolutionary computation."

- ❖ Challenges
  - The set of scalable test strategies for these future autonomous systems
    - ⇒ "Current DoD test and evaluation capabilities and methodologies while sufficient for tightly tethered human-in-the-loop systems are insufficient for the mission certification of complex autonomous systems operating in non-deterministic environments. Autonomous systems of the future will need to be tested so their mission capabilities, robustness, and failure modes are predictable to the warfighter. The principal challenge therefore is the set of scalable test strategies for these future autonomous systems."

- ❖ Recommendations
  - Evolutionary algorithm based simulation testing framework for failure identification
    - ⇒ "We propose a novel war game simulation-based test framework that utilizes evolutionary algorithms for identifying the mission failure modes. While the traditional application of evolutionary methods is for efficient synthesis or design, we propose the use of these methods for the efficient identification of failure scenarios from a mission satisfaction perspective."

Alan, S. (2015, May). *Technology Investment Strategy 2015-2018*. DOD R&E Autonomy Community of Interest (COI), T&E V&V (TEVV) Working Group.

- ❖ Challenges
  - Issues caused by Autonomous Systems themselves
    - ◊ State-Space Explosion
      - ⇒ "Autonomous systems are characteristically adaptive, intelligent, and/or may incorporate learning. For this reason, the algorithmic decision space is either non-deterministic, i.e., the output cannot be predicted due to multiple possible outcomes for each input, or is intractably complex. Because of its size, this space cannot be exhaustively searched, examined, or tested; it grows exponentially as all known conditions, factors, and interactions expand. Therefore there are currently no established metrics to determine various aspects of success or comparison to a baseline state enumerated."
    - ◊ Unpredictable Environments
      - ⇒ "The power of autonomous agents is the ability to perform in unknown, untested environmental conditions. Examples of environmental "stimuli" include actors capable of making their own decisions in response to autonomous system actions; producing a cognitive feedback loop that explodes the state space. Additionally, autonomous decisions are not necessarily memoryless and the state space is not just the intractably complex in the current situation, but also in the multiplicity of situations over time. Currently fielded systems have very limited robustness to dynamic / changing environmental conditions. Adaptive autonomous algorithms have the potential to overcome current automated system brittleness in future dynamic, complex, and/or contested environments. However, this performance increase comes with the price of assuring correct behavior in a countless number of environmental conditions. This exacerbates the state-space explosion problem."
    - ◊ Emergent Behavior
      - ⇒ "Interactions between systems and system factors may induce unintended consequences. With complex, adaptive systems, how can all interactions between systems are captured sufficiently to understand all intended and unintended consequences? How can autonomous design approaches identify or constrain potentially harmful emergent behavior both at design time and at run time? What limitations are there with the current Design of Experiments approach to test vector generation when considering adaptive decision-making in both discrete decision logic and continuous variables in an unpredictable environment? Since emergent behavior can be produced by interactions between small, seemingly insignificant factors how can we provide test environments or test oracles that are of sufficient fidelity to examine and

capture emergent behavior (in M&S, T&E, and continuous operations or run time testing)?

- ◊ Human-Machine Communication
  - ⇒ "Handoff, communication, and interplay between operator and autonomy become a critical component to the trust and effectiveness of an autonomous system. Current certification processes eliminate the need for "trust" through exhaustive Modeling and Simulation (M&S) and T&E to exercise all possible operational vignettes. When this is not possible at design time, how can trust in the system be ensured, what factors need to be addressed, and how can transparency and human-machine system requirements for the autonomy be defined?"

- Capability Gaps in AS V&V
  - ◊ Lack of Verifiable AS Requirements
    - ⇒ Currently, there is a lack of common, clear, and consistent requirements for systems that include autonomous requirements, especially with respect to environmental assumptions, Concept of Operation (CONOPS), interoperability, and communication. There is also a lack of clearly defined Measures of Effectiveness (MOEs), performance measures, and other metrics.
    - ⇒ Furthermore, there are deficiencies in ensuring the traceability of requirements to implementation (e.g., manufacturing or compiling) both manually and automatically. Automatic requirements extraction and validation is needed for future learning / adaptive systems. Finally, current autonomous systems requirements are not written or analyzed to ensure that they are verifiable.
  - ◊ Lack of Modeling, Design, and Interface Standards
    - ⇒ Currently, no standardized modeling frameworks exist for autonomous systems that span the whole system lifecycle (R&D through T&E). Therefore, a gap exists in traceability between capabilities implemented in conventional systems as well as adaptive, nonlinear, stochastic, and/or learning systems and the requirements they are designed to meet. This results in a need to integrate models that are both heterogeneous and composable in nature and existing at different levels of abstraction, including requirements, architecture models, physics-based models, cognitive models, test range/environment models, etc.
  - ◊ Lack of Autonomous Test and Evaluation Capabilities
    - ⇒ There is a current gap in T&E ranges, test beds, and skillsets for handling dynamic learning / adaptive systems. The complexity of these systems results in an inability to test under all known conditions, difficulties in objectively measuring risk, and an ever-increasing cost of rework / redesign due to errors found in late developmental and operational testing. Furthermore, the lack of formalized requirements and system models makes test-based instrumentation of model abstractions increasingly difficult. This limits design-for-test capabilities, including tests to evaluate human-autonomy interactions.
  - ◊ Lack of Human Operator Reliance to Compensate for Brittleness

- ⇒ Currently, the burden of decision making under uncertainty is placed solely on human operators. Certification, acceptance, and risk mitigation often assume the human operator can compensate for the brittleness currently found in manned, remotely piloted, or tele-operated systems. However, as systems move from relatively predictable automated behaviors to more unpredictable and complex autonomous behaviors, and as autonomous systems operate in denied environments in which they interact with human intermittently, it will become increasingly difficult for human operators to understand and respond appropriately to decisions made by the system. Thus, V&V of autonomous software should also take into account factors relating to human-machine interfaces, human performance characteristics, requirements for human operator training, etc.

- ◊ Lack of Run Time V&V during Deployed Autonomy Operations:
  - ⇒ Current automated systems rely on human oversight to guarantee safe and correct operation, with the human operator acting as the ultimate monitor, kill switch, and recovery mechanism for brittle automation. However, as systems incorporate higher levels of autonomy, it will no longer be feasible or safe to rely solely on human operators for system monitoring and recovery. Therefore, “sandboxing,” bounding, or encapsulation of learning / adaptive systems must be developed and supported in the V&V process so that higher levels of autonomy can be deployed in operational environments without full, exhaustive testing.
- ◊ Lack of Evidence Re-use for V&V:
  - ⇒ Results from TEVV do not, by themselves, accept operational risk, imply certification, or give authority to operate. However, TEVV results provide the collected body of evidence that is presented to a certification board, and ultimately the milestone decision authority (MDA), to determine an acceptable level of safety, security, performance, and risk for that specific platform. However, current assurances (arguments of safety and security that a system falls within an acceptable level of risk) are predominately manual, subjective, and often not reusable.

- ❖ Recommendations
  - 1. Methods & Tools Assisting in Requirements Development and Analysis
    - ⇒ Precise, structured standards to automate requirement evaluation for testability, traceability, and de-confliction
    - ⇒ This goal focuses on increasing the fidelity and correctness of autonomous system requirements by developing methods and tools to enable the generation of requirements that are, where possible, mathematically expressible, analyzable, and automatically traceable to different levels (or abstractions) of autonomous system design.

- ⇒ Formalized requirements enable automatic test generation and traceability to low-level designs, but note that TEVV representatives must be involved early in the requirements development process.
- 2. Evidence-Based Design and Implementation
  - ⇒ Assurance of appropriate decisions with traceable evidence at every level of design to reduce the current T&E burden
  - ⇒ Methods and tools need to be developed at every level of design from architecture definition to modeling abstractions to software generation / hardware fabrication, enabling the compositional verification of the progressive design process, thereby increasing test and evaluation efficiency.
  - ⇒ Quoting Tech Horizons: "Emphasis is on composability via system architectures based on fractionation and redundancy. This involves advancing methods for collaborative control and adaptive autonomous mission planning, as well as V&V of highly adaptable, autonomous control systems."
- 3 Cumulative Evidence through RDT&E, DT, & OT
  - ⇒ Progressive sequential modeling, simulation, test and evaluation.
  - ⇒ Methods must be developed to record, aggregate, leverage, and reuse M&S and T&E results throughout the system's engineering lifecycle; from requirements to model-based designs, to live virtual construction experimentation, to open-range testing. **[**We need] standardized data formats and Measures of Performance (MOPs) to encapsulate experimental results performed in early research and development, ultimately reducing the factor space in final operational tests.
  - ⇒ Additionally, statistics-based design of experiments [DOE] methods currently lacks the mathematical constructs capable of designing affordable test matrices for non-deterministic autonomous software. [New methods are needed]
- 4 Run Time Behavior Prediction and Recovery
  - ⇒ Real-time monitoring, just-in-time prediction and mitigation of undesired decisions and behaviors
  - ⇒ For highly complex autonomous systems, pre-fielding testing may not be enough. An alternate method leveraging a run-time architecture must be developed that can provably constrain the system to a set of allowable, predictable, and recoverable behaviors, shifting the analysis/test burden to a simpler, more deterministic run-time assurance mechanism.
- 5 Assurance Arguments for Autonomous Systems
  - ⇒ Reusable assurance case based on previous evidence "building blocks"
  - ⇒ [N]ot only do multiple new TEVV methods need to be employed to enable the fielding of autonomous systems, a new research area needs to be investigated in formally articulating and verifying that the assurance argument itself is valid.
  - ⇒ [A] structured argument-based approach must be developed in coordination with and as an integral part of the Test and Evaluation Plan (TEP) and the Test and Evaluation Master Plan (TEMP), providing a claim of how the V&V

activities will endeavor to quantify risks and mitigation strategies to inform risk-acceptance decisions.

- An engineering framework is presented
  - ⇒ Requirements definition -> requirements analysis -> logical analysis -> design solution -> implementation -> integration -> verification -> validation -> transition
  - ⇒ Requirements definition <-> transition via operational need and measures of effectiveness
  - ⇒ Requirements analysis <-> validation via system and measures of performance
  - ⇒ Logical analysis <-> verification via allocated functions and performance requirements
  - ⇒ Design solution <-> integration via component interface and definition

Visnevski, N. (2008). Embedded instrumentation systems architecture. *Embedded Systems Laboratory*, Niskayuna, NY.

EISA (Embedded Instrumentation Systems Architecture) is a control architecture for integrating embedded test equipment and synthetic (abstract/virtual) instruments into a centralized monitoring unit. Complex systems can involve a great number of measurement sensors and smart systems which may not be well integrated. This can greatly impede users' ability to control and monitor the status of the system overall, where comparisons between different subsystems are required. Many of these subsystems may be legacy systems with little flexibility and little compliance to standards. Use of wrapper functions to build compliance to standards and centralization can greatly improve users' abilities to coordinate and compare different subsystems.

- ❖ Challenges
  - T&E, V&V over entire lifecycle
    - ⇒ "[The DOD's] needs include developmental, operational, and continuous T&E of military weapons and equipment to ensure their operational readiness both at the test ranges and over the entire lifecycle of the assets."
  - Systems may use legacy sensors not compliant with current standards (like IEEE 1451)
    - ⇒ "the job of these nodes is to aggregate data from a variety of embedded legacy of smart sensors…"
    - ⇒ "Legacy software often lacks flexibility to program complex equations …"
  - Aggregate data from a large network of heterogeneous sensors
    - ⇒ "…data aggregation from a large network of heterogeneous sensors in a time synchronized and correlated fashion."
    - ⇒ "There are approximately 175 sensors in [the demonstrative implementation]. The complexity of the data acquisition system is the cause of several challenges for the MEPS testers."
    - ⇒ "… it is difficult to reconstruct event data at the time of the fault from different instruments because the data is not synchronized to a global time stamp."
    - ⇒ "Each instrument requires a separate calibration and configuration process…"
    - ⇒ "Each instrument utilizes its own user authentication and data control process. Developing a comprehensive data security process is difficult."
    - ⇒ "… separate data acquisition systems produced data at different data rates that were not correlated, not time synchronized, were stored in different databases, and posed challenges for post-processing."

- ❖ Recommendations
  - Use of virtual sensors where physical sensors cannot exist
    - ⇒ "[Virtual sensors] are virtual data collection points for which physical sensor (sic) does not exist."
  - Standardization and centralization

◊ Using standardized (e.g. IEEE 1451 compliance) wrappers

⇒ “This enabled continuous and integrated data and metadata aggregation for the entire test system. The test data was automatically time synchronized and stored in a single database, greatly simplifying post-test analysis.”

Visnevski, N. A., & Castillo-Effen, M. (2010). Evolutionary computing for mission-based test and evaluation of unmanned autonomous systems. *2010 IEEE Aerospace Conference,* Big Sky, MT, 1-10.

The authors suggest a dynamic system for the test and evaluation of autonomous systems, in which the test ‘evolves’ along with the algorithm in order to find new exploits and points of failure in a system under test, a “system-test co-evolution”. With so many variables that could change in a real operational environment, the authors suggest viewing these challenges as a Design of Experiments (DoE) problem and recommend “co-evolution” as a principle to be used in modeling and simulation (M&S) in order to test systems with high degrees of complexity. Traditional models of testing do not allow changes in the system being tested to take place, even those which the authors suggest are the best among current methods within stochastic optimization. The authors provide a framework for the components of a co-evolutionary M&S environment and present a case study using a 3D Computer-Generated Forces (CGF) model from MÄK Technologies.

❖ Challenges

- UAS designs may be able to “outsmart” tests that are static, and after repeat tests be able to find ways to exploit them in unintended ways
  - ⇒ “This idea is rooted in cognitive psychology and is based on the fact that human subjects can understand and “outsmart” fixed tests that are presented to them repeatedly. Therefore, a more refined evaluation strategy includes testing infrastructure that recognizes the fact that test subjects can evolve to “outsmart” the test.”
- Testing unmanned systems can present high costs in live environments
  - ⇒ “One of the serious stumbling blocks to development of sophisticated T&E methodologies for UAS is the scarcity and the high cost of test subjects.” (1)
  - ⇒ “This [live] type of simulation is the most demanding in terms of resources, range safety, instrumentation, etc. Although the simulation results may be considered realistic because they involve the actual physical system, the simulated operations are designed with many constraints.”
- Traditional Measures of Effectiveness of a system (MoE) may not match with Measures of Performance (MoP), an example being the Predator UAS
  - ⇒ “Real-life cases have demonstrated that the traditional approach to T&E based on the verification of performance requirements do not work properly for complex systems and that a paradigm shift is necessary. An example that is often cited to support this notion is the Predator MQ-1 UAS, which failed operational T&E but proved extremely useful on the battlefield.”
- Current methods of stochastic testing allow changes in the test environment only

        - ⇒ Though the authors recommend stochastic optimization as the best technique currently to test UAS systems, specifically evolutionary programming and evolutionary multi-objective optimization, they take issue with only allowing the test to evolve and not the system being tested as well, "…the static evolutionary model assumes dealing with fixed assets of systems under test that do not change over the course of the test."

- ❖ Recommendations
    - I - Stochastic optimization works best among current methods
        - ⇒ Stochastic optimization is the best technique currently to test UAS systems, specifically evolutionary programming and evolutionary multi-objective optimization.
        - ⇒ "Among those, evolutionary programming and evolutionary multi-objective optimization seem particularly appropriate from both the technical as well as the intuitive/esthetic standpoints. From the technical standpoint evolutionary computing has been shown to be fairly adaptable to imprecisions and fuzziness in the in-puts and in the problem formulation methods. From an esthetic standpoint one might argue that evolutionary computing approach to the problem of test planning is intuitively elegant in a sense that it supports the paradigm of test plans "evolving", over the course of multiple experiments, to un-veil more and more hidden and non-obvious limitations of systems under test."
    - II - Co-evolving tests in a simulated environment offer greater variability and efficiency
        - ⇒ "Unlike the static evolutionary model, 'system-test co-evolution' model assumes that test involves variability in both the extrinsic and the intrinsic parameters. This means that the systems under test are allowed to evolve with the test."
        - ⇒ "The 'system-test co-evolution' model assumes that test involves variability in both the extrinsic and the intrinsic parameters."
        - ⇒ "…two kinds of independent variables that may be selected in the test synthesis process – the intrinsic and the extrinsic ones. The former have to do with the variability in capabilities of systems under test, and the latter, with the variability in the environment and mission parameters used in the conducted test."
        - ⇒ Can bring down costs of testing UAS in general using modeling and simulation (M&S) by offering a wider variety of scenarios, "Although the simulation results may be considered realistic because they involve the actual physical system, the simulated operations are designed with many constraints."
        - ⇒ Achieves "a minimal set of experiments which yields the maximum information with respect to the hypothesis that need to be tested"
    - III - Use the Mission and Means Framework (M&M)

  - ⇒ May contribute to a "hierarchical relationship between mission effectiveness, tasks, capabilities and system components".
  - ⇒ Uses tasks to be completed as a basis for metrics, i.e., "The RoboCup Virtual Rescue competition represents a good test-case [of M&M], where mission-based capability driven UAST could be applied. Although the scenario is simulated, metrics such as number of victims found within certain time or energy constraints are related directly to measures of effective-ness. The mission is composed of a number of tasks such as: exploring, searching for victims, reporting victims, etc. Similarly, tasks may be performed only if certain capabilities are present."

- IV - Use co-evolving tests to test complex systems in unknown environments and for test planning
  - ⇒ Co-evolving tests can adapt to changes in the system under test and create simulations which can test highly complex systems. "The most complex tests at the 'Systems of Systems' (SoS) level are basically impossible to be executed as live simulations."
  - ⇒ Co-evolving tests "t will allow not only finding challenging test scenarios that uncover system design limitations, but also providing hints to UAS developers on how their systems can be modified to overcome these limitations."
  - ⇒ Further, "it will account for the growing machine-cognitive aspect of UAS. It is our belief that eventually we will see machines that can learn and display sophisticated cognitive capabilities. This may enable them to 'figure out' the test as it is being conducted and 'outsmart' it."
  - ⇒ "To maximize the efficiency of physical tests in terms of time and resources, test planning is crucial. The only way to avoid the curse of dimensionality in designing experiments with a large number of independent variables is by including as much knowledge of the process as possible. This is where modeling and simulation is most valuable. M&S can be seen as the main vehicle for incorporating knowledge about the system."
- V - Develop an M&S tool for "system-test co-evolution"
  - ◊ The recommended tool should have three components:
    - ⇒ Scenario Generator: "accesses libraries of models, which should be verified, validated, and accredited. There are two libraries of such models: one library with models of UAS and models of other entities relevant to the missions being simulated, and another library with models of the environment. Experiments generated by the search engine al-together with models are used to generate so-called scenarios"
    - ⇒ Effectiveness Evaluator: "uses metrics defined by the testers to evaluate the probability of mission success/failure. Since the simulation engine is stochastic, the outcomes for a certain experiment may vary for different iterations."

- ⇒ Search Engine: “the evolutionary computation based search engine may be considered the core of the automated test planner. The search may start from a set of randomized experiments. The main function of the search engine is to generate a new set of experiments using previous experiments and their outcomes.”
- ⇒ “The overall function of the three components is to generate scenarios with increasing difficulty for the systems under test. Hence, only extrinsic independent variables may be manipulated.”

Wegener, J., & Bühler, O. (2004). *Evaluation of Different Fitness Functions for the Evolutionary Testing of an Autonomous Parking System. In: Deb K. (eds).* Genetic and Evolutionary Computation – GECCO 2004. GECCO 2004. Lecture Notes in Computer Science, vol 3103. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-540-24855-2_160

> "The method of evolutionary functional testing allows for the automation of testing by transforming the test case design into an optimization problem. For this aim it is necessary to define a suitable fitness function. In this paper two different fitness functions are compared for the testing of an autonomous parking system."

❖ Challenges

- Manual testing can limit the number and quality of test cases
  - ⇒ "Electronic control units (ECUs) in cars take over more and more complex tasks. … For such applications errors in the ECU's software can result in high costs. Therefore, the aim is to find as many errors as possible by testing the systems before they are released. In practice, dynamic testing is the analytical quality assurance method most commonly used. Usually, a complete test is infeasible, because of the huge number of possible input situations. … In most cases, test case design is performed manually, requiring a considerable part of the project's resources."
- Available tests are not easy to automate
  - ⇒ "One important weakness of the testing methods available is that they cannot be automated straightforwardly. Manual test case design, however, is time-intensive and error-prone. The test quality depends on the performance of the tester."
- Evolutionary Testing requires an appropriate fitness function
  - ⇒ "The method of evolutionary functional testing allows for the automation of testing by transforming the test case design into an optimization problem. For this aim it is necessary to define a suitable fitness function."

❖ Recommendations

- Automation, specifically with evolutionary functional testing
  - ⇒ "The method of evolutionary functional testing facilitates the generation of test cases in order to detect functional errors during a directed search. The method of evolutionary functional testing transforms the test case design process into an optimization problem. Automated test result evaluation is a prerequisite for this process. The evaluation is done by means of the fitness function, which assigns a numerical quality value to a test result."
- Tests should be systematic and extensively automatable
  - ⇒ "In order to increase the effectiveness and efficiency of the test and thus to reduce the overall development and maintenance costs for systems, a test should

be systematic and extensively automatable. Both objectives are addressed by the method of evolutionary testing."

- Implied that the most appropriate fitness function is task/test specific. For the use case of the automated parking system, a fitness function based on an area criterion outperformed a fitness function based on a distance criterion.
  - ⇒ "This paper evaluates two different approaches to the definition of fitness functions for the functional testing of an autonomous parking system. … The results show that of both the criteria proposed in this paper, the area criterion can identify critical parking maneuvers better than the distance criterion introduced in. The area criterion provides a more efficient method of error detection in the parking system."

Zhou., Z., & Sun, L. (2019). Metamorphic testing of driverless cars. *Communications of the ACM, 63(3),* 61-67.

The authors address a major challenge in autonomous systems: providing assurance that bad behaviors will *not* happen. This requires testing large numbers of cases where the correct answer is known. To combat this problem, the authors recommend the use of metamorphic testing with fuzzed inputs. For problematic behaviors in AS, this boils down to looking for inconsistent behavior across what should be functionally identical problems (e.g., selecting a steering direction under different lighting conditions). Rather than testing if it always selects the correct steering angle, which demands knowing that in every situation, tests can cover space more efficiently by looking for where it gives inconsistent answers to fuzzed scenarios—a solution which can be automated.

- ❖ Challenges
  - The Oracle Problem
    - ⇒ "Software testing is, however, fundamentally challenged by the "oracle problem." An oracle is a mechanism testers use to determine whether the outcomes of test-case executions are correct. Most software testing techniques assume an oracle exists. However, this assumption does not always hold when testing complex applications."
  - System Requirements/Specifications
    - ⇒ "…difficulty of creating detailed system specifications against which the autonomous car's behavior can be checked, as it essentially involves recreating the logic of a human driver's decision making."
  - Proving the Null Hypothesis
    - ⇒ "…negative testing serves to ensure the program does not do what it is not supposed to do when the input is unexpected, normally involving random factors or events. Resource constraints and deadline pressures often result in development organizations skipping negative testing, potentially allowing safety and security issues to persist into the released software."
    - ⇒ "To a certain degree, tools called "fuzzers" could help perform this kind of negative software testing. During "fuzzing," or "fuzz testing," the fuzzer generates a random or semi-random input and feeds it into the system under test, hoping to crash the system or cause it to misbehave. However, the oracle problem makes verification of the fuzz test results (outputs for millions of random inputs) extremely difficult, if not impossible."

- ❖ Recommendations
  - Use "Metamorphic Testing" & Fuzzing Together
    - ⇒ "Our testing method: MT in combination with fuzzing."

- ⇒ "Metamorphic testing (MT) is a property- based software-testing technique that can effectively address two fundamental problems in software testing: the oracle problem and the automated test-case-generation problem. The main difference between MT and other testing techniques is that the former does not focus on the verification of each individual output of the software under test and can thus be performed in the absence of an oracle. MT checks the relations among the inputs and outputs of multiple executions of the software. Such relations are called "metamorphic relations" (MRs) and are necessary properties of the intended program's functionality."
- ⇒ "Their MRs required the drone should have consistent behavior, while finding that in some situations the drone behaved inconsistently, revealing multiple software defects. For example, one of the bugs was in the sense-and-avoid algorithm, making the algorithm sensitive to certain numerical values and hence misbehavior under certain conditions, causing the drone to crash."

Appendix B: Challenge & Recommendation Summary Tables

The following tables summarize the challenges and recommendations discussed in this paper.

Table 1: Challenges from System Complexity

Sub-Category	Detailed Challenge	Supporting Articles
Task Complexity	*Defining System Tasks*	Baker et al. (2019), Deonandan et al. (2010), Hernandez-Orallo (2017), Porter et al. (2020), Roske et al. (2012)
	Dynamic or Flexible Tasking	Ahner & Parson (2016), Ahner et al. (2018), Baker et al. (2019), Goerger (2004), Harikumar & Chan (2019), Haugh et al. (2018), Hill & Thompson (2016), Ilachinski (2017), Laverghetta et al. (2019), Lenzi et al. (2010), Luna et al. (2013), Macias (2008), Micskei et al. (2012), USAF Chief Scientist (2011), Zacharias (2019a)
State-Space Explosion	*Cannot Test Exhaustively*	Baker et al. (2019), Ahner & Parson (2016), Deonandan et al. (2010), Defense Science Board (2012), Giampapa (2013), Goerger (2004), Haugh et al. (2018), Ilachinski (2017), Lennon & Davis (2018), Menzies & Pecheur (2004), Micskei et al. (2012), Porter et al. (2018), Porter et al. (2020), Sparrow et al. (2018), Tate & Sparrow (2019), USAF Chief Scientist (2011), Wegener & Buhler (2004), Zacharias (2019b)
Stochasticity	*Environment are Unpredictable*	Ahner & Parson (2016), Defense Science Board (2012), Greer (2013), Ilachinski (2017), Lennon & Davis (2018), Lenzi et al. (2010), Menzies & Pecheur (2004), Micskei et al. (2012), Ring (2009), Subbu et al. (2009)
	System Responses are Non-Deterministic	Ahner & Parson (2016), Goerger (2004), Greer (2013), Hernandez-Orallo (2017), Ilachinski (2017), Laverghetta et al. (2019), Macias (2008), Menzies & Pecheur (2004), Micskei et al. (2012), Subbu et al. (2009)
Learning Enabled Systems	*Evaluating How, Not What, System Learns*	Ahner & Parson (2016), Hernandez-Orallo (2017), Ilachinski (2017)
	Tested System Stops Being Representative	Ahner & Parson (2016), Haugh et al. (2018), Menzies & Pecheur (2004), Micskei et al. (2012), Visnevski & Castillo-Effen (2010)
	Negative Learning	Ahner & Parson (2016), Haugh et al. (2018), Ilachinski (2017)
	Misaligned Testing vs. Learning Timescales	Ahner & Parson (2016)
Training Data	*Data Bias*	Haugh et al. (2018)
	Data Poisoning	Haugh et al. (2018), Qiu et al. (2019), Streilein et al. (2019)
	Operational Representativeness	Haugh et al. (2018)
Multiple Agents or Components	*Emergent Behavior*	Baker et al. (2019), Ahner & Parson (2016), Deonandan et al. (2010), Defense Science Board (2016), Ferreira et al. (2013), Greer (2013), Harikumar & Chan (2019), Haugh et al. (2018), Ilachinski (2017), Lennon & Davis (2018), Lenzi et al. (2010), Luna et al. (2013), Mueller et al. (2019), Scheidt (2017)
	Legacy Interoperability	Deonandan et al. (2010), Eaton et al. (2017), Harikumar & Chan (2019), Visnevski (2008)
Novelty	*Integrating Test Challenges of Cyber & Physical Systems*	Eaton et al. (2017), Harikumar & Chan (2019), Ilachinski (2017), Luna et al. (2013)
	Unknown Unknowns	Deonandan et al. (2010), Harikumar & Chan (2019), Luna et al. (2013), Scheidt (2017)
	Ability to Reuse Evidence	Ahner & Parson (2016), Deonandan et al. (2010), Durst (2019), Giampapa (2013), Lede (2019), Lennon & Davis (2018)
	Coevolved Capabilities	Haugh et al. (2018), Hill & Thompson (2016)
Transparency	*Need to Understand System Decision Making*	Ahner & Parson (2016), Arnold & Scheutz (2018), Giampapa (2013), Gunning (2017), Haugh et al. (2018), Lede (2019), Porter et al. (2020), Subbu et al. (2009), Tate & Sparrow (2020)
	Knowing Ground Truth and/or System Belief	Ahner & Parson (2016), Haugh et al. (2018), Roske et al. (2012), Zhou & Sun (2019)

Table 2: Acquisition System Limitations

Sub-Category	Detailed Challenge	Supporting Articles
Requirements	*Operational Relevance*	Ahner & Parson (2016), Kapinski et al. (2016), Micskei et al. (2012), Durst & Gray (2014), Schultz et al. (1993), Visnevski & Castillo-Effen (2010), Zhou & Sun (2019)
	Unverifiable Requirements	Hess & Valerdi (2010), Lede (2019), Micskei et al. (2012), Durst & Gray (2014), Zhou & Sun (2019)
	Rigidity of Requirements	Ahner & Parson (2016), Deonandan et al. (2010), Lede (2019), Luna et al. (2013), McLean et al. (2018)
	Defining LME Behavior	DoD (2019), Hill & Thompson (2016), Roske et al. (2012), Scheidt (2017)
Processes	*Cumbersome, Slow, or Non-adaptive*	Ahner & Parson (2016), Hess & Valerdi (2010), Ilachinski (2017), Macias (2008), McLean et al. (2018), Tate & Sparrow (2018)
	Determining AI Test Adequacy	Hess & Valerdi (2010); Porter et al. (2018)
	"Gameable" Tests	Arnold & Scheutz (2018), Hernandez-Orallo (2017), Visnevski & Castillo-Effen (2010)
Stovepiping	*Effort Coordination*	Deonandan et al. (2010), Haugh et al. (2018), Hernandez-Orallo (2017), McLean et al. (2018), Porter et al. (2018), Porter et al. (2020), Ring (2009)
	Cradle-to-Grave Tester Participation Needed	Ahner & Parson (2016), Visnevski (2008); Porter et al. (2018)
	Need Iterative Testing or Continuum of Testing	Haugh et al. (2018), Hess & Valerdi (2010), McLean et al. (2018), Porter et al. (2018), Visnevski (2008),
	Resourcing Conflicts	Ahner & Parson (2016), Goerger (2004), Macias (2008), McLean et al. (2018), Visnevski & Castillo-Effen (2010), Zhou & Sun (2019)

Table 3: Lack of Methods, Tools, or Infrastructure

Sub-Category	Detailed Challenge	Supporting Articles
Method Needs	*Black Box Testing and Generalization*	AFSAB (2017), Laverghetta et al. (2019), Lennon & Davis (2018), Porter (2019)
	Evaluating Decision Model Adequacy	AFSAB (2017), Ahner & Parson (2016), Porter et al. (2020)
	Formal Methods are Inadequate	Defense Science Board (2012), Giampapa (2013), Goerger (2004), Kapinski et al. (2016), Menzies & Pecheur (2004), Micskei et al. (2012), USAF Chief Scientist (2011)
	V&V Processes	Goerger (2004), Ilachinski (2017), Lennon & Davis (2018), Menzies & Pecheur (2004), USAF Chief Scientist (2011), Schultz et al. (1993)
	Multi-Causal or Multi-Agent Performance Attribution	Ahner & Parson (2016), Baker et al. (2019), Goerger (2004), Greer (2013), Harikumar & Chan (2019), Haugh et al. (2018), Porter et al. (2020) Roske et al. (2012), Sparrow et al. (2018), Visnevski (2008)
	Test Point Prioritization or Sequential Test	Ahner & Parson (2016), Deonandan et al. (2010), Hernandez-Orallo (2017), Hess & Valerdi (2010), Micskei et al. (2012), Miller (2019), Mullins et al. (2017), Porter et al. (2018), Porter et al. (2020), Simpson (2020), Sparrow et al. (2018)
	Efficient Regression Testing	Ahner & Parson (2016), Defense Science Board (2016), Haugh et al. (2018), Ilachinski (2017), Luna et al. (2013), McLean et al. (2018)
	Compositional Verification	Lede (2019), Luna et al. (2013), USAF Chief Scientist (2011), Durst & Gray (2014)
	Integrative Verification	Scheidt (2017) Porter et al. 2020), Porter & Wojton (2020), Stracuzzi et al. (2020)
Scalability	*High-Fidelity Simulations*	Brabbs et al. (2019), Durst (2019), Kwashnak (2019), Lenzi et al. (2010), Mullins et al. (2017), Sparrow et al. (2018), Subbu et al. (2009)
	Subjective SME Evaluations	Goerger (2004), Hernandez-Orallo (2017)
	Manual Testing	Kapinski et al. (2016), Laverghetta et al. (2019), Schultz et al. (1993), Visnevski & Castillo-Effen (2010), Wegener & Buhler (2004)
	Multi-Agent Testing	Baker et al. (2019), Giampapa (2013), Lenzi et al. (2010), Luna et al. (2013)
Instrumentation	*Need Decision Traceability*	Ahner & Parson (2016), Defense Science Board (2016), Haugh et al. (2018), Porter et al. (2020), Sparrow et al. (2018)
	Test Event Replication	Ahner & Parson (2016), Defense Science Board (2016), Goerger (2004), Laverghetta et al. (2019), Visnevski (2008)
Simulation	*Defining Necessary Fidelity/Resolution*	Brabbs et al. (2019), Caseley (2018), Ahner & Parson (2016), Durst (2019), Gil and Selman (2019), Greer (2013), Kapinski et al. (2016), Kwashnak (2019), Laverghetta et al. (2019), Lenzi et al. (2010), Mueller et al. (2019), Porter et al. (2020), Sparrow et al. (2018), Visnevski & Castillo-Effen (2010)
	Extremely Difficult to Model Sufficiently	Ahner & Parson (2016), Goerger (2004)
	Injecting Valid Inputs	Ahner & Parson (2016), Goerger (2004), Laverghetta et al. (2019), Porter (2020b)
Ranges or Infrastructure	*Inadequate Range Instrumentation for AI*	Ahner & Parson (2016), Lennon & Davis (2018), Miller (2019), Tate & Sparrow (2019), Zacharias (2019a), Zacharias (2019b)
	Need testbeds for AI	Lennon & Davis (2018)
	No National Lab for AI	Gil and Selman (2019)

Sub-Category	Detailed Challenge	Supporting Articles
Personnel	*Identification, Recruitment, & Retention of AI-Skillsets*	Ahner & Parson (2016), Lennon & Davis (2018), Macias (2008), Roske et al. (2012), Wegener & Buhler (2004), Zacharias (2019a)
	Workforce Capacity	AFSAB (2017), Haugh et al. (2018)

Table 4: Safety & Security Issues

Sub-Category	Detailed Challenge	Supporting Articles
Safety	*System Decision Brittleness*	Ahner & Parson (2016), Lennon & Davis (2018)
	Elevated Risk with Autonomy	Arnold & Scheutz (2018), Deonandan et al. (2010), Haugh et al. (2018), Laverghetta et al. (2019), McLean et al. (2018), Micskei et al. (2012)
	Obtaining Safety Releases	McLean et al. (2018)
	Ensuring Range Safety	Ahner & Parson (2016), McLean et al. (2018)
	Need Run-time Assurance/Self-Monitoring	Ahner & Parson (2016), Arnold & Scheutz (2018), Defense Science Board (2012), Harikumar & Chan (2019), Laverghetta et al. (2019), Lede (2019), Lennon & Davis (2018), Menzies & Pecheur (2004)
Security	*Novel Cyber Exploitation / Adversarial Machine Learning*	Ahner & Parson (2016), Eaton et al. (2017), Goodfellow et al. (2015), Haugh et al. (2018), Qiu et al. (2019), Streilein et al. (2019)
	Protecting System Model/Software	Qiu et al. (2019), Streilein et al. (2019)
	Tactical Exploitability	Haugh et al. (2018)

Table 5: Lack of Policy, Metrics, & Standards

Sub-Category	Detailed Challenge	Supporting Articles
Policy	*Recertification After Regulations Change*	Ahner & Parson (2016)
Metrics	*Traditional Metrics Are Insufficient*	Eaton et al. (2017), Hernandez-Orallo (2017), Ilachinski (2017), Laverghetta et al. (2019), Macias (2008), Mueller et al. (2019), Roske et al. (2012), Visnevski & Castillo-Effen (2010)
	Multi-Agent/SoS Performance & Interoperability	Deonandan et al. (2010)
	Good Decisions	Ahner & Parson (2016), Hess & Valerdi (2010), Laverghetta et al. (2019), Roske et al. (2012)
	Measuring Trust	AFSAB (2017), Ahner & Parson (2016), Durst (2019), Ilachinski (2017)
	Perceptual Accuracy Metrics	Ahner & Parson (2016), Giampapa (2013), Harikumar & Chan (2019), Roske et al. (2012)
	System Learning	Ahner & Parson (2016)
Standards	*No Common Modeling Framework/Standards*	Ahner & Parson (2016), Goerger (2004), Lennon & Davis (2018), Durst & Gray (2014), Ring (2009), Visnevski (2008)
	Lack of Widely Used Common Architectures	Durst & Gray (2014)
	Lack Benchmarks Relevant to DoD	Hernandez-Orallo (2017), Hess & Valerdi (2010), Mueller et al. (2019), Roske et al. (2012)

Table 6: HSI Challenges

Sub-Category	Detailed Challenge	Supporting Articles
Trust	*Ensuring Appropriately Calibrated Trust*	Gunning (2017), Haugh et al. (2018), Porter et al. (2020), Wojton et al. (2020)
	Lack of Common Definition	Durst (2019)
Teaming	*Assessing Alignment of Humans & Machines (e.g., goals, SA)*	Ahner & Parson (2016), Endsley (2019), Haugh et al. (2018), Ilachinski (2017), Lennon & Davis (2018)
	Emergent Human Behavior	Harikumar & Chan (2019), Ilachinski (2017), Porter et al. (2020)
	Measuring Team Performance	Ahner & Parson (2016), Deonandan et al. (2010), Greer (2013), Haugh et al. (2018), Ilachinski (2017), Porter et al. (2020)
Meaningful Human Control	*Complacency or Inattention*	Caseley (2018), Ahner & Parson (2016), Porter (2020a)
	Inability to Predict Performance	Arnold & Scheutz (2018), Ahner & Parson (2016), Ferreira et al. (2013), Gunning (2017), Ilachinski (2017), Subbu et al. (2009)
	Operations Exceed Reaction Time	Arnold & Scheutz (2018), Caseley (2018), McLean et al. (2018), Porter (2020a)

Table 7: Requirements, Design, & Development Pipeline

Sub-Category	Detailed Challenge	Supporting Articles
Requirements	*Testable Requirements*	Ahner & Parson (2016), Lede (2019), Lennon & Davis (2018)
	Methods for Writing Requirements	Ahner & Parson (2016), Lede (2019), Lennon & Davis (2018)
	Address with Assurance Argument	Ahner & Parson (2016), Lede (2019), Lennon & Davis (2018), Scheidt (2017), Sparrow et al. (2018)
Assurance Aiding Designs	*Safety/Control Middleware*	Arnold & Scheutz (2018); Thuloweit (2019)
	Built-in Transparency and Traceability	Defense Science Board (2012), Gunning (2017), Haugh et al. (2018), Luna et al. (2013), Micskei et al. (2012), Porter et al. (2020), Sparrow et al. (2018), Zacharias (2019b)
	Explainable AI	Gunning & Aha (2019), Haugh et al. (2018), Mueller (2019), Porter et al. (2020), Porter (2020a)
	Modularity	AFSAB (2017), Roske et al. (2012), Porter et al. (2020), Scheidt (2017), Atherton (2019)
	Open Architectures	AFSAB (2017), Eaton et al. (2017), Lenzi et al. (2010), Durst & Gray (2014), Porter et al. (2020), Zacharias (2019a)
	Reuse Certified Capabilities	Caseley (2018)
	Blind Evaluation Metrics	Hernandez-Orallo (2017)
Design and Development Processes	*A Priori Risk Analyses*	Deonandan et al. (2010), Ferreira et al. (2013), Harikumar & Chan (2019)
	Iterative or Cyclic Development Paradigms	AFSAB (2017), Defense Science Board (2016)
	Model-Based Design	Ahner & Parson (2016), Kapinski et al. (2016)
	Statistical Engineering	Ahner & Parson (2016)
	Digital Twins	Arnold & Scheutz (2018)
Coordinate, Integrate, Extend Activities	*Desegregate CT, DT, & OT*	AFSAB (2017), Ahner & Parson (2016), Deonandan et al. (2010), McLean et al. (2018), Porter et al. (2018), Roske et al. (2012)
	Early Operator Involvement	Defense Science Board (2012), Gunning (2017), Mueller (2019)
	Earlier Realistic Testing	Defense Science Board (2012)
	Extend Test Throughout Lifecycle	Ahner & Parson (2016), Deonandan et al. (2010), Defense Science Board (2016), Ilachinski (2017), Porter et al. (2020)
	Iterative, Adaptive, or Sequential Development and Test	Ahner & Parson (2016), Defense Science Board (2016), Gunning (2017), McLean et al. (2018), Micskei et al. (2012), Mueller (2019), Porter et al. (2020), Simpson (2020), Sparrow et al. (2018), Visnevski & Castillo-Effen (2010)

Table 8: Improving Methods, Tools, Infrastructure, & Workforce

Sub-Category	Detailed Challenge	Supporting Articles
Leverage Existing Methods	*Formal Methods*	AFSAB (2017), Haugh et al. (2018), Luna et al. (2013)
	Complex Systems	Caseley (2018), Roske et al. (2012)
	Statistical Methods	Roske et al. (2012)
Performance Testing Methods	*Compositional Verification*	Ahner & Parson (2016), Durst (2019), Harikumar & Chan (2019), Laverghetta et al. (2019)
	Integrative Verification	Durst (2019), Harikumar & Chan (2019), Porter et al. (2020)
	Task-Specific Indicators	Baker et al. (2019)
	Domain-General Indicators	Baker et al. (2019), Durst (2019), Harikumar & Chan (2019), Hernandez-Orallo (2017)
	Skill Growth/Transfer	Ahner & Parson (2016), Baker et al. (2019)
Prioritize Test Points	*By A Priori Risk*	Deonandan et al. (2010)
	Using DOE	AFSAB (2017), Haugh et al. (2018), Hernandez-Orallo (2017), Hess & Valerdi (2010), Porter et al. (2020), Roske et al. (2012)
	Adaptively/Sequentially	Hess & Valerdi (2010), Miller (2019), Porter et al. (2020), Simpson (2020), Visnevski & Castillo-Effen (2010)
	Generic Efficiency	Luna et al. (2013)
Develop Methods	*Generic Call for Research*	AFSAB (2017)
	Choosing/Defining Test Conditions	Ahner & Parson (2016), Arnold & Scheutz (2018), Defense Science Board (2012), Hernandez-Orallo (2017), Lenzi et al. (2010), Scheidt (2017), Zhou & Sun (2019)
	DOE Extensions	AFSAB (2017), Ahner & Parson (2016)
	Demystification of Decision Making	Ahner & Parson (2016), Gunning (2017)
	Improved M&S	Ahner & Parson (2016)
	Data Management	Ahner & Parson (2016), Visnevski (2008)
Develop Testing Tools	*Automate Performance Evaluation*	Micskei et al. (2012), Wegener & Buhler (2004), Zhou & Sun (2019)
	Automate Space Exploration	AFSAB (2017)
	Iterate or Extend Tools	AFSAB (2017), Scheidt (2017), Miller Report, Streilein et al. (2019)
Develop Infrastructure	*Digital or LVC Testbeds*	AFSAB (2017), Ahner & Parson (2016), Eaton et al. (2017), Gil & Selman (2019), Haugh et al. (2018), McLean et al. (2018), Micskei et al. (2012), Porter et al. (2020), Streilein et al. (2019)
	Range Realism	Defense Science Board (2012), Gil & Selman (2019)
	Range Instrumentation	Defense Science Board (2012)
Establish Partnerships	*Centralize AI Research*	Gil & Selman (2019), Hernandez-Orallo (2017)
	Industry-Academy-Defense	AFSAB (2017), Gil & Selman (2019)
	Knowledge-Sharing	Gil & Selman (2019), Hernandez-Orallo (2017)
Personnel	*Create AI Talent Pipeline*	Gil & Selman (2019)
	Train Current Workforce	Goerger (2004), Ring (2009)
	Accreditation Practices	Goerger (2004)

Table 9: Test Strategies

Sub-Category	Detailed Challenge	Supporting Articles
Develop a Strategy Based on Existing Practices	*Inspiration: Current Practices*	Defense Science Board (2016), Durst (2019), Roske et al. (2012)
	Inspiration: Behavioral Science	Durst (2019), Goerger (2004), Gunning (2017), Ilachinski (2017), Porter et al. (2018)
	Inspiration: Complex Software	Caseley (2018), Harikumar & Chan (2019), Ilachinski (2017), Durst & Gray (2014), Roske et al. (2012)
Automate Testing	*Automate Performance Evaluation*	Micskei et al. (2012), Wegener & Buhler (2004), Zhou & Sun (2019)
	Automate Space Search	Haugh et al. (2018), Micskei et al. (2012), Schultz et al. (1993), Visnevski & Castillo-Effen (2010), Wegener & Buhler (2004), Zhou & Sun (2019)
	Automate Performance Evaluation	Micskei et al. (2012), Wegener & Buhler (2004), Zhou & Sun (2019)
	Use ML to Analyze AI	Micskei et al. (2012)
	Extensive Simulation	AFSAB (2017), Defense Science Board (2012), Haugh et al. (2018), Micskei et al. (2012), Visnevski & Castillo-Effen (2010), Wegener & Buhler (2004)
Coverage in Low Fidelity, Validate in High Fidelity	*LoFi: Abstracted System Model*	Giampapa (2013), Greer (2013), Haugh et al. (2018), Micskei et al. (2012)
	LoFi: Agent-Based Modeling	Defense Science Board (2016), Greer (2013), Ilachinski (2017)
	LoFi: Look for Failures	Deonandan et al. (2010), Giampapa (2013), Greer (2013), Haugh et al. (2018), Luna et al. (2013), Schultz et al. (1993), Subbu et al. (2009), Visnevski & Castillo-Effen (2010), Wegener & Buhler (2004)
	LoFi: Look for Inconsistency	Harikumar & Chan (2019), Zhou & Sun (2019)
	LoFi: Look for Edge Cases	Durst (2019), Haugh et al. (2018)
	HiFi: Realistic Simulation	Defense Science Board (2016), Laverghetta et al. (2019), Visnevski & Castillo-Effen (2010)
	HiFi: Live Testing	Defense Science Board (2016), Laverghetta et al. (2019), Visnevski & Castillo-Effen (2010)
Build Body of Evidence	*Accumulate Over Time*	Ahner & Parson (2016), Laverghetta et al. (2019), Lede (2019), Lennon & Davis (2018)
	Reuse Existing Evidence	Ahner & Parson (2016), Defense Science Board (2016), Lede (2019), Lennon & Davis (2018)
Run Time Assurance	*Focus: Certify Run Time Monitor*	Arnold & Scheutz (2018), Ahner & Parson (2016), Lede (2019), Lennon & Davis (2018)
	Gapped Middleware	Arnold & Scheutz (2018)

Table 10: Provide Risk Assurance for Safety and Security

Sub-Category	Detailed Challenge	Supporting Articles
Safety	*Safety Middleware*	Eaton et al. (2017), Haugh et al. (2018)
Security	*Basic Research*	Ilachinski (2017)
	Train and Test Against Adversarial Examples	Goodfellow et al. (2015), Haugh et al. (2018), Qiu et al. (2019), Streilein et al. (2019)
	Red Teaming	Haugh et al. (2018), Streilein et al. (2019)
	Cyber Testing Across Lifecycle	AFSAB (2017), Streilein et al. (2019)

Table 11: Adopt Policies, Standards, and Metrics

Sub-Category	Detailed Challenge	Supporting Articles
Create a Common T&E Framework	*Call for a Framework*	Defense Science Board (2016), Ilachinski (2017), Macias (2008), Ring (2009)
	Specific Proposal	Durst (2019), Eaton et al. (2017), Giampapa (2013), Harikumar & Chan (2019), Hess & Valerdi (2010), Lenzi et al. (2010), Luna et al. (2013), McLean et al. (2018), Micskei et al. (2012), Porter et al. (2020), Scheidt (2017), Visnevski & Castillo-Effen (2010)
Develop Standards	*Standards*	Defense Science Board (2016), Ilachinski (2017), Lenzi et al. (2010), Luna et al. (2013), Visnevski (2008)
	M&S Strategies	Defense Science Board (2016)
	Architectures	Lenzi et al. (2010), Luna et al. (2013), Ring (2009)
	Definitions	Deonandan et al. (2010), Ferreira et al. (2013)
Develop Metrics	*Generic Call for Metrics*	Harikumar & Chan (2019), Ilachinski (2017), Ring (2009), Roske et al. (2012), Scheidt (2017)
	Create AI Metric Development Methodology	Visnevski & Castillo-Effen (2010)
	Coverage Adequacy	AFSAB (2017), Ahner & Parson (2016), Deonandan et al. (2010), Hess & Valerdi (2010), Micskei et al. (2012)
	Define (Un)Desirable & (Un)Expected Behavior	Ferreira et al. (2013)
	Trust	Ahner & Parson (2016), Defense Science Board (2012), Porter et al. (2020), Tate et al. (2016), Wojton et al. (2020)
	Characterize Learning	Ahner & Parson (2016), Defense Science Board (2016), Harikumar & Chan (2019)
	Cyber Metrics	Streilein et al. (2019)

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

1. REPORT DATE *(DD-MM-YYYY)* 09-2020	**2. REPORT TYPE** IDA Publication	**3. DATES COVERED** *(From - To)*

4. TITLE ANDSUBTITLE Test & Evaluation of AI-Enabled and Autonomous Systems: A Literature Review	**5a. CONTRACT NUMBER** HQ0034-19-D-0001
	5b. GRANT NUMBER ____ ____ ____
	5c. PROGRAM ELEMENT NUMBER ____ ____ ____
6. AUTHOR(S) Daniel J. Porter (OED); John W. Dennis (SFRD);	**5d. PROJECT NUMBER** BD-09-2299
	5e. TASK NUMBER 229990
	5f. WORK UNIT NUMBER ____ ____ ____
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Institute for Defense Analyses 4850 Mark Center Drive Alexandria, Virginia 22311-1882	**8. PERFORMING ORGANIZATION REPORT NUMBER** NS-D-14331 H 2020-000326
9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES) Director, Operational Test and Evaluation 1700 Defense Pentagon Washington, DC 20301-1882	**10. SPONSOR/MONITOR'S ACRONYM(S)** DOT&E
	11. SPONSOR/MONITOR'S REPORT NUMBER

12. DISTRIBUTION / AVAILABILITY STATEMENT
This publication has not been approved by the sponsor for distribution and release. Reproduction or use of this material is not authorized without prior permission from the responsible IDA Division Director.

13. SUPPLEMENTARY NOTES
Project Leader, Heather Wojton

14. ABSTRACT

A growing body of work has explored, at different combinations of depth and breadth, the issues of or recommendations for the test and evaluation of autonomous military systems. The current effort summarizes a portion of this literature. In Appendix A, we provide summaries of the individual studies, with specific extractions of the challenges or recommendations their authors call out. For Appendix B, we created tables that categorize these challenges and recommendations, aggregating which authors advocate for these different viewpoints. The main text of this paper explores these tables at a high level. Some perspectives are overrepresented by the processes used to acquire articles and presentations, and readers should not assume that the number of different cited authors equates to the weight of support for a perspective.

15. SUBJECT TERMS
test, evaluation, verification, and validation (TEV&V); Artificial Intelligence (AI); Joint Artificial Intelligence (AI) Center (JAIC); Artificial Intelligence Enhanced Autonomous Capabilities; autonomy framework

16. SECURITY CLASSIFICATION OF:			**17. LIMITATION OF ABSTRACT**	**18. NUMBER OF PAGES**	**19a. NAME OF RESPONSIBLE PERSON** Heather Wojton (OED)
a. REPORT Unclassified	**b. ABSTRACT** Unclassified	**c. THIS PAGE** Unclassified	Unlimited	175	**19b. TELEPHONE NUMBER** *(include area code)* (703) 845-6811