SYM-AM-23-051

Excerpt from the Proceedings of the Twentieth Annual Acquisition Research Symposium

Acquisition Research: Creating Synergy for Informed Change

May 10–11, 2023

Published: April 30, 2023

Prepared for the Naval Postgraduate School, Monterey, CA 93943.

Acquisition Research Program
Department of Defense Management
Naval Postgraduate School

The research presented in this report was supported by the Acquisition Research Program at the Naval Postgraduate School.

To request defense acquisition research, to become a research sponsor, or to print additional copies of reports, please contact any of the staff listed on the Acquisition Research Program website (www.acquisitionresearch.net).

Acquisition Research Program
Department of defense management
Naval Postgraduate School

Digital Engineering Enhanced T&E of Learning-Based Systems

Laura Freeman—is a Research Associate Professor of Statistics and dual hatted as the Deputy Director of the Virginia Tech National Security Institute and Assistant Dean for Research for the College of Science. Her research leverages experimental methods for conducting research that brings together cyber-physical systems, data science, artificial intelligence (AI), and machine learning to address critical challenges in national security. She develops new methods for test and evaluation focusing on emerging system technology. Previously, Freeman was the assistant director of the Operational Evaluation Division at the Institute for Defense Analyses (IDA). Freeman also served as that acting senior technical advisor for Director Operational Test and Evaluation (DOT&E). Freeman has a BS in aerospace engineering, an MS in statistics, and a PhD in statistics all from Virginia Tech.

Paul Wach—is a Research Assistant Professor in the Intelligent Systems Division of the Virginia Tech National Security Institute. His research interests include the intersection of theoretical foundations of systems engineering, digital transformation, and artificial intelligence. Paul is a member of the Intelligent Systems Division of the Virginia Tech National Security Institute. He is the President and Founder of the Virginia Tech student division of INCOSE. Wach also works for The Aerospace Corporation, leading enterprise digital engineering transformation. His prior work experience is with the Department of Energy, two National Laboratories, and the medical industry. Wach received a BS in Biomedical Engineering from Georgia Tech, MS in Mechanical Engineering from the University of South Carolina, and completed a PhD in Systems Engineering at Virginia Tech with graduation in December 2022.

Justin Krometis—is a Research Assistant Professor in the Intelligent Systems Division of the Virginia Tech National Security Institute. His research is in the development of theoretical and computational frameworks to address data analytics problems, such as how to incorporate and balance data and expert opinion into decision-making, how to fuse data from multiple sources, and how to estimate model parameters, including high- or infinite-dimensional quantities, from noisy data. His areas of interest include Bayesian inference, parameter estimation, machine learning, data science, and experimental design. Dr. Krometis holds a PhD in mathematics, a MS in mathematics, and a BS in mathematics, and a BS in physics, all from Virginia Tech.

Atharva Sonanis—is an MS in Mechanical Engineering student working with Jitesh Panchal at the Design Engineering Laboratory at Purdue University (DELP). Atharva's research interests include robotics, controls, computer vison, machine learning and systems. He received his BE in Mechanical Engineering from M.I.T. College of Engineering, Pune. While pursuing his bachelor's degree, he was selected as a Cummins Scholar, and received the opportunity to work at Cummins Technical Centre India in the R&D department. Additionally, he holds a diploma in Mechanical Engineering from Government Polytechnic, Miraj.

Jitesh Panchal—is a Professor of Mechanical Engineering at Purdue University. He received his BTech (2000) from Indian Institute of Technology (IIT) Guwahati, and MS (2003) and PhD (2005) in Mechanical Engineering from Georgia Institute of Technology. Panchal's research interests are in (1) design at the interface of social and physical phenomena, (2) computational methods and tools for digital engineering, and (3) secure design and manufacturing. He is a recipient of CAREER award from the National Science Foundation (NSF); Young Engineer Award, Guest Associate Editor Award, and three best paper awards from ASME; and was recognized by the B.F.S. Schaefer Outstanding Young Faculty Scholar Award, the Ruth and Joel Spira Award, and as one of the Most Impactful Faculty Inventors at Purdue University. He is a co-author of two books and has co-edited one book on engineering systems design. He has served on the editorial board of international journals including ASME Journal of Mechanical Design, ASME Journal of Computing and Information Science in Engineering. He is a program chair of the ASME IDETC/CIE conference, and the past chair of the ASME Computers and Information in Engineering (CIE) division.

Peter Beling—is a professor in the Grado Department of Industrial and Systems Engineering and Director of the Intelligent Systems Laboratory at the Virginia Tech National Security Institute. Beling's

research interests lie at the intersections of systems engineering and artificial intelligence (AI) and include AI adoption, reinforcement learning, transfer learning, and digital engineering. His research has found application in a variety of domains, including mission engineering, cyber resilience of cyber-physical systems, prognostics and health management, and smart manufacturing. He received his PhDin operations research from the University of California at Berkeley.

Abstract

The design of test and evaluation (T&E) programs requires new thinking for learning-based systems enabled by AI. A critical question is how much information is needed about the training data, the algorithm, and the resulting performance for testers to adequately test a system. The answer to these questions will inform acquisition of data/model rights for learning-based systems. The principal objective of this research is to understand how increasing government access to the models and learning-agents (AI algorithms) used in system design might decrease the need and expense of testing and increase confidence in results. The principal hypotheses investigated in this incubator project are that the number of samples needed to test AI/ML models to an acceptable degree of assurance ca be reduced if we have access to the models themselves (in mathematics or software), reduced still further if we also have access to the algorithms and data used to train the models, and reduced further yet if we also have access to systems models and other artifacts of the digital engineering process. Therefore, the cost of acquisition can be reduced if T&E programs are based on the optimal balance between the cost of acquiring the technical data/algorithm rights of AI/ML systems, and the cost of testing those systems. This research establishes theory and methods for exploring how T&E requirements can and should change as a function of the test team knowledge of the technical specifications of learn based systems (LBS).

Introduction

Artificial intelligence and machine learning (AI/ML) has moved beyond being a research field to being an essential element of next-generation military systems. The discipline of verification and validation of AI/ML enabled complex systems, however, in its nascent stage. Little is understood about how to identify changes in operating conditions or adversarial actions that might cause the performance of an AI/ML model to deviate from design limits (McDermott, 2021). The challenges in this regard are amplified when considering autonomous functions that may engage in self-learning over the long-life cycles seen in military systems.

The objective of this research was to develop approaches to the design of test and evaluation (T&E) programs and the acquisition of data/model rights for learning-based systems (LBS). Freeman (2020) proposes 10 different themes for how T&E will need to change for ML/AI systems. One theme is the need for a risk-based framework approach. This research seeks to explore the risks associated with varying levels of knowledge of ML/AI training data and model insights. The principal objective was to understand how increasing government access to the models and learning-agents (AI algorithms) used in system design might decrease the need and expense of testing and increase confidence in results.

The current approach to T&E involves treating the system in a black-box fashion, i.e., the system is presented with sample inputs, and the corresponding outputs are observed and characterized relative to expectations. While such an approach works well for traditional static systems, test and evaluation of autonomous intelligent systems presents formidable challenges due to the dynamic environments of the agents, adaptive learning behaviors of individual agents, complex interactions between agents and the operational environment, difficulty in testing black-box machine learning (ML) models, and rapidly evolving ML models and AI algorithms (Cody, 2019).

Our principal hypotheses are that the number of samples needed to test AI/ML models to an acceptable degree of assurance can be reduced if we have access to the models themselves (in mathematics or software), reduced still further if we also have access to the algorithms and data used to train the models, and reduced further yet if we also have access to systems models and other artifacts of the digital engineering process. Therefore, the cost of acquisition can be significantly reduced if T&E programs are based on the optimal balance between the cost of acquiring the technical data/algorithm rights of AI/ML systems, and the cost of testing those systems.

This paper develops theory based in systems theory that captures changes in the systems and the state-space in which it operates through the concept of systems morphisms. The Theoretical Background section provides overarching theory and a system concept model. The onion model describes different levels of system knowledge and a context for defining the abstraction of the system. The Experimental Testbed section describes two pilot scenarios to demonstrate how multiple phases of testing contribute to the evaluation of an AI enabled systems. The Bayesian Framework section presents the Bayesian analytical framework for combining information across the multiple phases of testing. This analytical framework also reflects the changing system configuration and context. The Potential Testbed and Future Work section discusses future work for validating the concept through a full system model and experiment. In summary, this work essentially constitutes the building blocks for investigating the cost-benefit for test data collection on a realistic system in future phases.

Theoretical Background

At the core of this research is systems theory outlined by Bertalanffy and Sutherland (1974). Specifically, we build from the lineage of the systems theorist Wymore (1967) defined the Mathematical Theory of Systems Engineering and has been credited for coining the term model-based systems engineering (Bjorkman et al., 2013). A mathematical mechanism used in Wymorian systems theory is the system specification morphism; where a morphism is a mathematical characterization of the preservation of equivalence between a pair of system specifications (Zeigler, 2018).

System specifications may be defined at many levels within a hierarchy. The hierarchy of system specification is a prominent aspect of a branch of Wymorian systems theory commonly referred as computational systems theory, or formally known as the Theory of Modeling and Simulation (Wach et al., 2021). Each level of the hierarchy of system specification reveals further detail as to the knowledge of the structure from external interfaces and interactions to internal component and coupling knowledge. Furthermore, within each level of system specification, a morphism essentially characterizes abstraction and elaboration of detail.

Simply put, parameter morphisms coupled with specifications within the hierarchy, is a mapping of parameter space along with state space. The parameter morphism is an explicit documentation of allowable deviations (approximations) from exact morphisms, as is the expectation with the input/output observation frame and network of systems morphisms, relative to changes in parameter sets. A simple example of a parameter morphism is the selection of the mean versus a distribution as a parameter test set.

Lastly, the framing of the hierarchy and associated morphisms is important to understand the systems theoretical context as a whole. First, the relationship between the input/output (IO) observation frame and the network of systems is a one-to-many specification relationship, meaning that one system specification at the IO observation frame can lead to

specification of many (maybe infinite) network of system specifications. However, each network of system specification can map to only one specification at the IO observation frame level. Second, a morphism at the IO observation frame level does not guarantee a morphism at the network of systems level. Third, however, a morphism at a network of systems level implies a morphism at the input/output observation frame level. These are systems theoretic concepts we use to underpin our methodology for T&E of LBS.

Methodology

The practice of engineering systems is reliant on use of surrogate analogies for T&E. In some cases, we may not have access to the fielded system until late in the program and, therefore, select a surrogate as an analogous representation of the current (phase appropriate) design of the system of interest. In other cases, the system of interest may be fielded and we want to understand observed behavior, for which we may use a surrogate, analogous environment for testing the fielded system (or analogous test system). These activities are typically thought of as necessary risk reduction, for which we characterize the validity of the analogies through the use of systems theoretic morphisms.

Consider the following example to provide further context: The IO observation frame morphism could be used to characterize the change in operational conditions and change in adversarial action. The network of systems morphism can be used to characterize the changes in implementation of a LBS subsequent to changes in operational conditions and adversarial actions. Furthermore, from the last paragraph of the previous section, a morphism between system implementations (i.e., network of systems morphism) implies a morphism at the mission level, which, therefore, is indicative of mission success.

Commonly associated with LBS is the onion model shown in Figure 1. In the outer layer, we have minimum knowledge of the system context, which we categorize as mission knowledge. In the second to outer layer, we begin to have knowledge of the interior structure in the form of a functional architecture. In the third to outer layer (second to inner layer), we have knowledge of the agent cognitive functions. In the inner most layer, we have maximum knowledge in the form of knowledge of the physical implementation of the system of interest. From a systems theory perspective, we can provide a view of validity of analogies relative to the onion model.

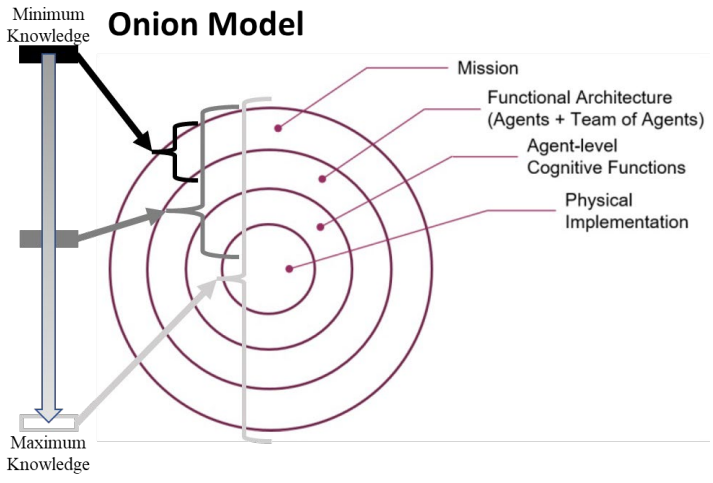

Figure 1. Onion Model Used to Understand Layers of LBS

In Figure 2, we provide a systems theoretic context. We show the real mission to the top left

and the preservation of equivalence to surrogate T&E context shown in the top right. We propose characterization of this equivalence through systems theoretic mechanisms, such as the IO observation frame and associated morphism. We also show the field system to the bottom left and the preservation of equivalence to surrogate model shown in the bottom right. We propose characterization of this equivalence through systems theoretic mechanisms, such as the network of systems and associated morphism.

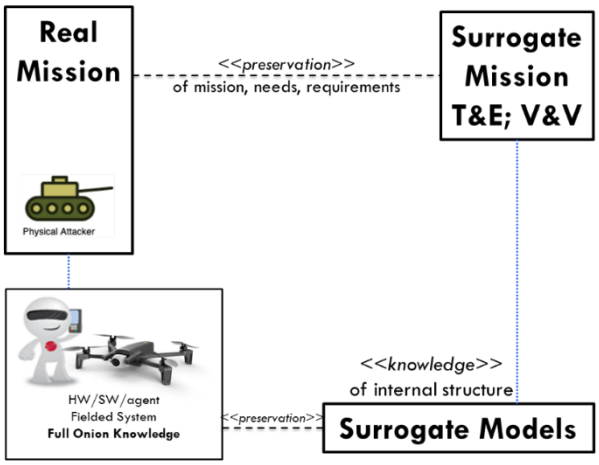

Figure 2. High-Level View of Systems Theory Perspective of the Onion Model

For this project we used the mission context of detection of a potential attacker, consistent with Silverfish (Carter et al., 2019). Rather than focus on the full system of systems of Silverfish, we selected to focus on the unmanned aerial vehicle (UAV) component as the system of interest for this research project. In Figure 3, we provide further explanation to our set of experiments within the context of systems theory and the onion model.

First, we have used the You Only Look Once (Agent YOLO) algorithm as our agent, which has an unknown T&E context conducted prior to our acquisition of the agent. Therefore, we cannot determine its morphic equivalence to the real mission and must conduct further testing. The new T&E mission analogies are expected to be characterized through systems theoretic morphisms. For this project, we used a series of T&E surrogate mission contexts of the potential attacker in the form of a soccer match (i.e., red versus blue) and automobile detection (truck versus other type of vehicle). While the soccer match was a simulation (video from the internet), the automobile surrogate mission context was both a simulation and physical test.

Second, we were not able to acquire the physical hardware expected for the fielded system at the onset of the project. Therefore, we relied on surrogate models, for T&E, that we believe to be analogous to the fielded system. Each surrogate model is expected to be morphically characterized to determine its equivalence relative to the fielded system. For this project we have selected a series of surrogate models for the UAV. First, the initial Agent YOLO may only be analogous as far as the cognitive function is concerned. Second, we used a surrogate drone, which has lower cost and quality of hardware than the fielded systems. Last, it should also be noted that even when we have access to the expected fielded system, the morphic validity of the analogies must also be confirmed. For this, we suggest that a digital twin (i.e., simulation) and final product (or physical twin) from low-rate initial production (LRIP) be used for an initial operational test and evaluation (IOT&E), both of which should be morphically characterized for its equivalence to the expected (or measured) reality.

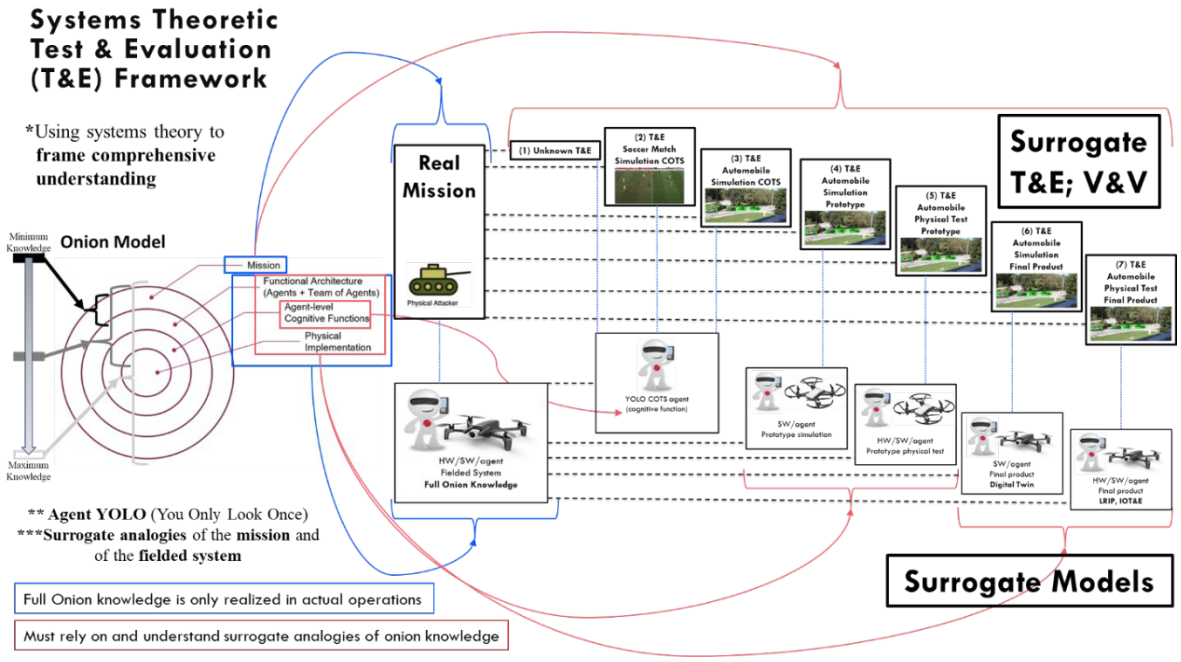

Figure 3. Proposed Systems Theoretic Test and Evaluation Framework

Because the full knowledge relative to the onion model can only be known once the final product design becomes the fielded system and is placed in its real mission context, we must rely on and understand surrogate analogies to provide confidence in mission success. We have selected to use Bayesian methods, such as discussed by Salado and Kannan (2018), to characterize confidence in mission success relative to knowledge on the morphic equivalence. Further detail on the use of Bayesian methods is provided in another section.

Experimental Testbed

The broader objective for creating the experimental testbed is to assist in validating the T&E framework for learning-based systems. For this research, the specific goal is to demonstrate how the T&E framework can be utilized for a specific scenario where the goal is to detect the presence of enemies, tracking them, and sending a signal for Silverfish protected field.

The testbed consists of (a) scenarios, (b) hardware, and (c) software. Two scenarios are created as surrogate problems as a part of creating experimental testbeds.

Scenario 1: Person Identification and Tracking in a Soccer Game: This scenario is based on a soccer game as a surrogate problem using stock video to detect players belonging to different teams and their location in the field. The players, shown in Figure 4, are classified into two different teams based on their apparel colors and patterns. The idea here is to showcase the different teams as allies and enemies. In addition to this, the location coordinates of the players are continually tracked.

Figure 4. Scenario 1: Person Identification and Tracking in a Soccer Game

Scenario 2: Vehicle Detection and Tracking: This scenario is based on automobile detection and tracking as a surrogate problem to detect vehicular traffic, location coordinates and their velocities (see Figure 5). The vehicles are categorized based on their sizes, i.e., small vehicles represent allies (friends) and large vehicles represent enemies. Similar to Scenario 1, the location coordinates of the vehicles are detected along with their velocities.

Figure 5. Scenario 2: Vehicle Detection and Tracking

The coordinates obtained from the two scenarios are mapped and visualized on a grid shown in Figure 6.

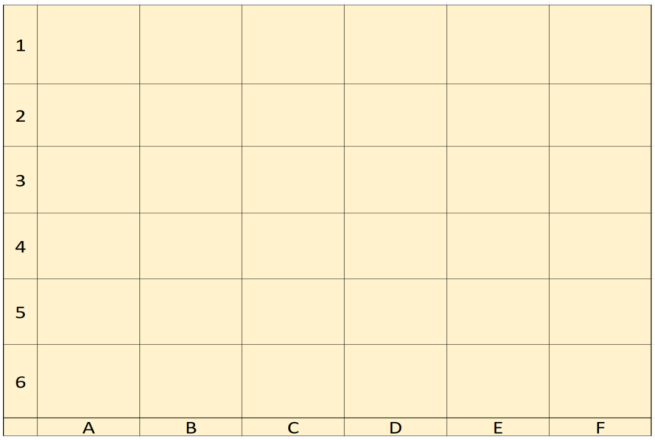

Figure 6. Grid Used for Visualization and Mapping

HARDWARE. For hardware implementation, two drones namely, Ryze Tello (lower fidelity prototype drone) and Parrot ANAFI (higher fidelity prototype drone) are used. The specifications of the Ryze Tello drone and the Parrot ANAFI drone are shown in Figure 7.

Figure 7. Comparison of the Specifications of the Higher and Lower Fidelity Drones

Lower fidelity prototypes are used to test whether the high-level design concepts can be translated into tangible outputs. On the other hand, higher fidelity prototypes provide outputs that are as similar as possible to the desired requirements defined initially. The drones capture videos which are then segmented into images frame by frame. The differences in the images from the two drones can be clearly seen in terms of the resolution, field of view and stability.

SOFTWARE. The primary goal of the software implementation is to identify the location coordinates of the allies and enemies and track them in real time. To do so, videos captured from the drones are used as input, and the output being series of location coordinates. This implementation broadly consists of four steps: image preprocessing, object detection and classification, object tracking, and mapping. Figure 8 provides a high level overview of the process.

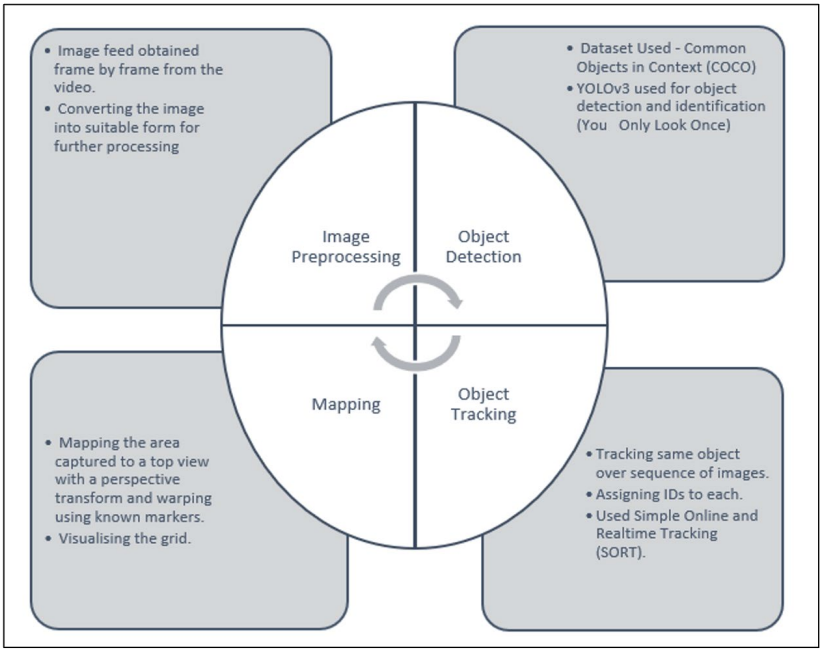

Figure 8. High Level Overview of Software Processes

Image Preprocessing: The goal of this step is to retrieve a series of clean images from the videos to prepare them for the further steps and to reduce computation time. The steps in image preprocessing are as follows:

1. Raw videos obtained from the hardware are segmented frame by frame into a series of images.
2. Images are resized to a lower size to increase computation speed.
3. Gaussian Blur is used to smoothen the images and to reduce unwanted noise.
4. Images are cropped to obtain the region of interest.

To simulate the different qualities of video camera from different hardware, i.e., a lower fidelity and higher fidelity input in Scenario 1:Person Identification and Tracking in a Soccer Game, the original video is used as a higher fidelity input and the blurred version of the original video is used as a lower fidelity input. Figure 9 shows the results for Scenario 2.

Figure 9. Drone Image Comparison (Above: Higher Fidelity, Below: Lower Fidelity)

Object Detection and Classification: The goal of this step is the detection, classification, and localization of the objects present in the frame. Here, the preprocessed images are used as the input, and passed through a trained or pre-trained object detection model to receive object location and classes. For this purpose, YOLOv3 (You Only Look Once, version 3) is used, which is an object detection algorithm that identifies specific

objects in videos or images. A custom object detection model is trained for the soccer game scenario to detect and classify the players into different teams. Whereas, for the vehicle scenario, to detect and classify allies (friends) and enemies, a pre trained model is customized using the COCO (Common Objects in Context) dataset (Lin et. al., 2014), which is a large-scale object detection, segmentation, and classification dataset. The COCO dataset has more than 2,00,000 labelled images and more than 100 categories.

Object Tracking: The goal of this step is to track the movement of an object, which involves tracking of the detected objects frame-by-frame and storing its location coordinates along with some relevant information. A unique identification number is assigned to each detected object for the duration of which it is continuously tracked. There are several challenges associated with object tracking such as occlusion, discontinuity in detections, etc. To tackle these issues, the Simple On-line Real-time Tracking (SORT) algorithm is used. We are successfully able to perform tracking of each object along with finding its approximate velocity.

Mapping: The output obtained from the object tracking step is utilized to map and visualize the allies and enemies on a grid. The visualization is useful for sending a signal to silverfish protected field. This is accomplished using warping techniques and perspective transformations of a known field or using markers to a visualization grid. Figure 10 is a representation of the soccer scenario in the grid format with exact location coordinates. The blue and white dots depict players in the teams whereas the black dot is the soccer ball.

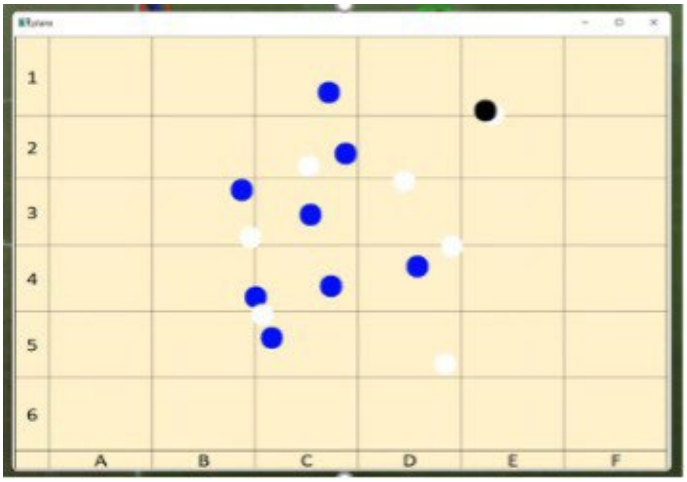

Figure 10. Grid Data Representation Example

The results obtained from above steps, i.e., the location coordinates of the detected objects are used to evaluate the detection accuracy of identified objects to be used in the Bayesian Framework, described next.

Bayesian Framework

We can characterize the relationships between different models—simulation environments vs. low-fidelity systems vs. higher-fidelity systems—via a Bayesian network. A Bayesian network is a graph model that describes the relationship between nodes via probabilities. To illustrate the concept, we consider a detection system with two outcomes—either the target is detected or it is not detected—and two true states—either the target is present or it is not—with four possible combinations. These cases are summarized in Table 1.

Table 1. Cases for a Target Detection Problem

Case #	Case	Target Present?	Target Detected?
1	True Positive	Yes	Yes
2	False Negative	Yes	No
3	False Positive	No	Yes
4	True Negative	No	No

One might imagine each of these cases having a different "cost" from a T&E perspective—i.e., a false negative (target is present but not detected) may have more operational cost than a false positive (target is falsely detected). The goal of T&E is ultimately to characterize that cost, e.g., to compute its *expected value*, i.e., the cost of each case (C_i) times the probability of each case (P_i):

$$E(C) = \sum_{i=1}^{4} C_i P_i$$

The Bayesian framework considers each of the probabilities P_i for the final fielded system as a function of the probabilities for the analogous systems. That is, if the probabilities for the simulated environment and a lower-fidelity prototype are P^{sim} and P^{low}, respectively, then the final probability P_i can be written in terms of conditional probabilities:

$$P_i = P(x \in C_i) \propto \sum_{i=1}^{4} P\left(x \in C_i \middle| x \in C_i^{sim}\right) P(x \in C_i^{sim})$$

Here the more complicated equation has necessitated more complicated notation: $P(x \in S_i)$ is the probability of Case i and $P(A|B)$ is the probability of A given B. So, the above equation means that the probability of, for example, getting a true positive (Case 1) in the fielded system is the probability of getting Case j in the low fidelity system multiplied by the probability of getting Case 1 in the fielded system given that we got Case j in the low-fidelity system, summed across j. This may seem like—and indeed is—a more complicated way of writing the same thing. However, if we can accurately estimate the conditional probabilities in, it allows us estimate the probabilities P_i and ultimately the cost by mostly running lower-fidelity tests. The same mechanism can then be used to capture the relationship between the lower-fidelity test and the simulated environment. The Bayesian network summarizing the relationship between these conditional probabilities is shown in Figure 11.

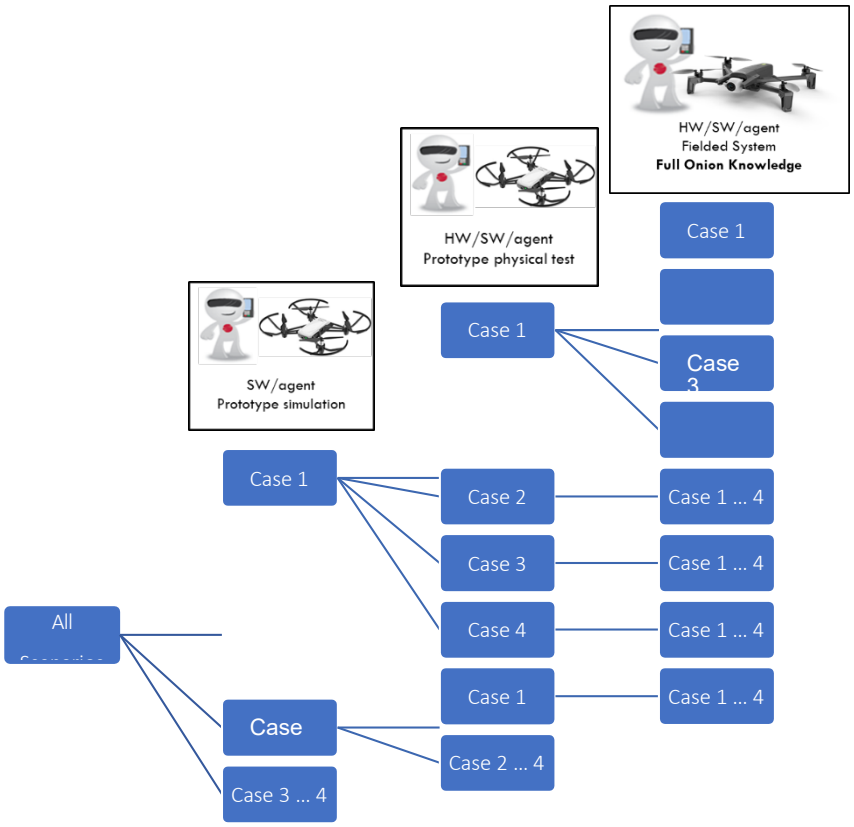

Figure 11. A Bayesian Network for the Detection Case. Scenarios are Divided into Four Cases (True/False Positive/Negative) Across Three Test Cases (Software Environment, Prototype/Low-Fidelity, Fielded System).

Next, we describe briefly how to estimate the probabilities in the previous section. In this case, we actually use a different kind of Bayesian procedure known as Bayesian inference. We begin with an estimate of the probability distribution called the *prior*, and then update that estimate as we test. For the detection case, we can model the outcomes with a binomial distribution with unknown success probability p. There is a fairly standard approach in the statistics community to estimating p. First, the prior is typically chosen to be a beta distribution $B(\alpha, \beta)$ where α, β are parameters that can be tuned to the problem. For example, one might give the prior a weight N_{prior} and start with a guess for p which we denote p_{prior}; we then would set $\alpha = p_{prior}N_{prior}$

and $\beta = N_{prior} - \alpha$. Then if tests yield s successes and f failures, we would update our estimate

to be $B(\alpha + s, \beta + f)$. This procedure is illustrated in Figure 12. Here $p_{prior} = 0.4$ but we see the inference procedure closing in on the true value of $p = 0.7$ as more tests are taken.

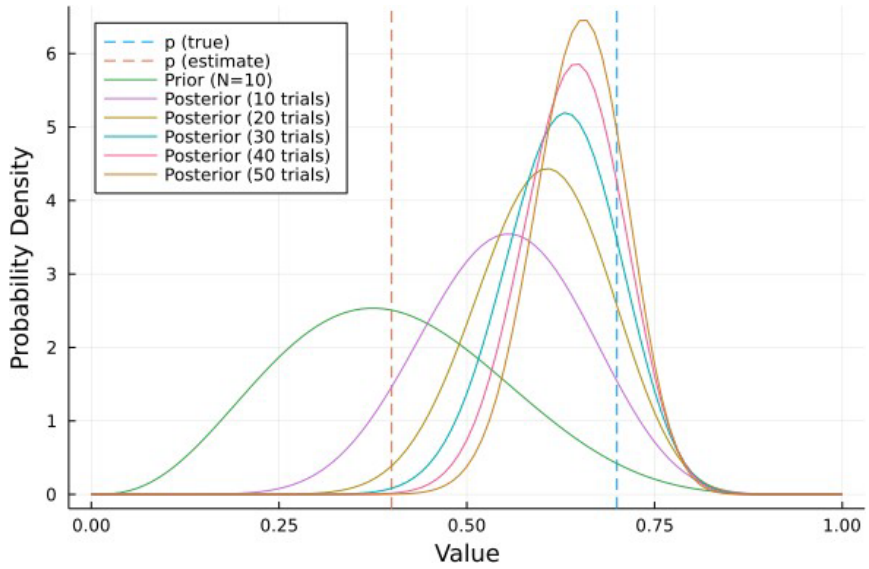

Figure 12. A Bayesian Approach to Estimating the Probability of Success in a Binomial Distribution; Here the Prior Estimate of the Probability of Success is 0.4 and the True Value is 0.7. The Inference Procedure Updates the Estimates as Tests are Conducted, Closing in on the True Value.

Potential Testbed and Future Work

A major challenge in conducting AI enabled systems research is that physical realizations are needed for T&E research. As a testbed for the methodologies, future work should explore these concepts on a full hypothetical weapons system, moving beyond the embedded AI algorithm. The system highlighted earlier, known as Silverfish, is a networked munition system designed to deny ground to the enemy using ground-based weapons, known as obstacles, that can engage unauthorized persons or ground vehicles within the denied area. Surveillance sensors including static infrared and video cameras and target characterization sensors, such as acoustic and seismic sensors, monitor the area to provide the operator with situational awareness regarding persons and vehicles. An unmanned aerial vehicle also provides surveillance and early warning information. Silverfish exists as a hybrid simulation/hardware emulation characterized in model-based systems engineering (MBSE) terms by a set of SysML models describing its architecture and functions from several perspectives. It also includes AI/ML models for detecting cyber-attacks on the UAV.

Future work could leverage the Silverfish testbed and expand the testbed into physical implementations beyond the computer vision use case. Physical implementations in addition to MBSE representations would enable the direct execution of a T&E program on the Silverfish testbed. Future work should also include purposefully varying the systems knowledge (based on the onion model), the complexity of the systems and its operating environments (number of morphisms), and determine minimally adequate testing as a function of those variables.

This paper stablished the theory and methods for exploring how T&E requirements can and should change as a function of the test team knowledge of the technical specifications of an AI enabled system. The research developed theory based in systems theory that captures changes in the systems and the state-space in which it operates through the concept of systems morphisms. The onion model describes different levels of system knowledge and a context for defining the abstraction of the system. The project experimented with two pilot scenarios to demonstrate how multiple phases of testing contribute to the evaluation of an AI enabled system. Finally, we present the Bayesian

analytical framework for combining information across the multiple phases of testing. This analytical framework also reflects the changing system configuration and context. In summary, this work essentially constitutes the building blocks for investigating the cost-benefit for test data collection on a realistic system in future phases.

References

Bertalanffy, L. V. (1968). *General system theory: Foundations, development, applications*. G. Braziller.

Carter, B., Adams, S., Bakirtzis, G., Sherburne, T., Beling, P., Horowitz, B., & Fleming, C. (2019). A preliminary design-phase security methodology for cyber–physical systems. *Systems*, *7*(2), 21.

Cody, T., Adams, S., & Beling, P. A. (2019, April). A systems theoretic perspective on transfer learning. In *2019 IEEE International Systems Conference (SysCon)* (pp. 1–7). IEEE.

Fleming, C., Elks, C., Bakirtzis, G., Adams, S. C., Carter, B., Beling, P. A., & Horowitz, B. (2020). Cyberphysical security through resiliency: A systems-centric approach. *arXiv preprint arXiv:2011.14469*.

Freeman, L. (2020). Test and evaluation for artificial intelligence. *Insight*, *23*(1), 27–30.

Lin, T. Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., ... & Zitnick, C. L. (2014). Microsoft coco: Common objects in context. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6–12, 2014, Proceedings, Part V 13 (pp. 740–755*). Springer International Publishing.

McDermott, T. A., Blackburn, M. R., & Beling, P. A. (2021). Artificial intelligence and future of systems engineering. *Systems Engineering and Artificial Intelligence*, 47–59.

Salado, A., & Kannan, H. (2018). A mathematical model of verification strategies. *Systems Engineering*, *21*(6), 593–608.

Wymore, A. W. (1967). *A mathematical theory of systems engineering: The elements.* Wiley.

Zeigler, B. P., Muzy, A., & Kofman, E. (2018). *Theory of modeling and simulation: Discrete event & iterative system computational foundations*. Academic Press.

Acquisition Research Program
Department of Defense Management
Naval Postgraduate School
555 Dyer Road, Ingersoll Hall
Monterey, CA 93943

www.acquisitionresearch.net