Safe Testing of Autonomous Systems Performance

David Scheidt, Robert Lutz, William D'Amico, Dean Kleissas, Robert Chalmers, Robert Bamberger,
Johns Hopkins University Applied Physics Laboratory
11100 Johns Hopkins Road, Laurel, MD
david.scheidt@jhuapl.edu, robert.lutz@jhuapl.edu, william.d'amico@jhuapl.edu, dean.kleissas@jhuapl.edu,
robert.chalmers@jhuapl.edu, robert.bamberger@jhuapl.edu

ABSTRACT

The role of unmanned platforms is rapidly expanding across a wide range of defense and homeland security missions. Currently operational unmanned vehicles are "tele-operated", using a command and control link to a remotely located pilot. However, operational complexity, operational pace, and a need to function in communication denied environments necessitate a trend toward autonomous unmanned vehicles. Autonomous systems that make independent decisions in complex engagements, such as the Navy's Autonomous Aerial Cargo Unmanned System, are currently under development and will require development and operational testing within the next 3-5 years.

Testing of autonomous systems presents some unique and vexing challenges. For instance, the infinite number of variations of test conditions that can exist to stimulate autonomous behaviors and the complexity of the interactions that can occur among multiple autonomous systems combine to make comparative measurement of autonomous system performance extremely difficult. Also, the inherent unpredictability of decision making by autonomous systems may result in decisions that are considered unsafe by managers of live test ranges. Advanced test and evaluation techniques that focus on the unique challenges of autonomy represent a clear and increasing need within the DoD.

The Safe Testing of Autonomy in Complex, Interactive Environments (TACE) Program is a research initiative to develop an advanced test infrastructure that can measure the performance of autonomous systems operating in complex Live-Virtual-Constructive (LVC) environments while ensuring that the autonomous system does not violate range safety policy. This paper will provide an overview of the TACE hardware and software architecture and will highlight the LVC testing that has been performed at the Aberdeen Test Center to validate TACE capabilities. A discussion of anticipated transition activities with DoD partner programs will also be provided.

ABOUT THE AUTHORS

DAVID SCHEIDT is a member of the Principal Professional Staff at Johns Hopkins University's Applied Physics Laboratory where he conducts research on distributed intelligent control systems. Mr. Scheidt has 30 years of experience in the research and development of distributed information management systems, robotics, artificial intelligence and process control systems. Throughout his career Mr. Scheidt has conducted research in concert with his development efforts, publishing over 50 peer reviewed publications for research funded by the National Computer Security Center (NCSC), ONR, NASA, DISA, OSD NII and the US Army. Mr. Scheidt currently conducts autonomous systems and intelligent controls research and is the principal investigator on research initiatives that focus on the intelligent diagnosis, reconfiguration and planning of ship auxiliary systems, spacecraft and unmanned vehicles.

ROBERT LUTZ is a principal staff scientist at The Johns Hopkins University Applied Physics Laboratory in Laurel, MD. His background includes 34 years of practical experience in the development, use, and management of models and simulations across all phases of the DoD systems acquisition process. He currently serves as the LVC and testing lead for the TACE project and as the Airspace Integration Modeling and Simulation (M&S) lead for the Navy's Triton Program. Mr. Lutz also serves as the Chair of the Simulation Interoperability Standards Organization (SISO) Board of Directors, serves on the Tutorial Board and Fellows Committee at the Interservice/Industry Training, Simulation and Education Conference (I/ITSEC), and is a guest lecturer on M&S-related topics in The Johns Hopkins University Whiting School of Engineering.

WILLIAM D'AMICO, Ph.D., is a principal staff scientist at The Johns Hopkins University Applied Physics Laboratory in Laurel, MD. His background includes over 40 years of research, development, and program management in gun-launched projectile systems, micro-electromechanical systems, and unmanned aerial vehicles.

Dr. D'Amico has conducted many flight tests at multiple test ranges and is familiar with the needs and requirements for developing technologies to support the test and evaluation community. He has been a key contributor to several TRMC-sponsored Central Test and Evaluation Project activities and T&E/S&T programs.

ROBERT CHALMERS is a Senior Staff researcher at the Johns Hopkins University Applied Physics Laboratory. A graduate of Florida State University, Mr. Chalmers is the software team lead for the TACE program, and has a lengthy background in the development of tactical modeling and simulation tools for a variety of military applications. For over a decade he has led JHUAPL teams in the development of algorithms, sensor and information models, mission simulations, human command interfaces, and CONOPS for the use of swarming autonomous unmanned vehicle systems and other related command and control constructs.

DEAN KLEISSAS is Senior Staff at the Johns Hopkins University Applied Physics Laboratory and a member of the Intelligent Systems Group. He received a B.S. in mechanical engineering and a B.S. in electrical and computer engineering from the University of Rochester in 2007. He also received a Masters degree in mechanical engineering with a focus in robotics and control from The Johns Hopkins University in 2012. Mr. Kleissas' research interests span computer vision, machine learning, robotics, and connectomics.

ROBERT BAMBERGER is a principal staff engineer at The Johns Hopkins University Applied Physics Laboratory in Laurel, MD. He has over 30 years experience in T&E of military communications, avionics, and directed energy systems. Mr. Bamberger has been working with unmanned and autonomous systems at APL since 2001, primarily in the areas of communications, platform integration, architecture development, sensors, and Test & Evaluation. He has authored/co-authored over a dozen papers on unmanned systems and presented this work at numerous conferences.

Safe Testing of Autonomous Systems Performance

David Scheidt, Robert Lutz, William D'Amico, Dean Kleissas, Robert Chalmers, Robert Bamberger,
Johns Hopkins University Applied Physics Laboratory
11100 Johns Hopkins Road, Laurel, MD
david.scheidt@jhuapl.edu, robert.lutz@jhuapl.edu, william.d'amico@jhuapl.edu, dean.kleissas@jhuapl.edu,
robert.chalmers@jhuapl.edu, robert.bamberger@jhuapl.edu

MOTIVATION

Testing a mature military platform with upgraded technologies and capabilities is not difficult. A trusted platform with a new major sub-system (power plant, sensor modality, etc.) can be evaluated against a new set of Key *Performance Parameters* (KPP) and *Key System Attributes* (KSA). The standard use of *Design of Experiment* (DOE) methods can reduce a large set of seemingly uncorrelated independent variables into an efficient test plan. However, for a trusted platform where behaviors have moved from "highly automated" to "autonomous," the independent variables are poorly defined and possibly unknown. One very challenging scenario will occur when the operator is not within direct command/control (C2) of the platform. Operating without a C2 link means that the autonomous system will independently determine a "worldview" using organic sensors. That worldview will change due to environmental conditions, sensor fidelity, other platform actions (friendly or unfriendly that are within range of the organic sensors), etc. The response of the autonomous system is not deterministic and can only be understood by observing the interplay between the decisions of the various actors. This represents a new class of non-deterministic test planning, execution, and evaluation. This paper describes a "test execution architecture" allowing LIVE/SAFE behaviors for platforms with autonomous behavior.

AUTONOMY TESTING PROCESSES AND GAPS

System test and evaluation is typically dictated by requirements that define the desired system response for all conditions. Requirements-driven design can be problematic for autonomous systems because the size of the condition-response matrix is intractably large, preventing test engineers from fully enumerating system requirements. The Defense Science Board's 2012 report on autonomous systems (DSB 2012) distinguishes between automatic systems, which respond to stimuli with a designed response, and autonomous systems, which use a designed approach to solving problems to determine system responses at run time, a control technique that is itself antithetical to an a priori system response matrix. The appropriate metrics for autonomous systems are not whether or not the system took a specific action, but the impact that action had on operational objectives. Operational objectives are defined as a set of goals, and constraints assigned to the system by the human operator either before or during a mission (Scheidt 2014). The impact autonomous actions have on operational objectives cannot be understood unless testing examines the responses to autonomous system actions by decision-makers outside of the autonomous systems (e.g., adversaries) *for all possible outside decision-maker actions* (Zurek 1990). The testing of autonomous systems is further complicated because human operators define objectives in real-time, which introduces an additional complexity. The variables that influence autonomous system decisions are not only complex, but interdependent and the operational effects of autonomous decisions can be emergent properties of the interaction between the autonomous system and the complex interactive world in which the system operates. The combination of complex, interdependent and emergent properties found in autonomous systems makes it infeasible for us to rely solely on hardware in-the-loop testing using design of experiments methods and the test and evaluation community recommends multi-phase T&E processes that include hardware in-the-loop testing as one element of the process.

Christoph Torens, a leading researcher in autonomous systems test and evaluation, proposes a six step test process for testing autonomous air systems (Torens, 2014). The steps in the process are designed to isolate six dimensions that Torens argues are required to comprehensively test an autonomous air system. The six steps in the process are: (i) Formal Methods, (ii) Static Tests, (iii) Unit Tests, (iv) Software in-the-loop, (v) hardware in-the-loop and (vi) Flight testing. The six dimensions that Torens argues must be addressed during testing are: (1) Test Effort, (2) System Under Test, (3) Scenario Complexity, (4) Coverage, (5) Feedback Time, and (6) Automation. The TACE system is designed to support the fourth and fifth steps in Torens' process: hardware in-the-loop and flight testing which, according to Torens, address the test issues of "System under Test" and "Scenario Complexity".

The Test Resource Management Center (TRMC) Unmanned and Autonomous System Test Roadmap (Tenorio, 2010) identified seven gaps that must be filled to support the testing of autonomous systems. These gaps are:

1. Tools must be developed that enable test personnel to predict unmanned and autonomous system behaviors
2. Tools must be developed that emulate mission and environmental complexity with assured safety
3. Tools must be developed that assess unmanned autonomous system effects and capabilities
4. Tools that support the production of autonomous system test protocols and test designs must be developed
5. Test beds and environments suitable for testing autonomous systems must be developed
6. Reference data sets that incorporate ground truth, decision and behavior data must be produced
7. Methods and tools for systemic unmanned systems testing must be developed

The Safe Testing of Autonomy in Complex, Interactive Environments (TACE) Program is a research initiative to develop an advanced test infrastructure that can measure the performance of autonomous systems operating in complex Live-Virtual-Constructive (LVC) environments while ensuring that the autonomous system does not violate range safety policy. The TACE system directly addresses two of TRMC's gaps: (2) emulation of complex missions and environments with assured safety and (3) assessment of unmanned system effects and capabilities. In addition TACE facilitates (5) production of autonomous system test beds and environments and (7) systemic testing of autonomous systems.

The Office of the Secretary of Defense Autonomy Community of Interest Test and Evaluation, Verification and Validation Working Group (TEVV-WG) has identified an autonomy TEVV that mirrors the traditional Systems Engineering "V".

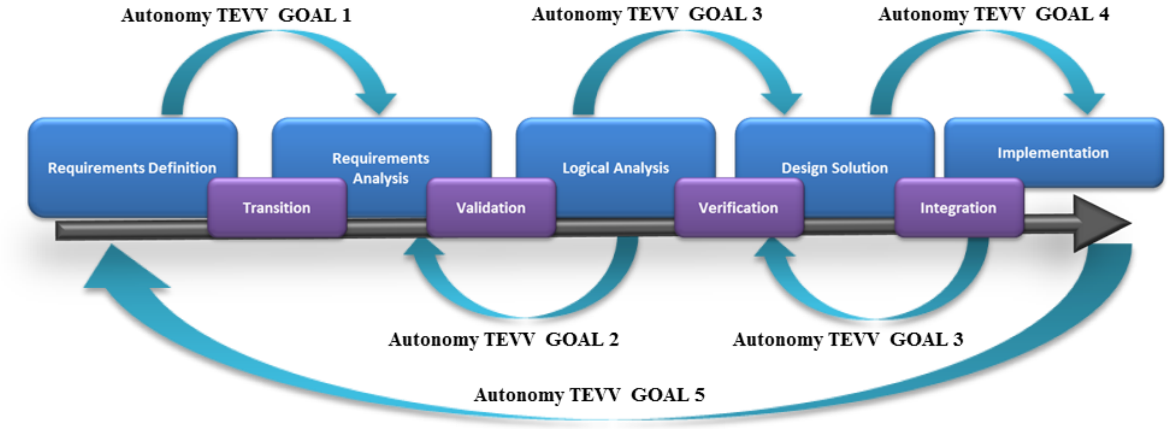

Figure 1. The OSD Autonomy TEVV WG Process

As shown in Figure 1, the autonomy process suggested by the TEVV addresses four identified autonomy TEVV goals which are: (1) Use precise, structured standards to automate requirement evaluation for testability, traceability, and de-confliction, (2) Assurance that appropriate decisions with traceable evidence are used at every level of design to reduce the current T&E burden, (3) Use progressive sequential modeling, simulation, test and evaluation throughout research, develop and operational test and evaluation, (4) Use real time monitoring and just-in-time prediction, to mitigate undesired decisions and behaviors during test and evaluation, (5) Use reusable assurance case based on previous evidence "building blocks" that form a basis for testing. TACE's use of a complex, interactive synthetic environment to stimulate systems under test supports OSD TEVV goal (3) while TACE's ability to provide for assured safe testing supports goal (4).

LIVE VIRTUAL CONTRUCTIVE (LVC) AUTONOMY TESTING

The complete testing sequence for an autonomous system involves test planning, execution, and evaluation. Each of these three steps must interconnect through an LVC simulation of the autonomous system (and the associated environment and actors). An LVC for an autonomous system realistically involves an actual platform (the live part), testers as real people driving simulated entities (the virtual part), and autonomous entities (friendly or adversarial being the constructive part). This nested set of models for platform dynamics and environmental conditions that impact vehicle performance and worldview sensing must be exercised to identify salient test use cases. Once the performance boundaries of the autonomous system are defined, then a carefully selected set of LVC experiments can be proposed through which the system under test (SUT) is brought close to a failure point. The "pre-execution" phase of the LVC may now involve hardware-in-the-loop testing to further refine the fidelity and confidence of the full LVC. The key part of the LVC is that LIVE/SAFE conditions are always maintained. The Watchdog function within the TACE architecture can accept static geospatial constraints as well as dynamic "no-go" regions. Both of these types of constraints can be changed and/or added in real time by the test director or the range safety officer. In addition to geospatial constraints, the Watchdog can implement platform restrictions. If the platform under test is well developed, then a detailed state machine description can be coded to prevent a maneuver that is beyond the operational limits of the hardware. The recorded results from these selected LVC experiments use the same simulation tools can then be compared to the original simulations with actual environmental conditions adjusted to the actual conditions that occurred during the test. The post-test comparison is relatively straightforward if simple "black box" testing – what just happened/pass or fail – is needed – that is already implemented. The more complex case of "white box" testing – why did that behavior occur/root cause analysis – is still under development and represents a significant challenge depending upon the type and level of autonomous behavior(s). The TACE architecture, however, can accept a test plan, execute that test plan, and record the data while maintaining LIVE/SAFE test conditions.

RELATED WORK

Agent-based modeling (ABM) refers to a design strategy whereby discrete entities (i.e., agents) within a system are assigned individual behaviors, and the behaviors and performance of the full system are determined by interactions among the agents. ABM is particularly powerful for modeling large complex systems, as developers can focus on the more tractable problem of modeling the behaviors of individual actors and let the emergent behaviors that result from the interaction of a diverse, autonomous set of agents define how the larger system performs.

ABM is hardly a new idea. In fact, early instantiations of the ABM concept were introduced as early as the 1970s, although more widespread development of ABM applications was not seen until the 1990s (Samuelson, 2005) (Samuelson, Macal, 2006). Due to the increasing complexity of modern test environments and a growing need within the U.S. Department of Defense (DoD) to study the collective performance of multiple interacting autonomous unmanned vehicles in a System-of-Systems (SoS) context, ABM was a very natural fit for the core design strategy that would underlie the TACE system. However, there were several unique requirements associated with the TACE system that introduced some significant challenges for the TACE implementation of ABM:

1) The need to partition agents into on-board components and off-board components. That is, the need to develop agent behaviors for elements of the TACE system that are on-board the System Under Test (SUT) and those that monitor and control SUT behaviors from an off-board Test Base Station (TBS). The communication protocols for agent interactions are different for the on-board and off-board components, and an entirely different communication mechanism was needed for interaction among agents that cross the on-board/off-board barrier. Adjudicating these different methods for communication among agents across the entire test environment required innovative near real-time translation capability and a common data model to ensure semantic compatibility.
2) The need to represent multiple levels of fidelity for a single agent. In particular, the need to simulate the on-board SUT sensor systems, but at low fidelity at the TBS and at higher fidelity on the SUT itself. The desire was to reduce bandwidth requirements (between the SUT and TBS) and the processing load for the sensor simulation on the SUT by filtering out undetectable entities at the TBS and send only potentially detectable entities to the SUT for processing.
3) The need to support interactions between live entities and virtual/constructive agents in the same test environment. This requires near real-time communication between live and simulated actors and a means to translate data model elements into the protocols needed to bridge the gap between live and simulated.

TACE DESCRIPTION

The TACE concept was conceived with two core goals in mind. First, there is a need to ensure range safety when testing inherently unpredictable autonomous systems on live ranges. Second, there is a need to accurately replicate the complexity of the operational environments that a SUT will encounter when executing its assigned missions. This latter need implies the need to integrate an appropriate set of LVC resources into the test environment that can interact with the SUT and stimulate the on-board autonomy so that the SUT behavior and performance can be properly evaluated. The TACE architecture is shown in Figure 2.

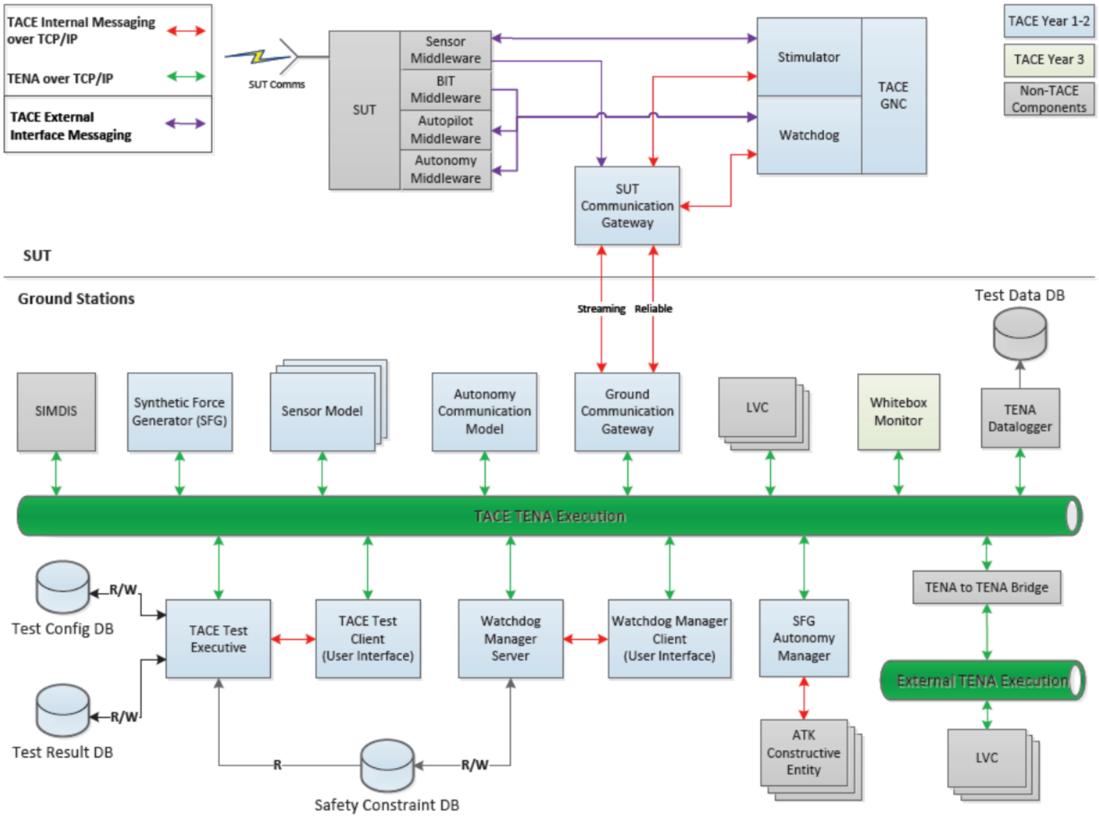

Figure 2. The TACE Architecture

The TACE system is partitioned into an on-board subsystem for the SUT and an off-board TBS. Off-board services are provided to test personnel through Test and Training Enabling Architecture (TENA) protocols (Scheidt, D'Amico, Lutz, 2014). The key components for the on-board system include the *Watchdog* and the *Stimulator*. The Stimulator provides high-fidelity modeling of SUT sensors so that the SUT is aware of detectable LVC entities that can potentially stimulate its autonomy. The Watchdog is designed to observe the actions of the autonomous SUT, determine in real-time whether any range safety or platform constraints have been violated, and issue commands to the SUT autopilot to remediate the situation if a violation occurs. The identification of range safety constraints are defined by the Test Manager prior to the test and the remediations for each constraint violation are defined according to the capabilities of a multi-level state machine application (called Executable Specifications) that was originally developed for NASA spacecraft. Communication of Watchdog commands to the SUT autonomy occurs through a thin client component that is specifically tailored to the SUT software/hardware architecture.

The TBS contains all of the hardware and software components necessary to conduct safe range testing of autonomous systems. The *Synthetic Forces Generator* (SFG) provides the representation of the virtual and constructive entities in the SUT's operational environment. For cases where it is desired that these virtual or constructive entities exhibit autonomous behaviors themselves, a *SFG Autonomy Manager* is provided that can interface autonomy engines directly to synthetic players. Bridges are also provided to link external entities (i.e., live players, virtual cockpits) into the operational environment representation. For completeness, TACE also provides both *sensor models* and *communication models* to ensure that all operational capabilities are properly represented in the synthetic forces. As the test progresses, the state of the SUT and all LVC entities is continually communicated over a wireless network to the Test Manager via the *TACE Test Client*. In addition, as the TACE Watchdog monitors the test for potential range safety violations, the state of the Watchdog is communicated over the same wireless network to the Range Safety Manager via the *Watchdog Manager Client*. Mechanisms to capture real-time data for both *blackbox* (i.e., what did the autonomy do?) and *whitebox* (i.e., why did the autonomy do what it did?) assessment of the SUT autonomy engine are also provided. The Navy's 3-D Analysis and Display Toolset *SIMDIS* can be used to provide additional displays of TACE testing for local and remote users.

Software Agent Architecture

Testing of autonomous systems requires that the interaction of the SUT's decisions and other decision-makers in the outside world be properly evaluated. An accurate representation of the interactions between the SUT and live, virtual and constructive forces included in the test requires the test infrastructure to model the decisions made by each "actor" during an engagement. Each actor must understand what is known, when it is known, and where it is known – particularly when evaluating multi-vehicle autonomous systems where one SUT may need to be evaluated while cooperating with live or simulated peers – but even for single systems, how they interact with surrounding friendly personnel and systems is critical to their function. Each synthetic actor in the test is represented by a cognitive agent. The agent interacts directly with its companion avatar within the synthetic forces generator, which in the current TACE build is the US Naval Air Systems Command's Joint Integrated Mission Module (JIMM). As shown in Figure 3, four key components use used to constitute an actor agent: (1) throughout the test a cognitive model produces decisions based upon the current situation being experienced by the actor, (2) a world model hold *beliefs* that represent the sum total of what is known by the actor at the current time, (3) beliefs can be generated by direct observations through a sensor model which interprets what sensors organic to the actor would observe in an actual engagement, and (4) beliefs can be acquired from other actors via a communications model which models the exchange of knowledge over telecommunications networks. Figure 3 shows two constructive actors; however, the agent framework can also be used to support virtual actors by displaying beliefs onto a human-computer interface and replacing the cognitive model with a human test subject.

Unlike tele-operated systems which require constant human supervision during use, autonomous systems are suitable for over-the-horizon operations, outside of communications operations. Recognizing this need, TACE's communication and decision models support local, intermittent communications outside of continuously connected communication paths with a command authority.

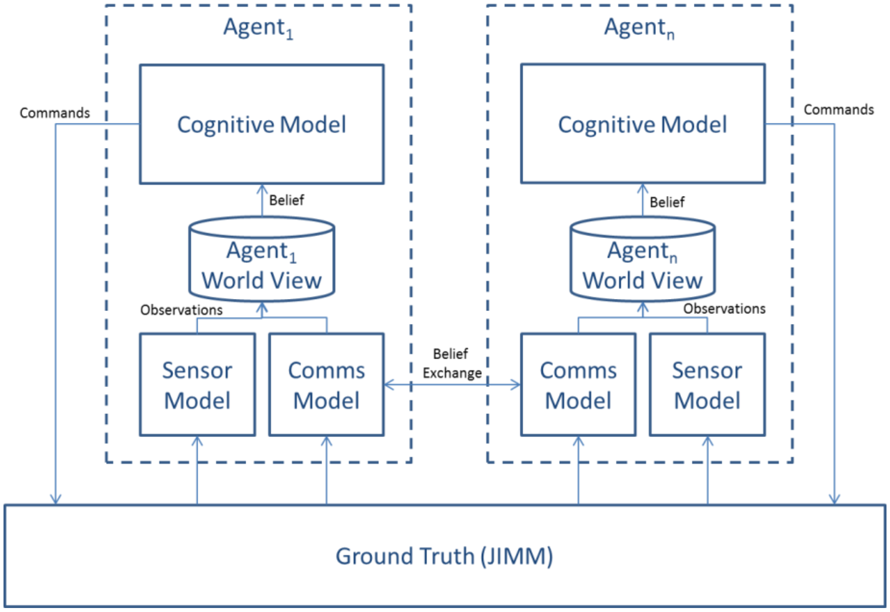

Figure 3. TACE's Multi-Agent Architecture

For development, TACE agents employed a cognitive model based on Dynamic Co-Fields (DCF) algorithm (Scheidt, 2004). DCF is a form or behavioral robotics (Arkin, 1998) that allows an actor engagement in multiple goal-oriented behaviors. The behaviors implemented as part of the TACE evaluation effort included search/patrol, pursuit, threat and collision avoidance, and for the safety watchdog, remediate to loiter. Communications between peer agents are by broadcast. There are neither central command nodes nor any sense of centralized communications routing. DCF agents also perform no predictive planning. Each agent acts in the immediate with only local decision making, though any information gathered from friendly agents (or prior intelligence) may be included in the local decision. The belief fusion process is such that old information is only retained until a newer instance of information about a matching entity or event is identified to replace it. In other words, DCF neither interpolates nor extrapolates from historical data. The nature of these algorithms is such that the decision of any agent at a moment in time can be expressed as a polynomial function of the agent's local state estimate – making them particularly amenable to black box evaluation.

TACE is not restricted to the use of DCF agents, however, and the fused belief DCF packets which would be passed over an open UDP broadcast in TACE, are wrapped as a black-box data payload within a more general TACE data message capable of accommodating any variety of alternate autonomy messaging structures and protocols.

TACE Wireless Gateway

While the TENA architecture provides a mature and robust platform for software communication within a hard-wired network, TENA is not commonly used over a wireless physical layer. As TACE requires reliable wireless communications between the SUT and TACE ground systems we do not use TENA to communicate to the SUT, rather we use a custom wireless gateway for TACE system to mobile autonomous vehicle communications. TACE's LVC capability assumes the existence of a range network infrastructure that is capable of providing reliable, secure physical, link and network layer communications between the SUT and ground systems. TACE employs a custom wireless gateway over these layers that routes TENA traffic over TCP using a fault-tolerant, priority message queuing system, as is illustrated in Figure 4. Note that TACE's Safe Testing capabilities, being on-board the SUT, do not require reliable communications. In fact, loss of communications to the ground due to unreliable range networks is a fault case that TACE's Safe Testing capability is designed to, and does, manage.

When designing the wireless gateway, several key requirements were addressed. First, the gateway must handle problems common with wireless communications such as lost connections, dropped packets, and varying throughput. The gateway must be compatible with the TENA and the TACE system with little impact to existing software. Finally, it also should also have the ability to prioritize TACE safety messages over all other traffic.

Two C++ applications based on the open-source ZeroMQ and Google Protocol Buffer libraries were developed to address these issues. The Ground Wireless Gateway runs on the ground test network and is connected to the TACE TENA execution, while the SUT Wireless Gateway runs on every mobile SUT.

The gateway is designed to work with the TACE infrastructure, seamlessly collecting and relaying all message traffic between the ground and SUT. The gateway applications maintain essentially three messages queues that transmit in priority order, with safety messages first, streaming sensor and telemetry data second, and gateway status last. The queues guarantee message delivery, and cache messages locally in the event of throughput issues or a loss of connection, and immediately resuming transmission once a connection can be reestablished. This ensures that even in the scenario where sensor and telemetry message volume is greater than the transmission rate, a safety message (e.g. emergency stop command) will at most have to wait for a single telemetry or sensor message to be completed before it is handled and relayed.

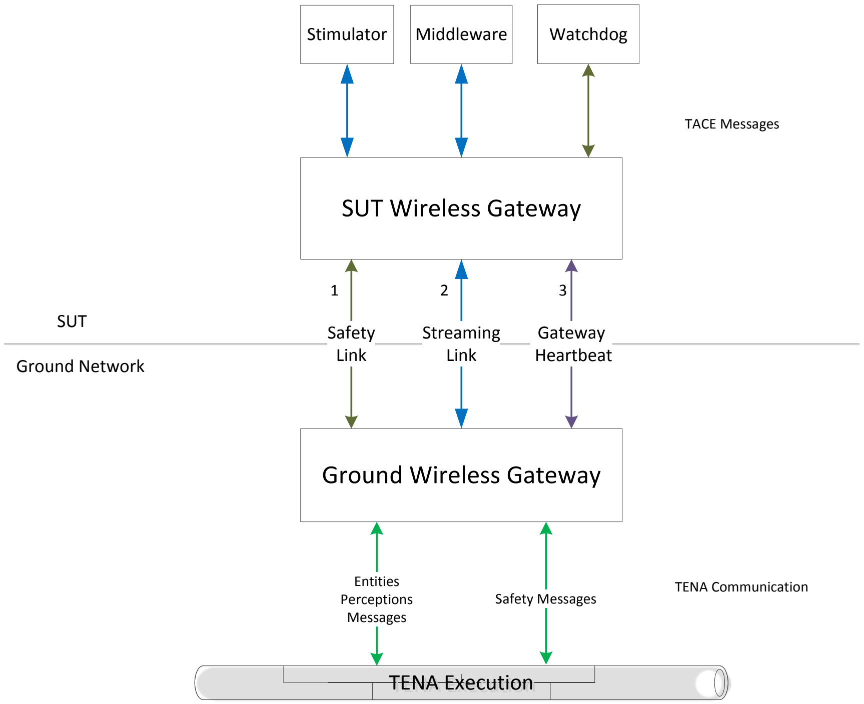

Figure 4. The TACE Wireless Gateway

TACE TESTING WITH AN AUTONOMOUS UNMANNED VEHICLE

TACE development is being performed in three unique phases, where the success criteria of each phase are to achieve a higher Technology Readiness Level (TRL). Modern agile software development practices (specifically, the Scrum methodology) have been employed throughout TACE development. As each Scrum sprint have produced a new layer of functionality, standard software test procedures have been exercised in a laboratory environment to verify the implementation of the individual functions as well as the integration of the functions into a unified system. Once laboratory testing produce positive results, the integrated TACE system is taken into the field to validate that all system-level requirements could be met on a live test range. The U.S. Army's Aberdeen Test Center (ATC) served as the location of all TACE live test activities to date. An operational view of the TACE Phase 1 test environment is provided in Figure 5.

The hardware used to support the TACE Phase 1 testing included the following:

- Procerus Unicorn UAV as the SUT configured with a Gumstix processor board, 2.4GHz Wave Relay device, Procerus Kestrel autopilot, and 900MHz Microhard wireless modem
- TACE ground station (laptop) configured with a 2.4GHz Wave Relay device
- Various TACE development platforms (GPU, desktops)
- Procerus Virtual Cockpit (VC) ground station (laptop) configured with a 900MHz Procerus Commbox
- Radio Controlled (R/C) rotorcraft with Ardupilot, GPS receiver, and 430MHz 3DR wireless module
- Futaba R/C controller
- Ground vehicle with Ardupilot®, GPS receiver, and 915MHz 3DR wireless module
- Laptop with separate 430MHz and 915MHz 3DR wireless modules

The TACE Phase 2 test environment was nearly identical to the Phase 1 environment, with a 3DR Aero fixed wing unmanned vehicle replacing the rotorcraft. The Joint Integrated Mission Model (JIMM) was the primary SFG for both Phase 1 and Phase 2. In addition, laboratory testing of the TACE system integrated with a Scan Eagle payload was performed in Phase 2 to demonstrate that TACE functionality can be successfully hosted on other unmanned vehicles.

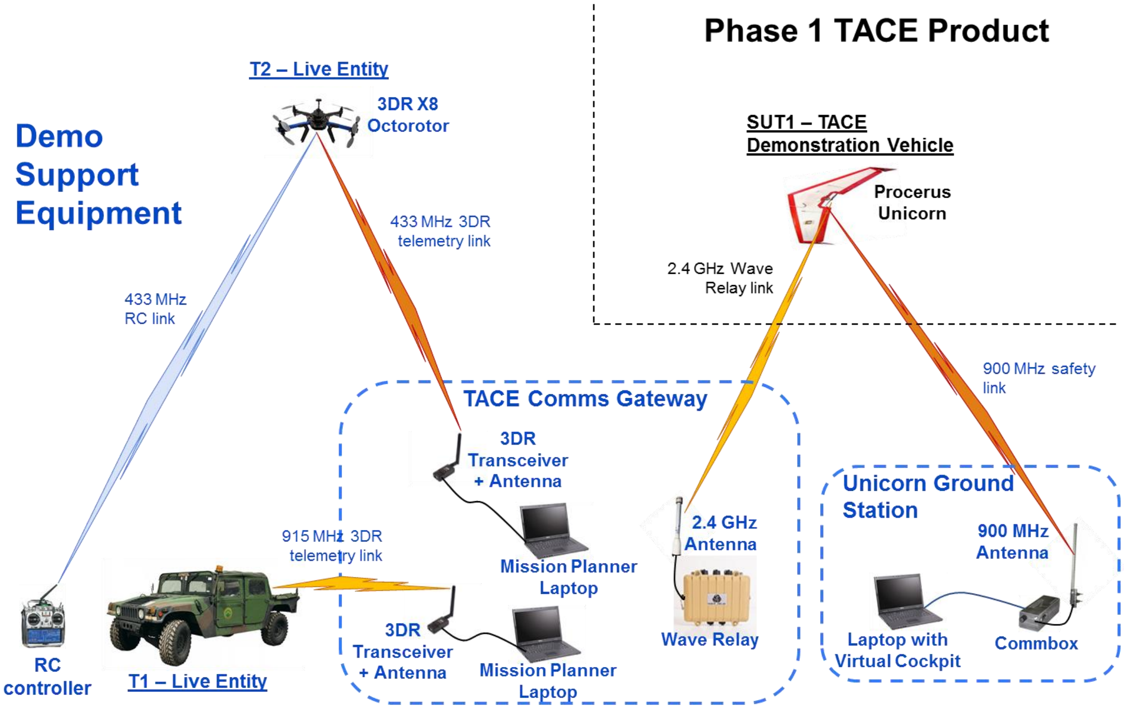

Figure 5. TACE Test Environment

To support both Phase 1 and Phase 2 testing, a sequence of vignettes was developed for the purpose of stimulating the SUT autonomy to violate defined range safety constraints so that the ability of the TACE Watchdog to properly remediate these violations could be studied. In Phase 1, this included geo-spatial constraints around "Go" and "No-Go" areas of the range and proximity constraints with regard to a desired minimum range from other live vehicles. Constraints related to the platform itself (i.e., a maximum bank angle that could not be exceeded) and the ability of ground controllers to take positive control of the SUT at any time where also examined in TACE Phase 1. Figure 6 provides an illustration of one of the proximity constraint vignettes as executed at ATC. In Figure 6 TACE telemetry shows an SUT (blue dot) responding to a proximity constraint violation by safely avoiding test assets (red dots) while interacting with synthetic assets (yellow dot).

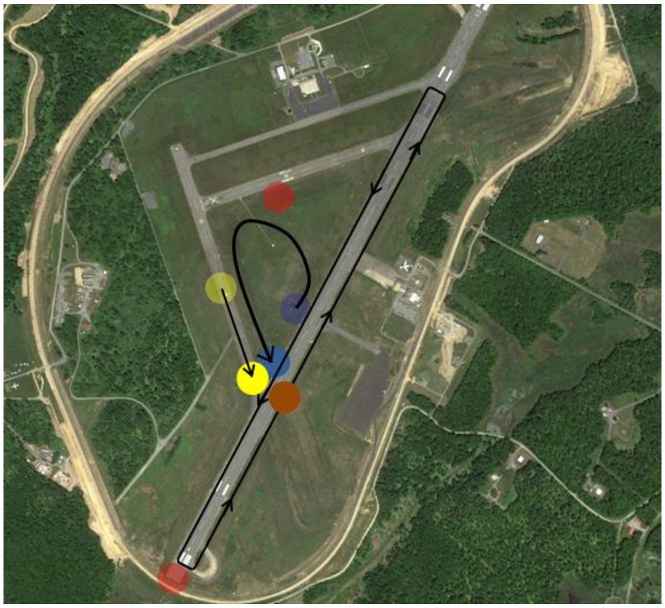

Figure 6. TACE Telemetry from a Safe Testing Experiment

Phase 2 mainly focused on increasing the complexity of the LVC test environment, improving the intelligence of the behaviors associated with synthetic forces, refining TACE capabilities developed during Phase 1, and extending Phase 1 with new entirely new capabilities (i.e., blackbox testing). The requirements associated with each unique TACE Phase 1 or Phase 2 capability, along with the detailed test procedures and success criteria aligned with each requirement, were captured in a test plan prior to each flight test. The successful execution of these test plans was captured in a final report ("TACE Phase 1 Final Report", 2014), which fully documented the tests that was performed and the results across the entire flight test program.

NEXT STEPS

Testing of hardware autonomous systems with TACE is considered a key step in the testing and evaluation of autonomous unmanned systems. JHU/APL is currently engaged in extended TACE by both maturing TACE to technology readiness level (TRL) 6 during 2015/2016 and to to TRL 8 by 2018 by incorporating new capability into the TACE system and by developing and integrating additional TEVV tools that complement TACE. TRL 6 TACE will be demonstrated testing a military-grade unmanned air vehicle on a DoD test range in Phase 3. A significant new capability that will be integrated into TACE in 2015 will be the addition of a “white box” monitoring capability that provides test range personnel with an ability to monitor the underlying motivation of autonomous SUT decisions during a test. Synergistic tools that are under development that will integrate with TACE to constitute a comprehensive autonomy TEVV toolkit include: (a) a software in-the-loop simulation-based test environment that produces hardware in-the-loop test plans as an output, (b) analysis tools that ingest test results from TACE to produce an assessment of autonomous system capabilities, vulnerabilities and risks, (c) high fidelity, complex, interactive reference environments that provide a basis for autonomous system regression testing and (d) formal methods tools that produce provable assertions that form a basis for TACE’s assured safe test capability.

CONCLUSION

Complex cognitive interactions between the decision-making apparatus on-board autonomous systems and decision-making actors *in the wild* presents a unique set of challenges for test range personnel that are tasked with testing and evaluating autonomous systems. This paper describes a novel test infrastructure that directly addresses two recognized gaps that must be addressed in order to test autonomous systems: first, TACE provides guaranteed safe testing by monitoring the autonomous SUT and over-riding autonomy commands that could cause the SUT to perform an act deemed by range personnel as unsafe; second, TACE provides a complex, interactive LVC environment that allows test personnel to examine how SUT interact in complex worlds populated with unpredictable actors that constantly observe, evaluate and respond to SUT actions. To provide these capabilities TACE employs a novel three part system that includes (1) a sophisticated ground infrastructure, (2) an on-board test subsystem and (3) a real-time network link that is used to connect the ground system to the on-board subsystem. Live flight experiments conducted at the Aberdeen Proving Grounds showed that TACE’s on-board component is capable of preventing autonomous SUTs from making decisions that could results in actions that test personnel have characterized as unsafe. Flight tests also showed that the on-board component is capable of injecting synthetic data into the SUT autonomy subsystem that accurately stimulates the SUT decision processes by depicting a realistic environment in which virtual and constructive actors observe each other and, in real-time, make responsive make realistic responses to the actions of adversaries and peers alike. TACE’s real time network link supports safe testing and LVC engagements by shedding low-priority packets as necessary to guarantee that higher priority packets are transmitted in a timely manner. Hardware in-the-loop testing showed that timely delivery of safety critical data could be assured by purposely dropping lower priority packets and also showed that TACE’s packet shedding strategy successfully maintains synchronicity amongst distributed LVC actors in lossy conditions. Hardware in-the-loop testing showed that the TACE ground systems provide range safety personnel and test managers with an ability to define test parameters and safety constraints before and during autonomous SUT tests. Live flight exercises of an autonomous vehicles were conducted during which the TACE ground systems enabled range safety personnel to monitor SUT operating status irrespective of SUT interactions with virtual and constructive stimulants; simultaneously test personnel were able to use the TACE ground system to assess SUT performance by examining SUT responses to realistic, complex, interactive LVC engagements TACE presented to the SUT. TACE’s twin capabilities of stimulating autonomous systems with realistic, complex, interactive LVC environments and providing for guaranteed safe testing, both of which were demonstrated in live flight tests, address keys gaps in existing test community infrastructure. By addressing these gaps TACE provides the test community with the necessary tools to provide the confidence and trust in system performance necessary for flight certification of unmanned autonomous vehicles.

ACKNOWLEDGEMENT

The Test Resource Management Center’s Test & Evaluation/Science & Technology program has established the Unmanned and Autonomous System Test (UAST) test technology area. Mr. Vernon Panei and Ms. Stephanie Riddle (NAS Patuxent River, MD) are the UAST Executing Agent and Deputy Executing Agent, and they have funded the TACE project, which consists of a JHU/APL diverse and highly talented technical team that includes

Brendan John, Kristine Ramachandran, Michael Biggins, Trideum, the Boeing Corporation, and John Wiley at the Army Test and Evaluation Command.

REFERENCES

Arkin R., *Behavior-Based Robotics*, MIT Press, 1998.

Kamiski, P (2012). Task Force Report: The Role of Autonomy in DoD Systems, *Office of Secretary of Defense (OSD) Defense Science Board (DSB)*.

Samuelson D. (2005). Agents of Change, *OR/MS Today*, February issue.

Samuelson D. (2006). Macal, C., Agent-Based Modeling Comes of Age, *OR/MS Today,* August issue.

Scheidt, D., Neighoff, T., Bamberger, R. Chalmers, R. (2004). Cooperating Unmanned Vehicles. *AIAA 3rd "Unmanned Unlimited" Technical Conference*. Chicago, IL.

Scheidt D., D'Amico W., Lutz R., (2014). Safe Testing of Autonomy in Complex, Interactive Environments (TACE), *The International Test and Evaluation Association (ITEA) Journal*, Vol. 35-4, pp. 323-331.

Scheidt, D., et al (2014). "TACE Phase 1 Final Report", The Johns Hopkins University Applied Physics Laboratory, REDD-2014-104.

Scheidt, D. (2014). Unmanned Air Vehicle Command and Control, *Handbook of Unmanned Air Vehicles*, Springer-Verlag.

Tenorio T. (2010). "Test Resource Management Center, Unmanned and Autonomous System Test Roadmap (draft)", *INCOSE Enchantment Chapter Meeting*, San Diego, CA, July mtg.

Torens, C., Adolf, F. (2014), "V&V of Automated mission planning for UAS", *SCI-274 Workshop Verification and Validation of Autonomous Systems*, Imperial College, London, June 24-25.

Zurek W. H., (1990). Algorithmic Information Content, Church-Turing Thesis, Physical Entropy and Maxwell's Demon, *Complexity, Entropy and the Physics of Information*, Sante Fe Press.