AFRL-RH-WP-TR-2018-0005

Realizing Autonomy via Intelligent Adaptive Hybrid Control: Adaptable Autonomy for Achieving UxV RSTA Team Decision Superiority (also known as Intelligent Multi-UxV Planner with Adaptive Collaborative/Control Technologies (IMPACT))

Mark Draper, Allen Rowe, Scott Douglass, Gloria Calhoun, Sarah Spriggs, Derek Kingston, Laura Humphrey, Michael Patzek, Jessica Bartik
Air Force Research Laboratory, Wright-Patterson AFB, OH 45433

Doug Lange, John Reeder, Bryan Croft
Space &Naval Warfare Systems Command, San Diego, CA 92152

Jessie Y.C. Chen, Michael Barnes
Army Research Laboratory, Orlando, FL 32826

Greg Trafton, Thomas Apker
Navy Research Laboratory, Washington, DC 92152

Kyle Behymer, Heath Ruff, Clayton Rothwell
Infoscitex Corporation, Dayton, OH 45431

Dakota Evans
University of Dayton, Dayton, OH 45469

Elisabeth Frost
Wright State Research Institute, Dayton OH 45435

January 2018
Interim Report

AIR FORCE RESEARCH LABORATORY
711 HUMAN PERFORMANCE WING,
AIRMAN SYSTEMS DIRECTORATE,
WRIGHT-PATTERSON AIR FORCE BASE, OH 45433
AIR FORCE MATERIEL COMMAND
UNITED STATES AIR FORCE

Notice and Signature Page

AFRL-RH-WP-TR-2018-0005 HAS BEEN REVIEWED AND IS APPROVED FOR PUBLICATION IN ACCORDANCE WITH ASSIGNED DISTRIBUTION STATEMENT.

ROWE.ALLEN.J.1259846044
Digitally signed by ROWE.ALLEN.J.1259846044
Date: 2018.03.15 09:45:30 -04'00'

ALLEN ROWE
Project Engineer
Supervisory Control and Cognition Branch

NALEPKA.JOSEPH.P.1230146710
Digitally signed by NALEPKA.JOSEPH.P.1230146710
Date: 2018.03.22 11:19:11 -04'00'

JOSEPH P. NALEPKA
Chief, Supervisory Control and Cognition Branch
Warfighter Interface Division

SMITH RODRIGUEZ.KRISTOFFER.A.1239709865
Digitally signed by SMITH RODRIGUEZ.KRISTOFFER.A.1239709865
Date: 2018.03.23 10:42:33 -04'00'

KRISTOFFER A. SMITH-RODRIGUEZ, LTCOL, USAF
Chief, Warfighter Interface Division
Airman Systems Directorate
711 Human Performance Wing

REPORT DOCUMENTATION PAGE

Form Approved
OMB NO. 0704-0188

1. REPORT DATE *(DD-MM-YY)*	**2. REPORT TYPE**	**3. DATES COVERED *(From – To)***
30 January 2018	Interim	09-01-13 – 07-11-17

4. TITLE AND SUBTITLE
Realizing Autonomy via Intelligent Hybrid Control: Adaptable Autonomy for Achieving UxV RSTA Team Decision Superiority (also known as Intelligent Multi-UxV Planner with Adaptive Collaborative/Control Technologies (IMPACT))

5a. CONTRACT NUMBER
FA8650-14-D-6500/0006

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER
62202F

6. AUTHOR(S)
Mark Draper, Allen Rowe, Scott Douglass, Gloria Calhoun, Sarah Spriggs, Derek Kingston, Laura Humphrey, Michael Patzek, Jessica Bartik (Air Force Research Laboratory), Doug Lange, John Reeder, Bryan Croft (Space and Naval Warfare Systems Command), Jessie Y.C. Chen, Michael Barnes (Army Research Laboratory), Greg Trafton, Thomas Apker (Navy Research Laboratory), Kyle Behymer, Heath Ruff, Clayton Rothwell (Infoscitex Corporation), Dakota Evans (University of Dayton), Elizabeth Frost (Wright State Research Institute)

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER
H0KJ (53290906)

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
711 HPW/RHCI
2210 Eight Street
Wright-Patterson Air Force Base, OH 4533-7511

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)
711th Human Performance Wing
Airman Systems Directorate
Air Force Research Laboratory
Air Force Materiel Command
Wright-Patterson Air Force Base, OH 45433-7334

10. SPONSORING/MONITORING AGENCY ACRONYM(S)
ASD(R&E)

11. SPONSORING/MONITORING AGENCY REPORT NUMBER(S)
AFRL-RH-WP-TR-2018-0005

12. DISTRIBUTION/AVAILABILITY STATEMENT
DISTRIBUTION STATEMENT A. Approved for public release: Distribution is unlimited.

13. SUPPLEMENTARY NOTES
This is an Interim Report for Work Unit 53290906, and documents effort performed for Autonomy Research Pilot Initiative. 88ABW Cleared 02/22/2018; 88BW-2018-0820.

14. ABSTRACT (Maximum 200 words)
Agility in tactical decision-making and mission management is a key attribute for enabling teams of heterogeneous unmanned vehicles (UxV) to successfully manage the "fog of war" with its inherently complex, ambiguous, and time-challenging conditions. This agility requires effective operator-autonomy teaming including the achievement of trusted collaboration and the flexible, high-level tasking required for team task sharing and decision superiority. A tri-service team has conducted Assistant Secretary of Defense for Research and Engineering (ASD/R&E) sponsored research focused on instantiating an "Intelligent Multi-UxV Planner with Adaptive Collaborative/Control Technologies" (IMPACT) by combining flexible play calling for task delegation, bi-directional human-autonomy interaction, advanced cooperative control algorithms, intelligent agent reasoning, and autonomic technologies to enable effective single operator command and control (C2) of cooperative multi-UxV missions. IMPACT operators, with intelligent agent assistance, were able to task and manage simulated 12 UxV (4 air, 4 ground, and 4 sea surface unmanned vehicles) in response to unexpected events that arose during base perimeter defense missions.

15. SUBJECT TERMS
Unmanned Systems, Supervisory Control, Intelligent Agents, Human Autonomy Interface, Cooperative Control Algorithms, Human Autonomy Teaming, Interface Design, Autonomics Framework, Artificial Intelligence, Command and Control

16. SECURITY CLASSIFICATION OF:			**17. LIMITATION OF ABSTRACT**	**18. NUMBER OF PAGES**	**19a. NAME OF RESPONSIBLE PERSON**
a. REPORT	**b. ABSTRACT**	**c. THIS PAGE**			Allen Rowe
					19b. TELEPHONE NUMBER *(Include Area Code)*
Unclassified	Unclassified	Unclassified	SAR	108	

Table of Contents

List of Figures

List of Tables

EXECUTIVE SUMMARY

Agility in tactical decision-making and mission management is a key attribute for enabling teams of heterogeneous unmanned vehicles (UxV) to successfully manage the "fog of war" with its inherently complex, ambiguous, and time-challenging conditions. This agility requires effective operator-autonomy teaming including the achievement of trusted collaboration and the flexible, high-level tasking required for team task sharing and decision superiority. A tri-service team has conducted Assistant Secretary of Defense for Research and Engineering (ASD/R&E)-sponsored research focused on instantiating an "Intelligent Multi-UxV Planner with Adaptive Collaborative/Control Technologies" (IMPACT) by combining flexible play calling for task delegation, bi-directional human-autonomy interaction, advanced cooperative control algorithms, intelligent agent reasoning, and autonomic technologies to enable effective single operator command and control (C2) of cooperative multi-UxV missions (Figure 1). IMPACT operators, with intelligent assistance, were able to task and manage a total of 12 UxV (4 air, 4 ground, and 4 sea surface vehicles) in response to several unexpected events that arose during simulated ongoing base perimeter defense missions. This executive summary provides a brief introduction to the main features of the IMPACT system, while the rest of this report provides detailed descriptions of all research aspects associated with this project.

Figure 1: IMPACT Control Station Prototype

Interfaces for Operator-Autonomy Teaming

IMPACT's displays and controls (Figure 2) feature video game inspired pictorial icons that present information in a concise, integrated manner to facilitate retrieval of the states/goals/progress for multiple systems and support direct perception and manipulation principles. Multi-modal controls (speech, touch, and mouse) augment a "playbook" delegation architecture and enable seamless transition between control states (from manual to fully autonomous). With this adaptable automation scheme, the operator retains authority and decision-making responsibilities that help avoid "automation surprises" (Calhoun, Ruff, Behymer, & Frost, 2017). By supporting a range of interactions, flexible operator-autonomy teamwork enables agility while responding to a dynamic mission environment. At one extreme, the operator can manually control UxV movement or build plays from the ground up, specifying detailed parameters. At the other extreme, the operator can quickly task one or more UxVs by only specifying play type and location with an intelligent agent determining all other parameters. For example, when an IMPACT operator calls a play to achieve air surveillance on a building, the intelligent agent recommends a UxV to use (based on estimated time enroute (ETE), fuel use, environmental conditions, etc.), a cooperative control algorithm provides the shortest route to get to the building (taking into account no-fly zones, etc.), and an autonomics framework monitors the play's ongoing status (e.g., alerting if the UxV won't arrive at the building on time). IMPACT's play calling interfaces also facilitate operator-agent communication on mission details to optimize play parameters (e.g., target size and current visibility) as well as supporting operator/autonomy shared awareness (e.g., illustrated by a display showing the tradeoffs associated with multiple agent-generated courses of actions across mission parameters). Play progress is depicted in a matrix display reflecting autonomics monitoring and a tabular interface aids play management (e.g., allocation of assets across plays). Additional detail on all the play-related interfaces is available (Calhoun, Ruff, Behymer, & Mersch, 2017).

Figure 2: IMPACT Operator-Autonomy Interfaces

Intelligent Agent Framework for Course of Action Generation

UxV allocation, tasking, and management capabilities were provided in IMPACT via an intelligent agent that was developed using the Cognitively Enhanced Complex Event Processing (CECEP) framework. This capability allows for an operator to communicate high-level details about a desired play call such as location or optimization criteria (e.g., time, fuel). In response, the agent provides the operator with a ranked set of courses of action (COAs) that were formulated based on low-level task details. This approach was expected to alleviate workload burden of the operator by having the autonomy focus on the low-level details while allowing the operator to tend to higher level mission objections.

CECEP is a complex event processing framework with extended procedural and domain knowledge aspects. Agents that use procedural knowledge were developed using a discrete finite state machine called behavior models that include states and transitions between states that are guarded by patterns. A pattern language called Esper was used to match complex patterns of operator and UxV behaviors to transition states at runtime. Behavior models were used to produce behaviors (e.g., feedback for the operator or UxV play execution). Agents that use domain knowledge were developed using cognitive domain ontologies (CDOs). A CDO is a rooted tree structure with features that are connected via relations. CDOs can be processed using the artificial intelligence process of constraint satisfaction to produce configurations, possible worlds, or COAs. In IMPACT, CDOs were developed to capture the domain for UxV play calling and produce COAs for play to vehicle(s) assignment.

UxAS Routing Algorithms

Current ground control stations for unmanned vehicles provide relatively low levels of autonomy, e.g. automatically commanding an assigned vehicle to follow a sequence of waypoints generated by a human operator. To increase the level of autonomy of UxVs, the Unmanned Systems Autonomy Services (UxAS) software architecture provides flexible and adaptive automated path planning, sensor steering, and inter-vehicle coordination for unmanned air, ground, and surface vehicles. UxAS consists of a collection of modular services that interact via a common message passing architecture, which makes it easy to add new services. Currently, UxAS provides approximately 50 services that automate vehicle route planning and sensor steering, coordinate behavior between cooperating vehicles, connect with external software and hardware devices, validate mission requests, log and diagram message traffic, and optimize play ordering with respect to total time required or distance traveled.

More specifically, UxAS provides services that automatically generate waypoints and sensor steering commands for search and surveillance plays over points, lines, and areas, with many tunable parameters. UxAS also provides services that generate routes between plays based on vehicle type, e.g. so that ground vehicles stay constrained to roads. In addition to services that plan static routes, UxAS provides services that can update routes and sensor steering commands adaptively online, including for teams of cooperating vehicles. In general, UxAS services plan vehicle routes that account for factors such as regulatory “no-fly zones,” physical boundaries such as roads and terrain, and kinematic vehicle constraints such as minimum turn radius. In IMPACT, the intelligent agent queries UxAS about the cost of routes needed to perform a play and uses the information to help determine which vehicles to assign. The appropriate UxAS services then carry out execution of the play by implementing routing, sensor steering, inter-vehicle coordination, and online adaptation during play execution.

Autonomics and Task Management

Autonomic approaches manage complex systems such that they exhibit self-adaptation in response to demands on the system or degradation of performance. One such autonomics approach is the Rainbow autonomics framework, developed at Carnegie Mellon University (CMU). Rainbow can manage systems that can be described as networks within the network model held within the framework. In IMPACT, the control team made up of humans and autonomous assistants was modeled as a network of servers that work tasks from task queues. Inherent in the autonomics framework are probes, gauges, and strategies. Probes read data from the underlying system, gauges aggregate that data, and strategies manipulate the network to improve performance.

Task manager capability includes the automatic generation of tasks from event information. By reading event information (e.g., from chat messages), the task manager generates tasks and parses out necessary information to aid in the completion of the task. Task guided human machine interfaces (HMI) help users complete tasks (e.g., calling plays) by making calls to appropriate tools within IMPACT and prepopulating with data from the events. Tasks can be directed towards autonomous assistants that are capable of completing some tasks, and queue management tools are provided to the operator.

Fusion and the Distributed Architecture and Services

Fusion is a software framework that enables natural human interaction with flexible and adaptable automation. A distributed service oriented architecture is employed that is composed of multiple disparate systems, unified representationally through negotiated communications protocols and physically through a common communications hub. The decentralization of the architecture enables logging, monitoring, and substitution of components with minimal effect on other components. Thus, several different systems can indirectly interact with one another through a publish/subscribe hub to provide a greater service to the user. All connected pieces communicate through a common messaging protocol to send and receive information. Connected services developed for IMPACT include intelligent agent reasoning among disparate domain knowledge sources, autonomics monitoring services, intelligent aids to the operator, cooperative planners, and advanced simulation via instrumented, goal oriented operator interfaces. The distributed architecture along with an extensible software framework enables the system to be expanded for other human-automation research.

The Fusion architecture includes the core (customizable) aspects that are common across applications as well as features that support the IMPACT project. The Fusion test bed also displays the scenario environment, presents mission events that prompt UxV management tasks, provides a workspace for the operator to team with autonomy to complete tasks, and records task performance measures. Other IMPACT specific components provide interfaces for calling and modifying plays, viewing agent generated candidate COAs, and presenting the results of an autonomics service monitoring play progress.

Operator-in-the-Loop Evaluation of Operator-Autonomy Teamwork

A high-fidelity human-in-the-loop simulation evaluation was used to compare the IMPACT prototype to a baseline system that represented the current state-of-the-art at the beginning of the effort. The baseline system included a subset of IMPACT’s capabilities such as the route planner and an associated interface. However, the baseline system lacked agent assistance, plan monitoring, and speech control. The experimental design was a 2 (Baseline, IMPACT) x 2 (low, high mission complexity) within-participant design with the order of conditions blocked by system (half of the participants used IMPACT first, the other half with Baseline) and counterbalanced across task complexity. Mission complexity was manipulated by varying the number and timing of tasks. Each of eight participants (all familiar with base defense and/or unmanned vehicle operations) performed four 60-minute base defense missions. Participants completed a variety of defense mission related tasks involving twelve simulated UxV. Participants’ task performance was better on multiple mission performance metrics with the

IMPACT system in comparison to the baseline system. Participants were also able to execute plays using significantly fewer control inputs with IMPACT as compared to baseline. The overall usability of each system was assessed using the System Usability Scale (SUS; Brooke, 1996). Participants rated IMPACT higher than baseline on all ten SUS items and the overall SUS score was significantly higher with IMPACT than with baseline. Participants also subjectively rated IMPACT significantly better than baseline in terms of its perceived value to future UxV operations as well as its ability to aid workload. In fact, every participant gave IMPACT the highest possible score for potential value, and all but one participant gave IMPACT the highest possible score for its ability to aid workload.

Outcomes and Way Ahead

The IMPACT project produced significant knowledge in a number of areas important to autonomy-related capabilities (see Appendix A for a listing of the many publications generated from this effort). Not only did the project spur advancements in component technology development, model development, and general design understanding/guidance, but much was learned from the integration of key autonomy-related technologies into a single multi-UxV control station application. IMPACT also produced a robust Department of Defense (DoD) "virtual lab" for continued human-autonomy teaming research. This was a key objective of the Autonomy Research Pilot Initiative (ARPI) process. A three station system (C2, Sensor Operator (SO), & Test Operator Console (TOC)) is available for organic wide-spectrum human-autonomy teaming (HAT) evaluations with sites currently at the Air Force Research Lab (AFRL), the Space and Naval Warfare Systems Command (SPAWAR), and the Army Research Lab (ARL). A new vision for future human-autonomy systems was successfully conveyed to DoD senior leadership via many interactive demonstrations of the IMPACT system. This vision clearly illustrates that the human will continue to have a prominent role in interacting with increasingly autonomous technology, dynamically flexing between supervisor, teammate, or manual controller as conditions dictate. Finally, IMPACT technologies have extended/transitioned in a myriad of ways. Other ARPI projects have leveraged IMPACT technology to advance their aims while new DoD projects (including Joint Capability Technology Demonstration (JCTD) support efforts and Defense Advanced Research Projects Agency (DARPA) programs) and several industry contractors now utilize IMPACT in autonomy technology development efforts. Additionally, IMPACT has become the core C2 autonomy piece within the TTCP Autonomy Strategic Challenge which is a 3-year, 5 nation effort to integrate and assess promising allied autonomy capability in mixed live/virtual multi-UxV littoral environments.

The IMPACT project has enabled a deeper exploration into the critical issues that influence flexible and effective human-autonomy collaboration. Although the IMPACT evaluation demonstrated value in several aspects related to operator-autonomy teaming, several deficiencies and gaps in understanding were also identified and improvements are underway. These include research related to novel methods for enabling bi-directional communication and management of temporal constraints, more naturalistic dialogue and sketch interactions, and consideration of information uncertainty in decision-making tasks. Additionally, research is investigating the effects of a decentralized replanning capability, real-time operator functional state assessment, and alternative team structures on overall human-autonomy teaming. The results of these follow-on efforts will provide a much richer understanding of this area.

1 BACKGROUND AND PROJECT OBJECTIVES

Future manned and heterogeneous unmanned forces must be able to work increasingly as agile synchronous teams to complete tactical reconnaissance, surveillance, and target acquisition (RSTA) related missions in complex, ambiguous, and dynamically changing environments. Advanced and highly reliable autonomous behavior and multi-unmanned vehicle (UxV) cooperative control planning algorithms will be required that are far beyond the capability of currently fielded systems. Therefore, rather than a rapid switch from current operations to fully functional autonomous cooperative RSTA teams, the likely transition path will involve incrementally fielding component autonomous behaviors as they are developed, with overall autonomous capability increasing over time. Thus a key challenge is, with the addition of incremental and imperfect autonomous behaviors, how best to ensure flexible, robust mission effectiveness across a wide range of situations and with the many ambiguities associated with the "fog of war".

Mission effectiveness will rely on increased agility: the rapid identification and management of uncertainties that can disrupt or degrade an autonomous team's ability to safely complete complex missions. Agility is especially critical to robust team decision making in highly challenging and rapidly evolving situations. One promising method for increasing agility over the long term is integration of intelligent agent, autonomics, and machine-learning technologies such that autonomous control technology "gets smarter" and thus more resilient over time. This is especially valuable for distributed, platform-centered autonomy. A complimentary method that is potentially far more powerful in the near-term (when communications links are maintained) is to establish an intuitive and effective dialog between the human team member and emerging autonomy. With this method, strengths of each can be maximally utilized to resolve ambiguities and achieve decision superiority, with autonomy being increasingly unleashed as trust in gained in these operations. Many researchers are exploring critical autonomy components (intelligent agents, machine learning, cooperative control planners, human autonomy interfaces, etc.) in isolation. The novelty of this project was that it integrated these approaches to explore, at a systems level, the best mix of adaptive technologies for realizing near-term RSTA team autonomy.

A vision that underlies IMPACT system design is conveyed in Figure 3. A black silhouette of a human operator is positioned in the upper left-hand side of the graphic. This operator is purposely not in the center of the picture, but rather placed toward the top edge of the system, to represent a supervisor that is more often "on the loop" versus continually "in the loop". The operator is managing multiple unmanned assets in a tactical area, and these assets are heterogeneous (air, sea and ground platforms) versus homogeneous platforms. The operator is interfacing with these systems through an advanced graphical interface, using multimodal methods including speech and touch for rapid, intuitive inputs. The operator can "see through" the interface to the environment itself, which again speaks to the need for interface design to allow for transparency into the plans and activities of autonomy. Lastly, machine intelligence is represented by the blue electronic avatar brain in the lower left-hand side of the graphic. This digital assistant is constantly monitoring and reasoning over the platforms, environment, and mission in order to assist the operator in situation assessment, decision making, and action execution. The human and machine intelligence are grouped to the left of the graphic to represent the need for "teaming" and naturalistic interaction between the two.

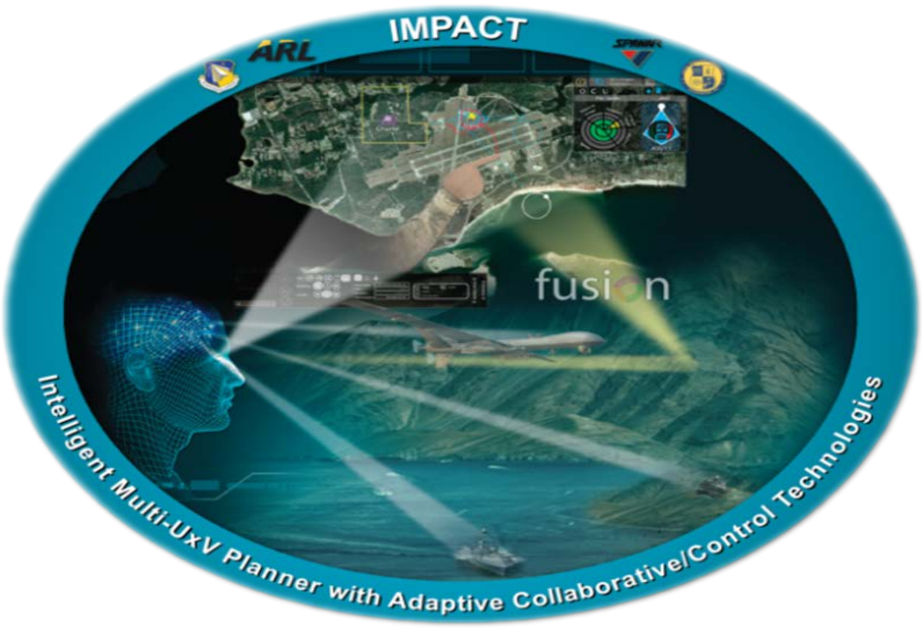


Figure 3: A Vision for Human-Autonomy Teaming

The overall objective of this project was to achieve flexible operational agility and resilience in developing autonomous behavior for UxV RSTA teams. A multidisciplinary, tri-service team developed and evaluated "Intelligent Multi-UxV Planner with Adaptive Collaborative/Control Technologies (IMPACT)". This new human-autonomy teaming capability combined flexible, goal-oriented "play" calling and human-autonomy interaction with intelligent agents and cooperative control algorithms that provided near-optimal task assignment and path planning solutions as well as adaptive/reactive capability. A key principle in the development of IMPACT algorithms was to be transparent, agile, and resilient.

The effort had four major objectives. Each was informed by and leveraged the others throughout the project's timeframe.

1. Increase robustness and transparency of autonomous control by expanding the capabilities of UxV cooperative control planning algorithms and optimization logic.
2. Advance state of the art for developing adaptive and reactive autonomous tactics through intelligent agent and machine learning approaches.
3. Identify and validate intuitive and adaptive interaction methods for human-autonomy dialog and novel displays for transparency into the autonomous behavior.
4. Integrate all component technologies into the IMPACT architecture and evaluate; compare against existing models and current state of the art for RSTA missions.
5. An additional objective was to leave behind a tri-service multi-UxV control station simulation testbed that incorporates IMPACT technology for continued human-autonomy research, development and transition.

Simultaneously developing a multi-UxV controller along with multiple candidate adaptive dynamic planning solutions ensures a thorough exploration of the relative influence and associated interdependencies of these technologies. Emerging IMPACT-related capabilities strive to maximize the robustness of incremental increases of autonomous behavior into RSTA teams, enabling unmanned systems to expand beyond supporting independent, disjointed tasks to more fluid, cooperative and harmonious actions that are goal oriented versus task oriented. This will maximize desired mission effects and assist in achieving decision superiority.

2 IMPACT OVERVIEW AND OVERALL TECHNICAL APPROACH

The IMPACT project was one of seven ARPI projects sponsored by the Office of Secretary of Defense for ASD/R&E. Over a 3-year period, a tri-service team (AFRL, SPAWAR, ARL & Navy Research Lab (NRL)) conducted research, development and integration of multiple autonomy-related technologies to enable single operator management of cooperative multi-UxV missions (Figure 4). Novel operator interfaces were also designed and evaluated to support the operator's ability to continually observe and direct autonomy components.

Figure 4: IMPACT Research Components with Associated Service Lab Contributions

The key to IMPACT was the simultaneous development, integration and assessment of several candidate agility tools to combat many "fog of war" events that can threaten mission success. By increasing both the human's and autonomy's ability to be agile to unexpected change, overall mission effectiveness can potentially be sustained across a wide range of contexts. By studying these technologies and their interactions concurrently, a robust and feasible solution set can be identified for real-world operations. With IMPACT, explorations began towards an effective balance between human/autonomy, global/local mission management, and designed/learning systems that adapts with evolving situations for a base security management application.

The core capability of IMPACT consists of a multi-UxV control station with cooperative control algorithm for tactical mission routing, intelligent agents for reasoning over domain knowledge for asset allocation and determining opportunities for action, an autonomics system to automatically monitor ongoing plans, and a human machine interface to couple human and machine capabilities. The human operator retains a spectrum of control from high level supervisor to manual controller, dependent upon context. It is from this core capability that extensions in platform autonomy, mission set, and environmental context can grow.

In addition to the overall objectives listed above, many detailed research challenges across several disciplines were addressed in the IMPACT project. A partial list is presented below.

- cognitively-based methods for dynamic agent reasoning
- agent-based C2 decision support tools
- flexible, transparent, and reactive cooperative control algorithms
- intuitive interfaces for human management of multiple autonomous assets
- models, methods, and guidelines for achieving agent transparency
- methods to acquire and manage incoming tasking
- use of autonomics for monitoring/managing play execution
- real-time predictive model of human operator automation monitoring
- automated verification and synthesis of mission plans for UxV teams
- machine learning of UxV tactics through human evaluation
- machine learning for task generation

The general approach to technical development was to mix component technology research and development with periodic integration and spiral system testing of the most mature components. First, a tri-service challenge scenario was agreed to. The application chosen was base perimeter defense, as this provided 1) a RSTA environment that is relevant to all DoD services, 2) a realistic challenge scenario for tasking of heterogeneous air/sea/ground unmanned assets, and 3) supports a wide range of possible events to demonstrate agility. Cognitive task analyses were then conducted with subject matter experts to define key tasks, decision points, and information requirements. Next, a service based system architecture and associated play sequencing was derived to underlie testbed development. Throughout, component technology development occurred (Figure 4), with significant cross-talk and increasing integration being promoted as the project matured. Lastly, two spiral evaluations occurred in the project to assess the military utility of the resulting IMPACT system prototype.

3 DETAILED TECHNICAL APPROACH: INTEGRATED SYSTEM COMPONENTS

The technology components that were successfully integrated into the IMPACT system testbed over a three-year period are described below. Note that although the technology components are ordered separately, the key to this project was the understanding gained through the interaction of these technologies within a military mission application. Thus the first component to be discussed is the Fusion Framework, which underlies the entire IMPACT system.

3.1 Fusion Framework

3.1.1 Motivation and Challenges

Robust autonomy-based frameworks enable evaluation of cooperation and coordination among widely disparate platforms such as remotely piloted aircraft (RPAs) and autonomous unmanned systems such as ground, air, and maritime entities. Tying these interactions into an immersive HMI improves evaluation of user behaviors and confidence in a low-risk environment. However, a unique challenge exists in unification of operator interactions, autonomous platforms, and intelligent aids. A common drive is to push towards more autonomy, diminishing the operator's involvement. Operators can provide useful information to autonomous systems, and autonomy can be used to augment operator capabilities, so an alternative is to develop and support symbiosis between the two. This symbiosis can be realized via a robust framework that provides user-tunable accessibility into this autonomy. This enables evaluation of user comfort, trust, and confidence with autonomous components. The associated ability to tune autonomy also drives future requirements for HMI design and accessibility (excerpt from Rowe, 2015).

To address the complexities involved in providing a common environment to explore these motivations and challenges, the Fusion Framework was developed. Fusion is a framework that enables natural human interaction with flexible and adaptive automation. It employs multiple components: intelligent agents that reason among disparate domain knowledge sources (Douglass, 2013); machine learning that provide monitoring services and aids to the operator (Vernacsics, 2013); cooperative planners (Kingston, 2009); and advanced simulation via an instrumented, goal-oriented operator interface (Miller, 2012). These empower experimentation and technology advancement across multiple systems (see Figure 5).

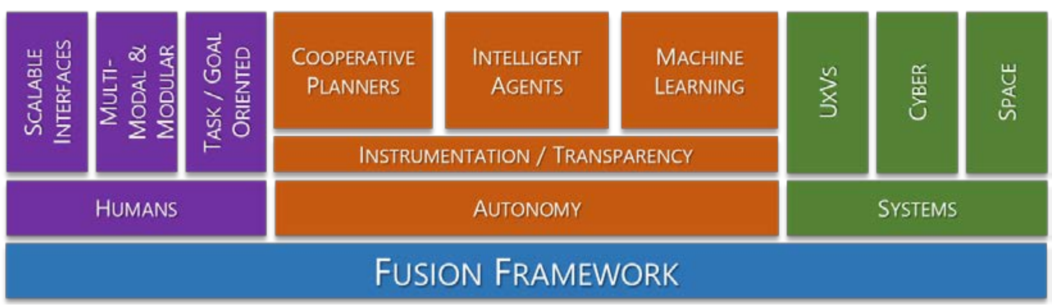


Figure 5: Fusion High Level Framework

3.1.2 Software and Hardware Acquisitions

The Fusion Framework was built with extensibility, maintainability, and commonality at the core fundamental level utilizing state of the art software development tools and processes consisting of the following:

1. Microsoft Windows 10 Operating System
2. Microsoft DotNet version 4.6.1
3. Microsoft Visual Studio 2015
4. Microsoft C# and Managed C++ Programming Languages
5. JetBrains ReSharper Code Analysis Tools
6. JetBrains YouTrack Agile Requirements Management
7. JetBrains TeamCity Build Management System

The hardware consists of high performance computing Microsoft Windows-based platforms. The Fusion Framework utilizes a services oriented architecture enabling components to be distributed across a distributed computing platform. The operator stations require a high-end graphics card such as the nVidia Quadro 4000 or higher series. Representative computing devices include the Dell Precision T7910 series, high-resolution touch screen Liquid Crystal Displays (LCDs) such as the Acer T272HUL Light Emitting Diode (LED) Touchscreen (2560 X 1440) and the Sharp PN-K322B 4K Ultra-HD LCD Touchscreen (3840 x 2160 resolution). To help facilitate development across all associated laboratories, a common hardware setup was procured.

3.1.3 Development and Implementation

3.1.3.1 Distributed Virtual Laboratory

The notion of a virtual distributed laboratory (VDL) connecting various DoD and contractor sites throughout the Continental United States is paramount to foster a more cohesive and distributed development and research environment. Fusion adopted a DoD open source model, enabling joint development across a variety of projects and collaborators, all contributing to a single source repository. The core development team is located at AFRL, and there are currently several offsite laboratory development teams. Fusion is hosted on a secure web server (VDL) and program access can be requested at https://www.vdl.afrl.af.mil/.

3.1.3.2 Software Development Approach

The Fusion software development team leverages SCRUM, an agile software development process (see Figure 6). The Fusion source code repository is hosted on VDL and a strict configuration management process is followed. Once a week, offsite developers submit their changes, and the core Fusion team integrates those changes and posts a new version of Fusion on VDL for the offsite developers and research team. Source code is managed through Git (a software configuration repository structure) using the Defense Research & Engineering Network (DREN). This process allows all offsite laboratories to keep up to date with the core Fusion team as well as keep their software well maintained.

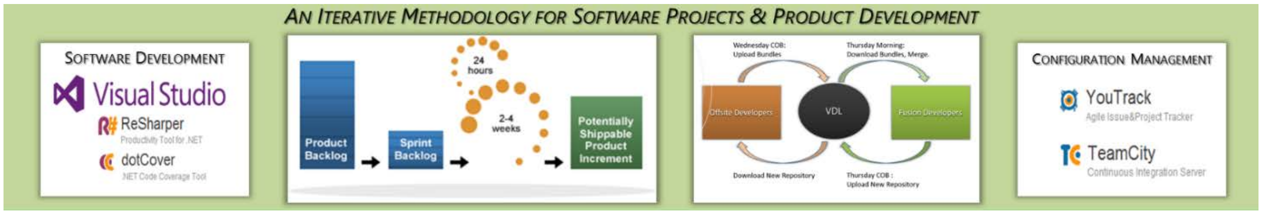


Figure 6: SCRUM Agile Software Development Cycle

3.1.3.3 Flexible Software Architecture

The Fusion Framework consists of a layered architecture supporting disparate research projects with a development kit to explore a variety of research goals. The framework consists of four fundamental layers: (a) the core framework layer, (b) the extensibility and application programming interface (API) layer, (c) the module / messaging layer, and (d) the application layer (see Figure 7). The core framework layer provides foundational software classes and an API. This layer enables functionality for module lifecycle, user profile, and display layout management. Additional features of this layer include system level notifications, multi-modal interactions with feedback, workspace management, asset management (vehicles, tracks, sensors, named areas of interest, etc.), geospatial information systems (GIS) data and earth mapping capability, as well as HMI elements. All software modules maintain a public framework API to support interface extensibility. This is accomplished in the extensibility and API framework layer. The module and messaging layer contains code written for single and specific purposes. This is the layer that contains HMI, utility classes, and messaging protocol support for communication to external software components. Finally, the application layer contains code related to executable applications such as a test-bed, utility application, or TOC. All code is written utilizing agile software development principles (SOLID: Single Responsibility, Open/Closed, Liskov Substitution, Interface Segregation, and Dependency Inversion) (Martin, 2012).

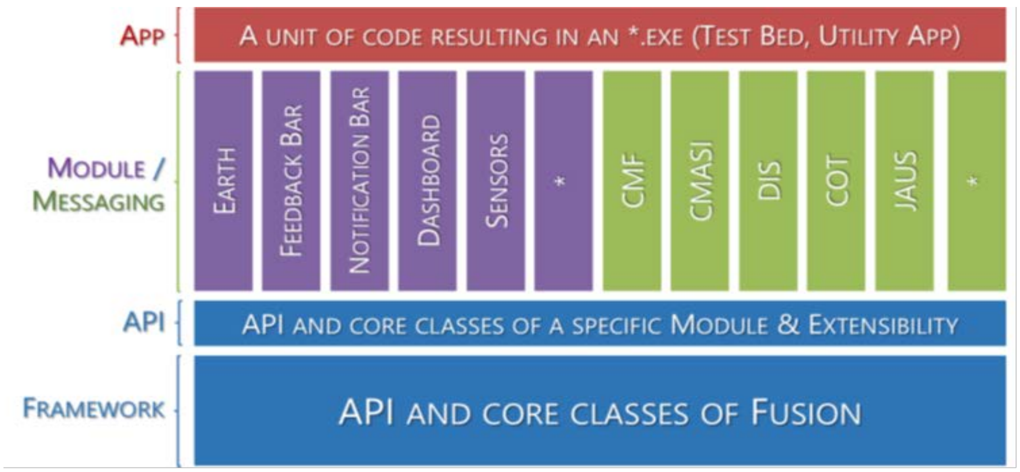


Figure 7: Fusion Layered Architecture

There are four primary research threads that Fusion is addressing to accomplish the goals of developing a framework for human interaction with flexible automation across multiple UxVs: (1) developing a software system that can generalize disparate and similar messaging protocols to be protocol-agnostic while allowing a many-to-many relationship between networked systems for the generation, distribution and consumption of network messages; (2) developing a software framework where every public element, regardless of its role as a model or user-interface element, is customizable, extendable and override-able by any other software developer in the system; (3) developing a software system that is fully instrumented to gather real-time user/machine interactions and system details for use in experimentation, software agents, and machine learning; and finally, (4) developing a software system that records the state of each of its components and makes it user-accessible to enable discrete and continuous retrospection of the system in real-time.

3.1.3.3.1 Cloud-based Simulation Architecture

The development team has established an API for external software components to communicate and interact with Fusion. To date, vehicle simulations, intelligent task allocation agents, vehicle planners, speech interpreters, chat systems, sensor visualization, operator assistance components, map layer data, and monitoring components have been incorporated into the Fusion network

API. These networked components employ various connection modalities (e.g., User Datagram Protocol (UDP), TCP/IP, ZeroMQ) and communicate using various messaging protocols (Light-Weight Message Control Protocol (LMCP), JavaScript Object Notation (JSON), Distributed Interactive Simulation (DIS), and custom protocols). In some form, all the components are linked together in their communications modalities by use of a centralized hub (see Section 4.1.3.5.1). Where appropriate, the connections and protocols are also realized into appropriate interface components in Fusion, and are intended to aid in creating a more immersive and interactive system for human-autonomy teaming.

The goal of this network API is to make the incorporation of external software as transparent and natural as possible while leveraging data efficiently. All of the instrumentation data is distributed to the centralized hub, and any component that wishes to consume the data can do so with a subscription. Likewise, communication messages from the other components are delivered to the same hub, and Fusion (or any other component) can subscribe and receive those messages. Each of the networked components may also communicate with another networked component using this same network structure. The publish/subscribe architecture present on the centralized hub makes for a natural assembly: all the associated data published by any software entity is available to any other service that needs to leverage it, thus enabling flexibility in the potential interactions between the services, including Fusion and its operator(s). It also establishes the framework that will be needed to extend the IMPACT system to support a multiple operator/multiple unmanned system interface thus enabling task/goal sharing and handoff among operators in the overall system.

3.1.3.3.2 Software Extensibility

Fusion is being used in several different projects, all of which share the goal of improving operator interactions with highly autonomous systems but have vastly different HMI designs and algorithms. Due to this, Fusion was built with the goal of extensibility throughout the architecture.

The Fusion infrastructure enables software developers to override aspects of the HMI by utilizing Fusion’s layered architecture to leverage the building blocks for HMI tools and services. The framework enables developers to add new HMI tools and services by overriding those building blocks and developing new modules. Thus, developers can override or extend aspects of Fusion without altering the original or previous extensions. Modules can be either universal or project-specific. Through this, the researcher can choose which modules are loaded, and therefore affect how the Fusion HMI appears and reacts to user inputs.

One example of the extensibility currently realized in Fusion is the vehicle symbol. In test beds that allow operators to control or supervise unmanned systems, vehicle symbols are important and appear in multiple areas in the HMI. Within Fusion, vehicle symbols appear on the map, in various notifications, on the vehicle status tool, in tasking tools, in many project specific tools, and other locations. Project specific vehicle symbol designs can easily be represented within the Fusion framework with a single line of code in the project-specific vehicle symbol specification, all vehicle symbols in the Fusion test bed can then be replaced. These features can then be realized at run-time vs. at source code implementation.

Extensibility saves a great amount of development time and empowers designers to test multiple solutions. A HMI can be designed and implemented in multiple ways and, depending on which modules the user loads, a specific design is realized. This facilitates experimentation on design candidates.

3.1.3.3.3 Interface Instrumentation

Data collection, agents, and machine learning all require the real-time capturing of data, which must be stored or packaged and sent across the network. HMI interaction is a prime example of one of these critical data sources. This capability was built into the Fusion framework to provide a non-invasive mechanism to the developers and provides a host of information, post-hoc and real time. Every user interaction, such as button clicks, typing, and mouse clicks are recorded and saved to a database.

All instrumented data is also packaged and sent through the network to any service connected to the centralized hub such as; agents, machine learning algorithms, cognitive modeling services, or other automated services that subscribe to the data source. Instrumentation of all operator interactions is critical for effective evaluation of human-autonomy teaming performance measures. This feature can be used to advance the capabilities of machine reasoning.

3.1.3.3.4 Human-Autonomy Dialog through Retrospection

All of the instrumentation data can be used for retrospection, allowing it to be re-played post process or played back during runtime. Retrospection has two main applications (and potentially more): experimenters can observe what was occurring to analyze why an operator performed an action or series of actions, and operators can “pause” and “rewind” the scenario to get another look at something that occurred in the past, further enhancing the human-autonomy dialog.

The concept of an operator being able to review the actions of an autonomous agent prior to the execution of those actions introduced the concept of a sandbox display. The sandbox is an area of the HMI where the operator can invoke actions that are not instantly carried out by the UxVs. This allows the user to evaluate autonomy-proposed actions and tweak various parameters prior to committing to them. Other displays within Fusion still depict current vehicle activities in real time, so the operator maintains effective situation awareness (SA), therefore giving the operator more insight into the autonomous component actions and reasoning. Another use of the sandbox is to play back the scenario using the instrumented data to see what occurred at some point in the past. This could possibly help operators make more informed, quicker decisions in the future. Further development within the Fusion framework is required to fully enable this feature and work is underway to explore those possibilities. The concept of a Sandbox display is discussed in more detail throughout this report.

3.1.3.4 Fusion Visual Framework

The Fusion visual framework is broken into six key concepts: (a) Login, (b) Layout, (c) Notification, (d) Feedback, (e) Canvas, and (f) Tiles (see Figure 8).

Figure 8: Fusion Visual Framework Components

While most of Fusion is customizable, there are a few core aspects that are common across all projects. Each project maintains specific scenarios that contain the instructions on which modules should be loaded as well as how the Fusion visual framework is laid out and operates. Fusion requires a user login and profile which contains information about a specific user such as last selected scenario and visual layout. There are also several key HMI components common across all scenarios, such as screen layouts, canvases, feedback/notification bars, and tiles. All of which are completely configurable to meet the needs of the scenario.

The layout system gathers information from the operating system on the number of physical displays connected as well as their resolution. To avoid confusion, Fusion internally renumbers the screens based on their top left corner position, where ordering is from left to right, top to bottom. This allows the layout to be consistent across varying machines with potentially different screen layouts and resolution configurations. The Fusion layout identifies which physical screens are to be used, what canvas to show, if the notification/feedback bars are to be shown, and if that screen is configured as a sandbox.

Each screen can have either an earth canvas, a blank canvas, or a custom canvas. The canvas can be thought of as an artist's canvas of which to place a variety of HMI elements. The HMI elements can be embedded in the canvas itself, such as the earth, or can be a space to place tiles. Custom canvases can be made to suit any projects' needs. Two additional core HMI elements include the notification bar and the feedback bar.

3.1.3.5 IMPACT Architecture

The IMPACT architecture is composed of a number of services, many of which are connected through the centralized hub/ZeroMQ Hub. These services include Fusion, Dialog, Aerospace Multi-Agent Simulation Environment (AMASE), SubrScene, CECEP, UxAS, plan monitoring, state server, database sources, speech support, and One Semi-Automated Forces (OneSAF; see Figure 9).

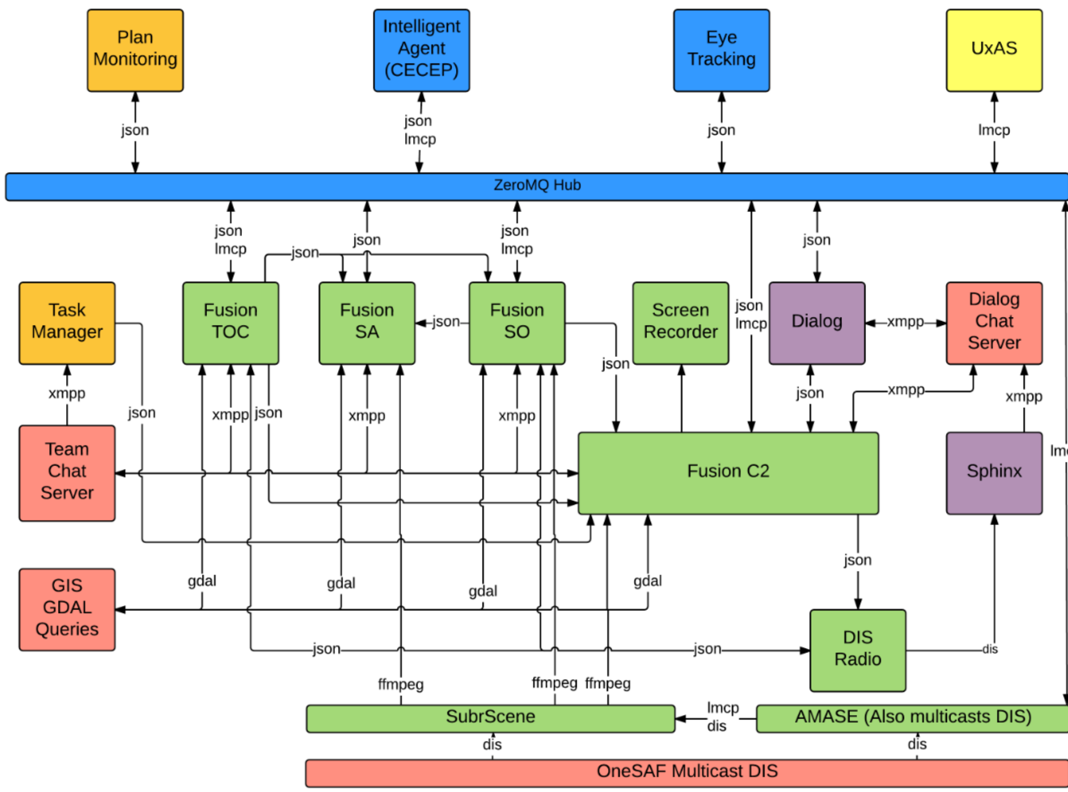


Figure 9: IMPACT Architecture

3.1.3.5.1 Hub

As the central point in the architecture, the hub has the responsibility of vectoring messages to any subscribed service. The hub directly supports ten connections in the full scale IMPACT configuration. It forwards, as appropriate, messages in either LMCP or JSON format. The Transmission Control Protocol (TCP) connections support LMCP messages only, but all other connections are protocol-agnostic.

The hub is a Java implementation with configurable socket connections. It employs ZeroMQ publish sockets and pull sockets to collect messages from and deliver messages to various connected software entities. It additionally supports a configurable connection as a client to a TCP server. The various connections are bridged in the hub such that any connection that sends a message can have that message forwarded to all other subscribed connections. In the full scale IMPACT architecture, the ten connections are: a simulation (AMASE), a component for tactical route planning and execution (UxAS), an intelligent agent (CECEP), an autonomics component for plan monitoring and feedback, the dialog support, the state server, and four Fusion connections (C2, sensor operation, test operator support and a status display for the test operator). The IMPACT hub employs a two-part messaging protocol, with the header defining the message type and the body containing the content. The header then dictates which messages are delivered. The convention within the IMPACT messaging structure is to define the protocol of the message followed by a class description. The hub enables connections of all the software components to AMASE, and thus enables a straightforward mechanism for adding other simulations or exchanging them for real-world data feeds.

3.1.3.5.2 OneSAF

OneSAF was used to create the complex scenarios representing the friendly and opposing forces in the environment. It sends UDP multicast packets of DIS entity states and the entities it specifies move according to the scenario set up in OneSAF. AMASE then collects the UDP multicast entities, translating them into the LMCP entity specification and then evaluating whether the OneSAF entities are detected by the operator-controlled vehicles. These perceptions are then sent on so they can be acted on as appropriate in Fusion. OneSAF serves the role of providing some external entities that can be acted on, and otherwise is not subject to much of the IMPACT architecture.

3.1.3.5.3 AMASE

AMASE is a Java-based unmanned vehicle simulation. It simulates, at limited fidelity, unmanned surface vehicles, unmanned air vehicles, and unmanned ground vehicles. While advanced vehicle routing and task execution is handled by UxAS, AMASE supports basic waypoint-based navigation on all three platform types. This includes an A* algorithm for finding road-constrained routes for ground vehicles and simulating that surface vehicles become immobile if they leave a defined “water region”. For every vehicle that AMASE simulates, a configuration and initial state is created. When AMASE is running the simulation, it updates the states as appropriate. Depending on the vehicle, AMASE supports multiple navigation modes. Air vehicles in AMASE support loiter, flight director, and waypoint navigation modes. Surface vehicles support flight director and waypoint modes. Ground vehicles support waypoint navigation mode only, as they are specified within AMASE to stay on the prescribed road network.

AMASE consumes DIS entities generated by OneSAF to populate its internal LMCP entity list and generate detection events as appropriate. AMASE also sends DIS entity states for its vehicles to SubrScene so the vehicle can be visible to others in the sensor feed.

3.1.3.5.4 Fusion Instances

Fusion supports Protocol Buffers, Cursor on Target messaging, STANAG 4586, LMCP, JSON, and DIS messages. Fusion connects to the hub through the ZeroMQ sockets described earlier. It sends JSON messages through the hub to CECEP and other Fusion instances and LMCP messages to UxAS and AMASE. It receives LMCP messages from AMASE and UxAS; JSON messages from Fusion instances, CECEP, Dialog, and plan monitoring; video feeds from SubrScene, and GIS information from data servers. Fusion displays this information, where appropriate, in the display mechanisms described earlier. The play communication aspects are a specialized subset of the messages, and are described in Section 4.3.

3.1.3.5.5 SubrScene

SubrScene renders and delivers simulated video feeds. It consumes LMCP vehicle states, translates them internally to DIS entity states, then renders the sensor feeds according to these states. It also collects the UDP multicast DIS entity packets being published by OneSAF to populate other entities in the view. It produces UDP multicast mpeg encoded video streams that Fusion displays. The same mechanism can be employed to deliver alternate sensor feeds from some other source, such as a live camera feed from an unmanned vehicle.

3.1.3.5.6 CECEP

The CECEP agent supports play calling by evaluating possible allocations of unmanned vehicles against play constraints. It also fields queries from the dialog. It is central to the IMPACT concept of play calling, as it provides a mechanism for abstraction of an operator’s management away from vehicle specifics and towards task requirements. More details are described in Section 4.3.

3.1.3.5.7 UxAS

UxAS handles planning requests formatted as LMCP messages. It connects to ZeroMQ ports on the hub, collecting tasks and requests. It translates these requests into responses, with complete waypoints and loiters as appropriate. For planning purposes, it also returns information regarding estimated times enroute which enables ranking and evaluation of competing plans for CECEP. During task execution, UxAS monitors vehicle states and updates waypoints and sensor steering commands as the task unfolds. More details are described in Section 4.2.

3.1.3.5.8 Dialog

The Dialog component connects to the hub through the ZeroMQ sockets. It sends data to CECEP to support queries and Fusion for translating speech commands into HMI responses. It maintains a direct connection to Fusion in addition to the hub connection. It also enters an Extensible Messaging and Presence Protocol (XMPP) chat room to provide live transcripts regarding the interpreted commands spoken to the system.

3.1.3.5.9 Plan Monitoring

The plan monitoring connects to the hub through ZeroMQ ports. It evaluates ongoing plays against their associated plan to determine the quality of the execution. It provides alerts to operators and status updates whenever some global constraint is violated. (more details in Section 4.4).

3.1.3.5.10 State Server

The state server connects to the hub through ZeroMQ ports. It collects all LMCP and JSON messages, capturing and recording the state of the overall system.

3.1.3.5.11 Speech Recognition

Speech is captured with the transcript collected by the speech support component. It then translates the speech into a set of utterances, which is parsed and managed by the dialog. It connects to an XMPP chat server to transcribe the data for the dialog. It does not connect directly to the Hub. Its connections are through an HTTP server to an XMPP gateway and a connection to a DIS radio through Fusion. Fusion generates the DIS radio messages, then a speech module records the speech to an audio file. The audio file is parsed with the PocketSphinx based speech engine, and the resulting speech interpretation hypothesis is transmitted to the XMPP server, which is parsed by the dialog.

3.1.4 Capabilities Developed

The core Fusion framework provides additional capabilities to execute project specific scenarios.

3.1.4.1 Geospatial Information Systems (GIS) Mapping Capabilities

A common tactical situation display was necessary to provide a rich GIS mapping experience for the user. The Fusion team developed SharpEarth which is a 3D mapping tool used to display geospatial data and layers onto a 3D representation of the earth. SharpEarth was created as a wrapper in C++ to extend the functionality of the C++ toolkits OsgEarth and OpenSceneGraph provides into the C# programming language. Extending such massive toolkits enables Fusion to have a feature rich map with a large library of GIS layers such as Web Mapping Service imagery, Elevation data, Weather Layers, and Tiled Image Layers (tiff, png, jpg), and to also support the display of any 3D object on the map such as shapes, text, icons and indicators. Touch and mouse manipulations of the map are fully supported and can easily be managed through a well-defined interface allowing complex interactions on the map such as shape manipulations. The earth can be displayed as either a tile or a canvas inside of Fusion and can be completely customized through an earth configuration file without the need to change the code base.

3.1.4.2 System Help

Fusion provides a robust interface for displaying richly formatted help files specific to each component. These are defined as HTML and completely configurable by the system developer. The help system is integrated at all levels of interaction with the overall system.

3.1.4.3 Media Manager

The Media Manager (Figure 10) allows the user to view and markup images, view videos, and listen to audio files. By default the media manager monitors the output directory for the current Fusion run and loads any existing media therein as well as any new media that becomes available in this directory. Additional media directories to load and monitor can also be specified.

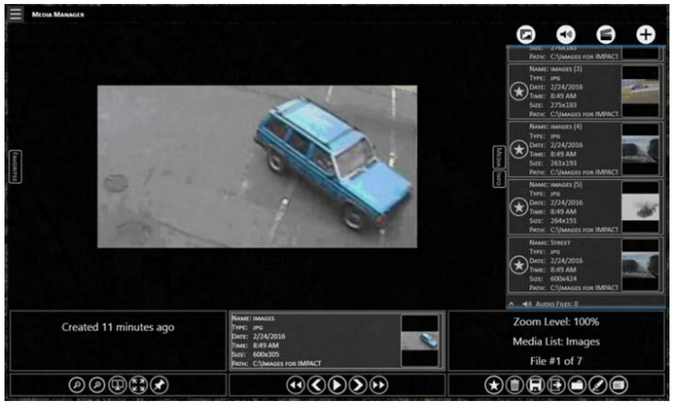


Figure 10: Media Manager

3.1.4.4 Vehicle Dashboard

The Vehicle Dashboard (Figure 11) is a visual overview of the all the components of a specific vehicle. All the components on this view are to provide at-a-glance information to allow the user to view all major events associated with each vehicle. The vehicle popup head up display (HUD) provides the bulk of the high level information the user would need to make reactive decisions to the scenario on a per vehicle basis.

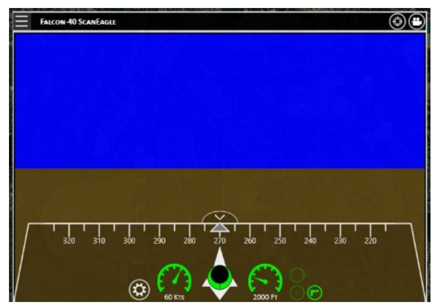


Figure 11: Vehicle Dashboard

3.1.5 Component Testing

The Fusion framework and software development team utilizes a test-driven development cycle. This ensures that all aspects of the software system are properly peer-review and tested throughout the lifecycle of the project. Test cards, use cases, and robust protocols were developed throughout the project enabling an effective iterative development of the overall system. Test engineers as well as human factors researchers could then effectively perform their respective testing of the system at the appropriate level. The overall goals for the Fusion project were to foster a rich testing environment across a multi-domain scenario and project specific trade space.

3.1.6 Lessons Learned and Next Steps

Large scale software development can pose a rather complex set of challenges to a diverse and geo-graphically distributed team. The concept of a VDL with a robust configuration management scheme quickly became necessary. Providing a common software framework for all to utilize across the VDL was paramount to the success of the IMPACT project, resulting in the Fusion Framework. Setting up a shared repository, utilizing an agile development process through SCRUM, and scheduling regular technical interchange meetings proved necessary to the sustainment of the IMPACT project.

Future enhancements to the framework include: (1) support for distributed operations (multiple C2 stations interacting), (2) enhanced retrospection within decentralized communication incorporating platform autonomy, (3) visualization of decentralized asset data when "re-synchronizing" with the centralized C2 system, and finally (4) maintenance and extension of the functionality of the Fusion framework to support manned-unmanned teaming and future autonomy-based projects.

3.2 UxAS

3.2.1 Motivation and Challenges

Current ground control stations for unmanned vehicles provide relatively low levels of autonomy, e.g. automatically commanding an assigned vehicle to fly to a sequence of waypoints generated by a human operator. The goal of UxAS is to provide increased levels of autonomy for control of heterogeneous teams of UxVs. Toward this end, UxAS provided flexible and adaptive automated path planning, sensor steering, and inter-vehicle coordination for unmanned air, ground, and surface vehicles in IMPACT. Specifically, UxAS addressed the following challenges raised in this project:

1. Reducing operator workload by implementing automated vehicle path planning and sensor steering tasks that underlie each play.
2. Increasing the reactivity of certain plays by implementing autonomous vehicle behaviors that enable inter-vehicle coordination and adaptation to changing conditions.
3. Increasing play flexibility by providing tunable parameters that are usable by both human operators and other forms of autonomy implemented in IMPACT.
4. Providing support for air, ground, and surface vehicles.
5. Implementing batch processing to support agent reasoning over alternative COAs.
6. Architecting the software to enable the rapid addition of new capabilities and support improved test, evaluation, verification, and validation.

3.2.2 Software and Hardware Acquisitions

The UxAS software package was produced and used throughout the IMPACT ARPI to implement automated vehicle path planning, sensor steering, and inter-vehicle coordination. The core UxAS framework and associated IMPACT services have been approved for public release and are available to interested developers for follow-on work at https://github.com/afrl-rq/OpenUxAS.

Improvements were also made to the AMASE to support new plays, vehicles, and message sets developed in IMPACT. An open version of AMASE is approved for public release and is available at https://github.com/afrl-rq/OpenAMASE.

3.2.3 Development and Implementation

Over the past 10 years, researchers at AFRL have been developing and flight testing algorithms to automate cooperative Unmanned Air Vehicle (UAV) missions. As part of the IMPACT effort, the software that implements cooperative UAV autonomous decision-making and route planning underwent a modernization effort in order to make it easier to extend and maintain. Currently, UxAS software forms the foundation for experimental research programs ranging from human-machine interaction to decentralized cooperative control.

UxAS consists of a collection of modular services that interact via a common message passing architecture. Similar in design to the Robot Operating System, each service subscribes to messages in the system and responds to queries. UxAS uses the open-source library ZeroMQ to connect all services to each other. The content of each message conforms to the LMCP format. Software classes providing LMCP message creation, access, and serialization/deserialization are automatically generated from simple Extensible Markup Language (XML) description documents. These same XML descriptions detail the exact data fields, units, and default values for each message. Since all UxAS services communicate with LMCP formatted messages, a developer can quickly determine the input/output data for each service. In a very real sense, the message traffic in the system exposes the interaction of the services that are required to achieve autonomous behavior.

Consider a simple example: the automated construction of the flight pattern to conduct surveillance of geometric lines (e.g. perimeters, roads, coasts). A “line search task” message describes the line to be imaged and the desired camera angle. Using this input description, a line search service calculates the appropriate waypoints to achieve the proper view angle. When the UAV arrives at the first waypoint corresponding to the line search task, the line search service continuously updates the desired camera pointing location to smoothly step the camera along the intended route during task execution.

In addition to surveillance pattern automation, UxAS contains services that automate route planning, coordinate behavior among multiple vehicles, connect with external software and hardware devices, validate mission requests, log and diagram message traffic, and optimize task ordering. In all, UxAS has approximately 50 services. In the IMPACT system, UxAS works in collaboration with the intelligent agents to determine allocation of vehicles by conducting route planning and tailoring on-task behavior.

3.2.4 Capabilities Developed

UxAS was originally designed to test cooperative control algorithms. While some IMPACT plays utilize UxAS for automating complex cooperative behavior (such as the blockade and cordon plays), much of the contribution to the overall IMPACT system revolved around leveraging foundational capabilities such as route planning and surveillance pattern calculation (Kingston, Rasmussen, & Humphrey, 2016). Additionally, accommodating flexible play calling required careful consideration of the ways in which the automated behaviors could be tailored for the situation at hand. Finally, the software design itself was updated to be modular and more easily verified and validated.

The remaining subsections correspond to the challenges described in Section 0 and the primary capabilities developed to address those challenges.

3.2.4.1 Automated Path Planning and Sensor Steering

Autonomous vehicles operating in real-world contexts must reason soundly about the availability and timeliness of routes that reach their goal locations. Of particular importance is the ability to plan routes in spaces that are constrained by regulation (e.g. air tasking orders), environment (e.g. terrain), and physical motion limitations (e.g. minimum turn radius). During IMPACT, UxAS route planners for both fixed-wing aircraft and ground vehicles were created. A robust interface specification allows for other route planners to be easily added to account for additional vehicle types (such as underwater vehicles).

The developed aircraft path planner ensures that minimum turn radius constraints are met while simultaneously avoiding “no-fly” geometric regions. The technique is based on a triangularization of the space and a subsequent search for a series of adjacent triangles that have edge lengths greater than the minimum turn radius of the vehicle. Although this precludes consideration of narrow (yet feasible) corridors in the environment, the resulting path is guaranteed to meet the turn radius constraints of the vehicle and is extremely fast even for complex environments (e.g. ~3ms for typical UAV airspace constraints). It should be noted that this technique is limited to fixed-altitude operations; however, in our experience this is rarely an issue because integrating with military airspace generally requires that UAVs fly in certain altitude slots during operation.

The focus of our path planning research has been to support rapid, robust planning techniques for aircraft; however, to allow heterogeneous vehicle operation, incorporation of path planners for other types of vehicles is needed. UxAS is architected to abstract the path planning from higher-level decision making; this natural separation allows the development of a common interface to adapt additional planners to fit in the whole system. To demonstrate this, we developed a ground vehicle route planner that plans routes on Open Street Maps road networks. This ensures that ground vehicles adhere to feasible roads, yet abstracts that detail so that decision logic need only reason about timing and availability. A modified version of the aircraft path planner was used to plan routes for surface vehicles, although it is anticipated that a path planner that works directly with surface vehicle limitations will ultimately be needed for real-world applications.

3.2.4.2 Autonomous Vehicle Behaviors and Inter-Vehicle Cooperation

In simple terms, a play call consists of reasoning over vehicle availability and response time to fulfill the play goals. Once the vehicles that will service the plays have been determined, the actual unfolding of the play is handled by UxAS. Each play is ultimately decomposed into a series of steps that the autonomous vehicles follow to reach the goal. For example, once an "escort" play is in progress, the assigned autonomous vehicles must react to the supported convoy's motion. During this play, UxAS estimates the motion of the convoy and determines the proper leading, overhead, and trailing surveillance locations. As the convoy updates its position, the team adjusts their surveillance goals to constantly keep watch.

Similarly, each play involves its own unique "on-play" behaviors ranging from optimal search pattern calculation to cooperative port blocking maneuvers. Although many plays have a static sort of behavior (e.g. traverse a pattern for optimal coverage), numerous plays rely on adaptation to changing circumstances (e.g. movement of enemy and friendly vehicles).

In addition to determining the on-play autonomous behavior for both single and multi-vehicle plays, UxAS provides a mechanism for incorporation of external on-play behavior software. By leveraging the modular architecture, new plays that require different on-play behavior can be easily incorporated.

3.2.4.3 Tunable Play Parameters

One of the key goals of IMPACT was to provide operators a means to tailor autonomous behavior to handle situations that were never explicitly designed. By providing the ability to update plays rapidly, it is anticipated that their applicability will be wider. A major consideration for design of autonomy, therefore, is the choice of ways in which the operator is allowed to "tune" a particular play. Working with the operator interface and agent teams, parameters for each play that should be tunable during operation were identified and implemented. For example, in simple point surveillance tasks, the approach angle can be directly specified to ensure a view from a particular direction, or it can remain free for the autonomy to choose. UxAS provides many such parameters for flexibility including stand-off distance, loiter shape, time-on-target duration, and search pattern directions. Coupled with the ability for the operator to rapidly update the airspace constraints, a wide range of very precise goals can be met without the operator resorting to placing waypoints manually.

3.2.4.4 Support for Ground and Surface Vehicles

Although UxAS is primarily focused on aircraft, particular design attention was paid to separating high-level reasoning and decision making from the details of planning for particular vehicles. This clean separation and abstraction allows seamless inclusion of other vehicle types. Currently, UxAS uses simple assumptions to plan for both ground and surface vehicles and provides a baseline capability for a complete air/ground/water mission. It is anticipated that replacing the baseline calculations with software that accounts for the complexities of these vehicles will use the same interface to connect with the system and thus leverage the entirety of IMPACT with minimal change.

3.2.4.5 Batch Processing

As the decision space grows in larger multi-vehicle missions, a means to rapidly calculate and aggregate the necessary timing data is required in order to optimize vehicle allocations. UxAS therefore allows queries to be made in a "batch" mode in which entire timing tables are calculated and formatted for ease of use in higher-level decision logic. This is done in a deliberately scalable manner so that such requests can be handled in parallel (i.e. on a cluster of computers). The intelligent agents heavily rely on these calculations to make recommendations to the operator.

3.2.4.6 Improved Architecture

A common theme for each capability described above is the notion of a modular, extensible architecture. This entails a principled separation of underlying technologies, essentially finding the proper seams that allow the ability to optimize where possible while simultaneously abstracting functionality so that replacements and updates can be made without side-effects to the overall system. UxAS underwent a thorough re-architecting during the IMPACT effort to meet the challenges of supporting many types of plays and vehicles. A major benefit of this is the ability to target testing and simulation toward key parts of the system while retaining confidence in overall system operation.

3.2.5 Component Testing

UxAS currently includes 17 automated unit tests that help ensure core capabilities remain functional when new capabilities are added or changes to existing capabilities are made. Additional automated unit and system-level tests are currently under development to allow quick verification as the software changes.

In addition to simulation and user evaluations, UxAS is frequently flight tested as a critical part of the Intelligent Control and Evaluation of Teams (ICET) project. On average, ICET conducts flight tests 3 times yearly. In these events, UxAS runs onboard small UAVs and provides the route planning and high-level reasoning capabilities needed for conducting cooperative surveillance missions. Much of the functionality developed for IMPACT is utilized directly in these flight tests. During the week of 20 Feb 2017, the entire IMPACT system was used to simultaneously control live aircraft and simulated ground vehicles in a cooperative mission.

3.2.6 Lessons Learned and Next Steps

As autonomous systems grow in capability, the software needed to realize that capability also grows. Joining several nascent technologies into a complete functioning system was the largest obstacle facing the IMPACT team, and a number of lessons were learned throughout the integration process. For instance, the decision to use a common message passing framework to connect components paid great dividends and should be strongly encouraged in similar programs. However, this alone is not sufficient: great care must be taken to ensure that all parts of the system remain synchronized to the current version of the message set. Additionally, functional dependence between pieces of the system can cause wide-ranging side-effects to the system as a whole. For example, a simple change in how the simulation behaves at terminal waypoints caused a ripple effect in which other parts of the system could no longer determine when plays had finished. A standardized process for careful regression testing and identification of wide-reaching changes should be rigorously employed to minimize such disruptions.

The future for UxAS includes incorporating these lessons learned in order to improve the ability to continuously verify and validate its functionality as its scope expands. To this end, the AFRL Summer of Innovation will apply cutting-edge V&V techniques to UxAS to formally capture its architecture and analyze its properties. The resulting V&V-amenable revision of UxAS will then be used on a government-provided basis in the AFRL Loyal Wingman program, which aims to augment a manned fighter with unmanned teammates.

3.3 Intelligent Agents

3.3.1 Motivation and Challenges

Early in the effort, stakeholders participated in a technical interchange meeting to define the agent team’s objectives for addressing technical challenges in the IMPACT project. The objectives included advances in the following areas of USAF operational capabilities C2 of a set of heterogeneous UxVs using the Fusion UxV control station:

1. Advance current methods for modeling human-system interaction and integrating executable cognitive models into human-machine teaming systems.
2. Reduce overload of human operators by providing agent based decision making tools that solve common problems in the decision making domain involved in the C2 space of UxVs.
3. Allow a single operator to coordinate with an intelligent agent to control multiple heterogeneous UxVs simultaneously.
4. Provide transparency of agent decision making to increase situational awareness of the operator.
5. Provide answers to vehicle and play related queries that are input into the system by the human operator using voice or text input.
6. Develop background behaviors that are used to maintain behaviors that:

a. Provide a default play for vehicles that are not assigned to other tasks.

b. Trigger a Highly Mobile execution mode that assumes an elevated security risk is present in the base defense scenario.

c. Efficiently utilize available vehicles to provide a test scenario's base defense coverage without a large planning burden on the human operator.

3.3.2 Software and Hardware Acquisitions

To address IMPACT technical challenges, artifacts and agents were developed using the CECEP software architecture. CECEP is a complex event processing framework with extended procedural and domain knowledge aspects. Short term functionally salient memory have been integrated into the CECEP Framework that are shared by various components as events in an event cloud. In the IMPACT effort, agents were specified used a modeling language called Research Modeling Language (RML) 2.19. RML is a modeling language developed in the tool Graphical Modeling Environment (GME). Code generation tools were used to produce executable code artifacts in from the RML agent models.

Agents that use procedural knowledge are developed using a FSMs representation called behavior models (BMs). BMs include states and transitions between states that are guarded by patterns. A pattern language called Esper Pattern language is used in CECEP to match complex patterns in the event cloud for BM state transitions. BMs can also produce behaviors such as play calling feedback for the operator or assign vehicles to plays. BMs developed in the CECEP framework were the technical approach for addressing all human interaction monitoring, as well as, UxV monitoring, management, and allocation assignment aspects of the technical challenges.

Agents that use domain knowledge are developed using feature models called cognitive domain ontologies (CDOs). A CDO is a rooted tree with features that are connected via relations. Four types of relations are supported in the framework sub-parts, choice-points, multi-choice-points, and instance sets. CDOs can be processed using the AI process of constraint satisfaction to produce possible configurations or possible worlds. CDOs can produce all constraint compliant solutions, or a single best solution. In IMPACT, CDOs were used to address technical challenges regarding agent decision support, agent decision transparency, and background behavior allocations involving UxVs.

3.3.3 Development and Implementation

The CECEP architecture was integrated with external services to support IMPACT play calling. Services such as UxAS, Fusion interface, plan monitoring capability, and UxV simulator (AMASE) were integrated with CECEP. A shared specification for messaging event structure was developed in order to allow for data sharing between various services. External event sources were translated and placed into the working memory for agent and adapter consumption. Truth data containing relevant data, such as vehicle types and configurations, were made available to cognitive agents developed in IMPACT. This allowed for more agent reasoning and suggestion of optimal vehicles for play call allocation. Figure 12 illustrates how the CECEP agents interact with external services in the play calling process.

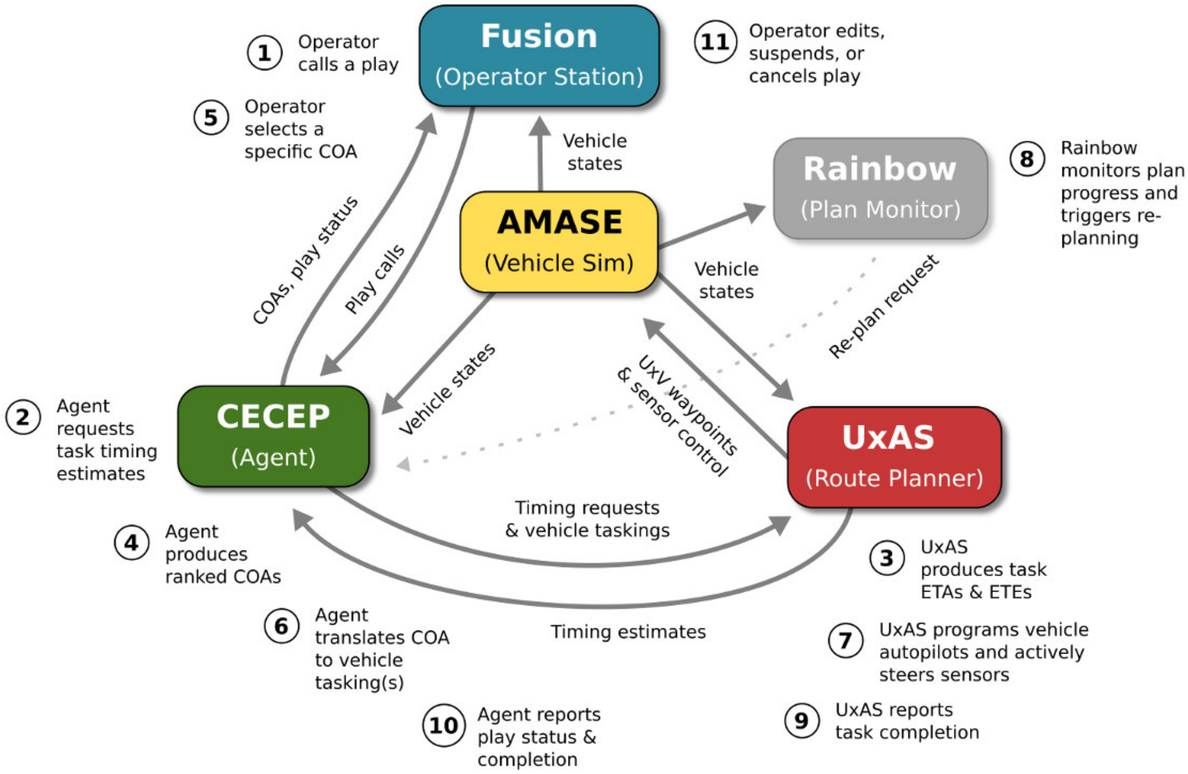


Figure 12: Play Calling Process in IMPACT

3.3.4 Capabilities Developed

Products of the ARPI IMPACT effort include an enhanced CECEP framework, CECEP agents, and CECEP adapters. CECEP model execution capabilities, originally developed for previous research and development efforts, were successfully extended in this effort. Agent modeling conducted by the agent team unearthed additional opportunities to extend the CECEP framework's capabilities and resulted in new functionality. Each capability in the section below corresponds to the numbered technical challenge objective listed in section 4.3.1.

3.3.4.1 Integrate Cognitive Models – Human-Machine Teaming

The agent team has developed three methods for integrating with a CECEP agent. The first method is to implement communication protocols in an external system that is compliant with existing CECEP communications. The second method is to create a CECEP adapter that connects to an external system and outputs event information into the format used within CECEP. The third method of communicating with a CECEP agent was developed specifically for this effort. The agent team developed a ZeroMQ hub that provides the ability to connect multiple event sources without regard to language or platform. The ZeroMQ hub component was developed to manage messages, represented as events, between components. An adapter was used to interface CMASI data, process it, and place it in CECEP's Esper event cloud where it can be used by CECEP agents and adapters. CECEP agents were developed to model human cognition and integrate executable cognitive models into human-machine teaming systems.

3.3.4.2 Decision Making Support for Reducing Operator Workload

The agent team developed a play calling CDO that, when processed through constraint satisfaction, produces all vehicle COAs for achieving a play call. Constraints are applied to ensure sensor, weapon, and other capabilities will meet the demands of the called play. A sorting algorithm is applied to rank all COAs and provide the human with a suggested best COA for a play call. Plays are sorted by minimized time, minimized fuel usage, minimized detectability, maximized presence, maximized crowd control capability, and/or maximized tracking capability. Environmental conditions, sensor NIIRS, and vehicle availability were also factored into the sorting algorithm.

Constraint programming using Java constraint solvers was used to process the IMPACT play CDO. Constraint programming is an approach to programming that combines reasoning with computing. Problems expressed as constraint satisfaction problems are defined using a set of domain variables and relationships between these variables in the given problem domain. Additional decision support capabilities such as communications range support, play chaining, and play delaying are supported by the agent. In IMPACT, every UxV asset has an effective operating range. This range constrains the area where plays can be called for an asset. Vehicles are configurable to include a communications relay payload that can be used to support a play call outside of an asset's effective operating range. The intelligent agent provides communications relay decision support for plays involving vehicles that will go out of communications range.

3.3.4.3 Operator Control of Multiple UxVs

Agents were developed that assist an operator with play calling in a semi-autonomous fashion. The IMPACT experimental design team, from 711HPW/RHCI, defined requirements for 26 play call types. Each play call has a unique set of play details, and the IMPACT agent team developed 26 play BMs to support all play call types. These BMs monitor operator interface interactions and environment information relevant knowledge for COA generation.

Multiple plays can be managed concurrently and this is handled in the agent's resource manager capability. In the resource manager, play state is managed and conveyed to the operator for improved system transparency. A play can have a state of active, ready, or not ready. Active plays are plays that have been accepted and are currently executing. Ready plays are those plays that have a solution and are waiting for operator interaction for acceptance. Not ready plays do not have constraint compliant solutions. Not ready plays commonly occur when a payload requirement is specified by the operator, but there isn't a vehicle available that has the desired payload.

3.3.4.4 Agent COA Transparency

One of the major challenges of autonomy and automated decision making is in providing transparency on automated decisions and actions. One method to validate agent decisions, which was adopted in IMPACT efforts, was to convey the information (constraints and domain conditions) used to make the decision. An explanation capability was developed by the agent development team. Each constraint in the IMPACT play calling CDO has a corresponding explanation. If a COA is constraint compliant, an explanation can be generated from that constraint that describes why the COA was not constrained from the solution space. The set of all active constraints with their corresponding explanation is provided to the operator to explain why a particular COA was valid.

3.3.4.5 Support Voice and Text Queries

BMs were developed to match patterns for operator voice and text query events and provide auditory and textual responses to the operator for play calling. A capability was developed to answer operator questions about vehicle to play, or capability to play assignments for predefined play locations such as: "How soon can I get an IR sensor on the Ammo Dump?." The operator can generate a visual of the play by saying "show me" or execute the play by saying "confirm". Other queries such as "what is a ***vehicle*** doing?" or "what is a ***vehicle's*** fuel state?" were supported as well. A total of 9 different query types were support in the agent.

3.3.4.6 Developed Dynamic Background Behavior

Background behavior capabilities were supported for all UxVs in IMPACT. Two background behavior modes were developed in IMPACT (i.e., normal full coverage patrol (NFCP) and highly mobile (HM)). In both NFCP and HM, all UxVs patrol their assigned areas of responsibility in or around the base. The default patrol state can be set to either NFCP or HM with the operator being able to quickly toggle between the two. When a critical event occurs (e.g., a threat to the base perimeter), the operator is able to switch to an HM patrol with a single button interaction.

The agent was developed to manage background behavior reassignments for play calls and cancelations, play completes, play pauses, manual control, and background behavior mode changes (HM or NFCP). The developed background behaviors minimize arrival time of allocated assets to NAIs or routes. Clusters are predefined for the eight ground NAIs by pairing four sets of two NAIs. Both background behavior modes support acclimation of UxV to task assignments when a play is called.

3.3.5 Component Testing

Testing was accomplished using TestRail, TeamCity, and two distinct methods of testing. Since some elements of play calling would be difficult to test automatically, we implemented both a manual test system and an automated test system. We performed both manual and automated testing using regression tests for modules and adapters as well as continuous integration testing (Jenkins) for the full system. Regression tests were coded for each agent and adapter so that when any changes were made, we would know immediately if the changes broke any of the other components based on the developed tests. Over 400 regressions tests were produced for agent testing.

3.3.6 Lessons Learned and Next Steps

CECEP agent development involves a significant amount of tribal knowledge that is learned primarily from interactions with experienced CECEP agent developers. Initially in the IMPACT project, there was limited documentation describing CECEP features and modeling. By the end of the project CECEP modeling documentation was improved to the point where less training was required for new developers.

Team coordination challenges existed in IMPACT due to the geographically diverse teams involved and the varied sequentially dependent responsibilities between those teams. We reduced the problems using weekly integration meetings, but still had occasional problems due to inconsistent availability of meeting minutes. Communication is essential and the IMPACT team improved in this area by the end of the project.

Several development processes were lacking early on in the effort, but were later improved. These included management of manual testing, automated testing, and configuration management. The agent team has since moved towards using elements of agile software development. We found that shared development of models was difficult given the limitations of our modeling toolset. Improvements were made to allow for concurrent development of agents.

CECEP evolved for this effort to include previously unavailable features. The most important of which is the ability to dynamically generate behavior models during runtime. Previous to this effort, any required behavior models had to be pre-allocated prior to runtime. This new capability allowed for multiple play calls of the same type, using generative behavior models that are dynamically allocated at execution time.

The agent development work that was completed in IMPACT has led to additional opportunities. The technical approach and lessons learned from the IMPACT project will be carried forward into future efforts. The direction of research for the agent team is likely to involve mission planning, operational design, decision making assistance, and the integration of hardware accelerated constraint solvers to speed up the constraint solving process.

3.4 Autonomics for Plan Monitoring

3.4.1 Motivation and Technical Challenges

With a single operator overseeing teams of heterogeneous unmanned vehicle platforms performing multiple concurrent missions comes increased complexity that is difficult to monitor. A motivating factor for the live monitoring of mission plan progress and anomalies is increased operator SA and reduction of information overload. Measuring holistic plan health along with vehicle telemetry, mission tasks and plans can provide an at-a-glance summarization of current conditions. To that end, we have chosen an autonomics based approach, a self-managing process, whereby a C2 scenario is formally modeled and thresholds for acceptable mission plan and global values are established and evaluated.

3.4.2 Software and Hardware Acquisitions

The hardware and software utilized was covered in Section 4.1.

3.4.3 Development and Implementation

With autonomics selected as our technical approach, our next step was to establish the communication and translation of IMPACT data into model components. Due to the complexity of IMPACT plans, and in order avoid the time-consuming task of manually creating models, we developed a mechanism to dynamically generate model components from IMPACT data. Configuring these models with templates defined a priori, we established plan health and global constraint evaluation. Finally, a HMI was developed with information useful to a Test Operator in which a concept of Working Agreements was explored.

3.4.3.1 Autonomics

Since the purpose of Plan Monitor is to provide feedback on the live monitoring of plans and anomalies, we chose to use an autonomics approach. Inspired by the human immune system, autonomics can monitor a system's attributes to provide an automated response when an undesired system state is detected. Plan Monitor leverages the Rainbow Autonomics Framework, developed at CMU, which enables autonomous management of networks through:

- the ability to dynamically monitor and analyze a network's properties,
- the ability to detect breaches on a network's architectural design assumptions, and
- the ability to effect changes on a network in response to breaches in design assumptions.

Rainbow grants software architects the ability to establish a model of a network through the use of ACME, a formal architectural design language. The model reflects network properties and rules enforce a network's design assumptions. Rainbow's probes and gauges dynamically translate the state of the network into the model while strategies, tactics and effectors provided automated adaptation to effect changes on the network. A key observation in IMPACT is that the structure of a plan allows plans to be represented as networks thereby granting the ability to manage plans with Rainbow.

Since Rainbow is written in the Java programming language, Plan Monitor was developed in Groovy, a superset language of Java. Plan Monitor is an extension of the Rainbow framework and communicates with IMPACT through the network hub using ZeroMQ. Its software elements are as follows:

- Model
  - Establishes components representing vehicles, tasks, areas of interests and zones.
  - Establishes components representing plans by connecting the components above to reflect plan structure.
  - Establishes templates, rules and thresholds ensuring the integrity of plans.
- Probes
  - Subscribe to network hub and ingest relevant messages. Report data to appropriate gauges.
- Gauges
  - Update components and properties in the model. May provide additional data processing.
  - Employ method to dynamically generate model components from network hub messages.
- Strategies and Tactics
  - Call effectors to act upon detection of poor plan health.
  - Call effectors to act upon global constraint breaches.
- Effectors
  - Publish plan health for display to the operator.
  - Publish constraint violation notifications for display to the operator.

3.4.3.2 Component Generation

Since an IMPACT scenario can be complex with multiple types of plays, manually developing models for each case is non-trivial. To that end, Plan Monitor employs a method of generating model components and connections between components dynamically. Using the Java reflection API, LMCP object metadata read from the hub is used to generate corresponding structures in the model.

An LMCP object to model conversion follows this pattern:

- LMCP object is read from the hub.
- Reflection tools collect field names, types and values (including those of parent classes in the object's class hierarchy)
- Presence of an abstract model component (template) for that object's class is verified and generated if it does not exist. (This operation happens once per object type.)
- Presence of a concrete model component (instantiation) for that object's instance is verified and generated from a template if it does not exist. (This happens once per unique object instance.)
- Component is updated using object's field values if the values are different.

This pattern allows for the generation of a model for any plan developed in IMPACT.

3.4.3.3 Plan Health

The primary function of Plan Monitor is providing plan health information to the operator. Through the constant monitoring and evaluation of plans we communicate their real-time status. Status falls within three categories: Nominal (Green), Lower Caution (Yellow) and Upper Warning (Red) with extent of deviation correlating to severity of status. Plans generally have two phases that Plan Monitor must consider:

- En-route – Vehicle has been assigned to a mission plan and is on its way. Parameters include:
  - Fuel – Thresholds are set by templates in the model for each vehicle type.
  - Speed – Thresholds are determined by plan metadata – each plan includes a set of way points for vehicles to follow with each way point establishing expected vehicle speed.
  - ETE – Thresholds are cached upon instantiation of plan by using the distance between way points and expected vehicle speed. Real-time vehicle telemetry is compared to its expected position along the route establishing ETE quality.
- On-Task – Vehicle has reached its destination and is performing its task. Parameters include:
  - Fuel – Thresholds are set by templates in the model for each vehicle type.
  - Speed and Task Quality – Thresholds are determined by type of task associated with a plan.

On-Task health is determined by the tasks associated to a plan. Although there are over twenty available plays for an operator to use, they generally follow six patterns. Plan Monitor categorizes these plays by their purpose and characteristics in order to facilitate the calculation of plan health. Plan categories are as follows:

- Search Plan –vehicles focusing their cameras on points or lines in the world.
- Watch Plan –vehicles focusing their cameras on a vehicle.
- Escort Plan –vehicles maintaining a distance from a vehicle.
- Cordon Plan –vehicles maintaining a distance from a point in the world (to section off an area).
- Blockade Plan –vehicles maintaining a distance from a point in the world (to actively obstruct passage of vehicles).
- Comm-Relay Plan – Special case as this plan is not called but generated to support a called play in need of communications relay.

While the details involved in these patterns may vary (friendly vs non friendly vehicle targets), it is sufficient to measure On-Task health. For multi-vehicle plans, health for each vehicle is compared and the lowest quality parameters are combined into a single plan health update.

3.4.3.4 Global Constraints

A secondary function of Plan Monitor is effecting the IMPACT scenario through notifications. Currently, there are three types of notifications:

- Fuel Notifications – A rule is established in vehicle model templates with each vehicle type defining its fuel threshold. A low fuel notification is published upon threshold breach.
- Restricted Operating Zone (ROZ) Notifications – A rule is established in a strategy triggered upon the generation of a ROZ Violation component in the model. A ROZ violation notification is published upon vehicle or way point presence in ROZ.
- Flight line Notifications – A rule is established in a flight line model template with location of flight line and vehicle response time thresholds. A flight line violation notification is published when there are no vehicles within time thresholds.

These constraints are evaluated at all times, regardless of whether there are any on-going plays.

3.4.3.5 Customized User Interface and Working Agreements

Due to the generic qualities of Rainbow, its built-in HMI provides tools useful to developers of Rainbow applications. However, the end-user is likely not concerned with the information presented by these tools as they communicate actions relating to Rainbow's internal processes. This provides an opportunity to explore a customized user interface with utility relevant to the network it is managing.

Plan Monitor uses the GroovyFX API to establish and render its HMI. It features three sections:

- A list with active plan data reflecting the name and type of plans currently monitored.
- An event log providing detailed plan related activity.
- A working agreements section enabling operator configuration of strategies.

Figure 13: Rainbow Gauge HMI

Behind the scenes, a HMI is controlled through a Rainbow gauge (Figure 13). This gauge initializes the HMI and employs a visitor software design pattern to establish itself as a HMI controller. HMI components are tied to settings in the model and changes are communicated through the gauge. This link between HMI and model allows strategy selection to be controlled by means of rule conditions. Consider the following rule applied to vehicle templates:

rule fuelRule = invariant !GUI_ALLOW_FUEL_UPDATES or EnergyAvailable > FUEL_CRITICAL;

Here we ensure that vehicle fuel is above a threshold and observe how the GUI_ALLOW_FUEL_UPDATES property affects the rule. This property is tied to a checkbox component in the Working Agreements HMI section. If the checkbox is unchecked, the property value is false and disables the rule despite EnergyAvailable falling below threshold. Thus, disabling strategies tied to vehicle fuel status is by means of the HMI. This mechanism supports a goal in our work with autonomics which is to promote transparency and collaboration between human machine teams. Working agreements establish a policy dictating what the autonomy is allowed to do. A motivating factor behind this effort is the realization that a human operator may have critical information about the world that the autonomy does not.

3.4.4 Capability Developed

Through automated at-a-glance plan health evaluation and constraint notifications, Plan Monitor helps increase situational awareness and works to mitigate information overload. Since plan monitoring is performed autonomously, it has the potential to scale with increases in complexity. This capability is important now but will become more important in future scenarios as single operators supervise increasing numbers of vehicle platforms and concurrent mission plans.

3.4.5 Lessons Learned and Next Steps

Since Plan Monitor is collecting streams of data from the network hub, it was a challenge to identify which data was necessary to accurately measure plan health. The current method takes into account the structure of a plan to establish plan quality but does not incorporate all related messages. For example, a plan's constraints can be selected upon play calling which are then published to the network hub as additional messages. The generated plan will consider those constraints. A next step for Plan Monitor would be to listen to these constraint messages and use them when calculating plan health. This will likely increase the quality and accuracy of plan health.

Another area for future research involves increased influence of autonomic adaptation. Currently, Plan Monitor is limited to affect the IMPACT scenario through plan reports and constraint notifications but it can potentially do much more. The ability for Plan Monitor to directly issue re-planning or directly call plays is possible through the use of re-plan and play calling strategies. These can be called upon the detection of poor plan quality, constraint or policy violations. Finally, by combining a third-party planner into play calling strategies, Plan Monitor could potentially call custom plays.

3.5 Task Management

3.5.1 Motivation and Technical Challenges

The IMPACT scenario is reliant on the integrated interaction between autonomous systems and a human supervisor. This interaction generates a large number of tasks to be completed by the operator of the IMPACT C2 station. With the onset of a high volume of tasks, tasking can easily become overwhelming and disorganized, leading the operator to become less effective and focused in completing assigned tasks. To address these issues, a management system capable of determining user tasks, dividing tasks into a hierarchy, presenting tasks to the user, and providing a mechanism to execute an action for each task was developed.

The tasks associated with this effort are tasks associated for execution by the human and not the autonomous vehicles for which the tasking is applied. Autonomous systems or agents can support the human within the management of the tasks in terms determining and sorting the tasks. In cases of high workload an autonomous assistant can off load tasks, based on some workload agreement as to when and how this would occur, and perform the functions of that task.

3.5.2 Software and Hardware Acquisitions

Software acquisitions required for the development of the Task Manager include Visual Studio Professional 2015, a development environment, and ReSharper a tool added to Visual Studio to help analyze code quality, eliminate errors, support code base changes, editing and compliance tools. Several computer systems were used to mirror the full IMPACT system in order to provide live demonstrations and integration test capabilities. For the design of the Task Manager these machines served a dual role as they also aided in both testing and development.

3.5.3 Development and Technical Approach

3.5.3.1 Task Model

Before discussing the implementation of the Task Manager, the basic structure for the Task Manager called the Task Model will be examined. The Task Model helps to structure tasks into a workable hierarchy that creates a clear delineation of the order in which a task is to be completed and the action points at which the user must give a supervisory decision in executing the task. The structure of the model used for Task Manager is a bipartite directed acyclic graph. The defined structure is based on a task-method-task paradigm. In this manner, tasks are decomposed into methods, which in turn are composed of subtasks. This task-method-task structure is shown in Figure 14.

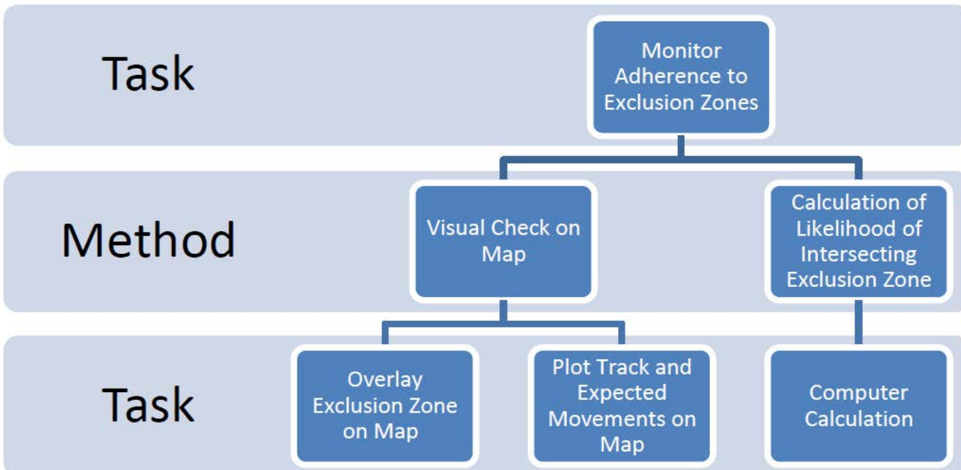


Figure 14: Task-Method-Task Structure for Task Model

When a task is defined, it is either assigned to a human supervisor or an autonomous agent assisting the supervisor. In order to complete the task, the assignee must choose from one or more methods. The method may consist of one or more subtasks. This structure repeats recursively until no further subtasks are required to complete the task. The Task Manager utilizes this model when generating tasks. Each task may be broken up into one or more subtasks, while subtasks may be broken down into even more subtasks. Decision points occur when an operator must pick which subtask would be best utilized to complete a task. The method by which a task is executed is called a play. A play, in IMPACT, is a set of commands and guidelines the operator provides to an autonomous system. The successful execution of a play will complete a task or subtask.

3.5.3.2 Task Manager Software Architecture

Task Manager is an internal module of the IMPACT system and is written in the C# programming language. It utilizes the model-view-view-model (MVVM) software architecture pattern. MVVM provides a methodology to separate the HMI from the backend logic or data model. The layout of the HMI is provided by two Extensible Application Markup Language (XAML) outlines. The first is used to give the concrete layout of the Task Manager tile as seen in IMPACT. The second XAML file provides templates which can be applied for incoming tasks. The backend logic of the Task Manager consists of the following key components:

- Eventing
    - Provides notifications to other components regarding task status
    - Initiates functionality to add, remove, and move tasks from their respective repositories
    - Handles repeat task events
- Processors
    - Processes asynchronous events, such as chat messages
    - Compares chat messages to regular expression definitions
    - Extracts chat messages by header or by room type
    - Creates tasks and subtasks
- Tasks
    - Determines available plays for task type
    - Calls plays associated with task type
    - Populates play workbooks for specific tasks or methods
- View Models
    - The primary logical components
    - Facilitate the function and usage of many of the other components
    - Determines display and sorting of tasks in repository
    - Executes interactions between user and backend logic
    - Source of data bound to HMI
- Interfaces and Resources
    - Provide layout information to the HMI
    - Binds data from backend logic to the HMI

3.5.3.3 Task Generation

Determining an instance in which a task must be created requires the Task Manager to employ IMPACT's networking hub. Tasks can be generated in a variety of ways. For instance, a query can be sent to an operator regarding an asset or whether a UAV can fly into a restricted area. Every event in IMPACT is forwarded through the hub. By parsing chat and notification messages that pass through the hub, it is possible to ascertain the required information needed to generate tasks. Task generation follows the subsequent pattern:

- A chat message is received from a designated room in the IChat repository.
- The contents of the message are compared to a map of regular expressions.
- Each regular expression pattern provides an associated task definition.
- If the message matches to a regular expression pattern, a task is instantiated by providing the task definition. This task is referred to as the parent task.
- Any subtasks associated with the parent task definition are assigned to the parent task.

Keywords in the chat messages help to determine the task category which in turn helps to determine the play type. For instance, if a chat message is received from by the Task Manager with the following text, "Unidentified watercraft heading towards the shore (Boat Golf) at 30.427560, -87.145746," a task is provided for the operator such as "Provide Overwatch." This task can be completed by selecting a subtask to, "Call Point Search," or to "Surveil Watercraft." The current categories of tasks include intruder events, environmental events, vehicle failures, base defense events, Random Anti-Terror Measures (RAMs) and queries. Other important information gathered from the task generation include time the task was assigned, time the task was completed, sequence number of the task, and the priority associated with the task category.

A secondary method used to generate tasks is to parse message headers as messages pass through IMPACT's networking hub using the ZeroMQ protocol. This method is employed for tasks generated by the occurrence of constraint and ROZ violations. In these cases, the message origin does not come from a chat room, but directly from the IMPACT hub. Messages from the hub are parsed and filtered to search for a specific header, "json:ML:Global.Notify". When these messages are discovered, a task is generated in a similar manner to the pattern outlined above.

3.5.3.4 Play Calling from Task Manager Interface

An essential function of the Task Manager is allowing the operator the ability to call a play from a task listed in the Task Manager. Each task assigned to an operator in an IMPACT scenario requires the execution of a play for itself or a subtask in order to be completed. A task may have multiple play options available for execution. Many plays, such as queries, have a single associated option. Other plays, however, may have multiple methods in which it can be assigned. The plays associated with each task are determined by the Quick Reaction Checklist. When the user selects a play, the Task Manager will auto-populate a workbook that can be used to execute the chosen play. Play options are gathered from the metadata of the task and the workbook is spawned using IMPACT's internal workbook spawner. Spawning the workbook required usage of some of IMPACT's internal functions.

3.5.4 Capability Developed

By coalescing tasks into a singular point of reference, Task Manager helps to increase situational awareness, allows for quick actions to urgent events, and helps to focus many sources of information into a contained space. The user now has the ability to act upon tasks as they arrive or to act according to priority and Task Manager provides a means to quickly execute plays relevant to each task. As scenarios increase in scope and complexity, the role of the Task Manager will increase to better help balance workload, provide information, and efficiently execute actions.

3.5.4.1 Entry Point for New Technologies

Task Manager has added significant functionality to the IMPACT system. Two key areas have leveraged Task Manager as the entry point to introducing new and valuable functions that can greatly expand the capabilities of IMPACT. The first is the ingestion of new data messaging schemes. Specifically, Task Manager was used as the entry point to allow the ingestion of CBML data into the IMPACT system. This important advancement provides new avenues toward collaborations with research being conducted by our allies.

Task Manager can also be an entry point into introducing new concepts in HAT. For integration into IMPACT, we developed autonomous search and detection algorithms with the intention of having a human-in-the-loop interaction to enhance the algorithm's effectiveness. A scenario was developed where the algorithms were placed into an object detection operation. The idea being that one of the search algorithms would act as an autonomous agent sweeping an area for the target. An operator would be given the ability to interact with the agent, providing information regarding possible locations of the object and to increase or decrease the probability of a target's location at a given coordinate.

Incorporating an interface through the Task Manager was successful and permitted operator interaction with the autonomous system. This was accomplished by subscribing to ZeroMQ messages that passed through the hub. When a HAT message is discovered a task is generated in the Task Manager. Through the task pane, the operator is able to interact with the autonomous agent by sending messages to the agent regarding the search area.

3.5.4.2 Autonomous Assistant and Load Balancing

The latest innovations currently being incorporated into the Task Manager involve helping the operator balance workload by tasking an autonomous assistant with extraneous tasks. The autonomous assistant provides the following services:

- Receiving tasking in order to balance the load of the operator
- Executing the allocated tasking
- Alleviating operator tasking on lower priority or repetitive tasks
-

In the future, these responsibilities will expand to include:

- Providing the operator feedback on task statuses or reminders for high priority tasking
- Providing suggestions or data for decision points of crucial operator tasks
- Adjusting load balancing of tasks to fine tune load balance for the operator
-

The autonomous assistant designed to be tasked in two ways. First, the operator can manually assign tasks to the autonomous assistant by simply clicking a button. Tasks can also be reassigned to the operator by selecting the task from the Autonomous Assistant's task list. Second, the Autonomous Assistant may also be tasked via a simple load balancing algorithm. When the operator begins to be over tasked, lower priority tasks (such as queries) can be assigned to the Autonomous Assistant. When the Autonomous Assistant receives a task, it immediately executes the task in its queue.

3.6 Human Machine Interface Design

3.6.1 Motivation and Challenges

Current interface design approaches are insufficient to support future envisioned unmanned systems missions, in which a single operator will collaborate with autonomous systems to manage multiple heterogeneous unmanned vehicles. These approaches often emphasize vehicle control rather than accomplishing tasks or completing mission objectives, an approach that doesn't scale when an operator moves from controlling a single vehicle to controlling multiple vehicles. Existing approaches also provide little transparency into supporting autonomy, in contrast to Lee's guidance to convey the system's purpose, process, and performance (Lee, 2012). Moreover, current human-machine interaction is typically rigid and inflexible, failing to provide support for trusted, bi-directional collaboration and high-level tasking between operators and autonomy (Lee and See, 2004; Hooper, Duffy, Calhoun, & Hughes, 2015).

For joint human-autonomy teaming, the operator must maintain overall SA not only of system status and mission elements but also the intent of multiple systems themselves (Chen & Barnes, 2014). This includes providing the operator the status of the autonomy's processing and the rationale for its recommendations to help support a shared mental model of who is doing what (as well as when and why; Linegang, et al., 2006). An even more critical design challenge is efficiently supporting collaborative human-autonomy dialog (Gao, Lee, & Zhang, 2006), enabling the human operator and autonomy to suggest, predict, prioritize, remind, critique, and/or caution each other, especially in response to changing goal priorities and environmental conditions. The operator also needs the ability to drill into the autonomy, adjust its parameters, or override its operation (Calhoun, Goodrich, Dougherty, & Adams, 2016). This involves providing cognitive support and coordination mechanisms with respect to a skills, rules, and knowledge framework (Rasmussen, 1990).

Thus, improved controls and displays are needed to support operator and autonomy teaming. Taking into consideration the heterogeneous UxVs domain, these interfaces need to facilitate the retrieval of actionable information, generate shared awareness of operator and autonomy state/intent, and help heterogeneous members coordinate in task completion (Goodrich & Olsen, 2003; Ososky, et al., 2012). To ensure agility, the HMI must support a range of control options whereby the operator can, depending on mission demands, be ‘on the loop’ supervising UxVs as they autonomously carry out assigned tasks, as well as being ‘in the loop’, exercising tele-operation to precisely control a particular vehicle/sensor temporarily (Air Force, 2015). Moreover, several control modalities should be available for the operator to choose which is best suited for the task at hand (Oviatt, 1999; Draper, 2007). The interface paradigm also needs to support multi-UxV control to enable new capabilities such as wide area search cooperation, inspection with multiple perspectives, tracking of moving targets, and communication relay to mitigate intermittent communication issues (Eggers & Draper, 2006; Martinage, 2014). This effort aimed to develop a new interface paradigm that addresses the above identified challenges and enables agile teams to benefit from the autonomy technologies also being advanced in this effort.

3.6.2 Software and Hardware Acquisitions

The HMI were designed to complement existing controls and displays that constitute the basic Fusion simulation framework. All hardware and software acquisitions supporting the developed HMI are described earlier. Additional software was produced specific to the HMI as described below.

3.6.3 Development and Implementation

Development of the HMI approach relied heavily on cognitive task analysis data (collected from subject matter experts familiar with unmanned vehicle operations and/or base defense missions), information control and display requirements identified in analysis of the tri-service challenge scenario, and the capabilities afforded by the autonomy components (especially the intelligent agent). Also, perspectives addressing issues impacting human-autonomy teaming were considered (see Klein, Woods, Bradshaw, Hoffman, & Feltovich, 2004; Woods & Hollnagel, 2006; Chen, Barnes, & Harper-Sciarini, 2011), as well as established human factors and ecological interface design principles (e.g., Vicente & Rasmussen (1992) and Kilgore & Voshell (2014)) in determining the content, layout and interaction metaphor of the new interface paradigm. The majority of the interfaces were designed to effectively support human rule- and knowledge-based behavior, given that the autonomy was anticipated to handle the majority of vehicle movement that is traditionally associated with skill-based behavior (Vicente &

Rasmussen, 1992). That said, a wide spectrum of control methods was implemented ranging from tele-operated (mouse/keyboard) control to high level plays that define the actions of one or more UxVs (Miller & Parasuraman, 2007). Intermediate methods involved interfaces that support the human and autonomy working together, with both making inputs to collaboratively plan or complete a task or play. An adaptable automation control scheme (Opperman, 1994) was utilized that extended a play-based approach from an earlier effort involving single operator management of multiple air vehicles (Calhoun, Draper, Ruff, Barry, Miller, & Hamell, 2012; Draper et al., 2013). This design perspective is also more aligned to a mission- and team-centered approach whereby the human and autonomy collaborate in decision making and flexibly interact to share dynamic mission goals required for a base security defense scenario with multi-domain resources. Use of this design perspective and implementation plan was realistic and effective for implementing HMI that supported the goals of the effort.

3.6.4 Capability Developed

This section will briefly describe the HMI implemented to support the play-calling control method. Each sub-section will describe how the interfaces were employed with mouse and/or touch input. (For most manual inputs there was a companion speech command that, if uttered, resulted in auditory and visual feedback to confirm the command was recognized. Also, the speech command resulted in the same control action and visual/auditory feedback had a manual modality been exercised). In this brief overview, the symbology employed across the interfaces will first be described. This will be followed by an introduction to the interfaces by which the operator calls a play, indicating the task type and location, relying on the autonomy to specify all other play details. Next the interfaces that enabled the operator and autonomy to work together to specify other play details will be illustrated. This will include interfaces by which the autonomy's reasoning is communicated to the operator. Finally, interfaces that support the operator's monitoring of play status and progress will be described. For further information, see Calhoun, Ruff, Behymer, & Mersch (2017) and Calhoun, Ruff, Behymer, & Frost (2017).

3.6.4.1 Concise UxV/Play HMI Symbology

The novel displays and controls feature video gaming type icons or pictographs (Nakamura, & Zeng-Treitler, 2012) to communicate UxV goals/states/progress in a concise, integrated manner and support the human's direct perception and manipulation (Shneiderman, 1992). For instance, icons were designed to represent plays (and associated UxV type) for supporting the targeted base defense mission (Figure 15). The inner symbol (e.g., plus sign, line, and square) represent common mission requirements to surveil a location, route, and area with additional inner components for other base defense plays. The asset types are redundantly coded by shape and location on the outer circle. Employing the symbology across HMI and the multiple monitors in the IMPACT control station (Figures 1 and 2) supports maintaining the operator's visual momentum (Woods, 1984) as information is retrieved and integrated. Visual momentum is further aided by mapping color-coding to plays such that all symbology associated with each on-going play has a unique color. This also helps the operator maintain global perspective when discerning which UxVs on the map are coordinating on the same play.

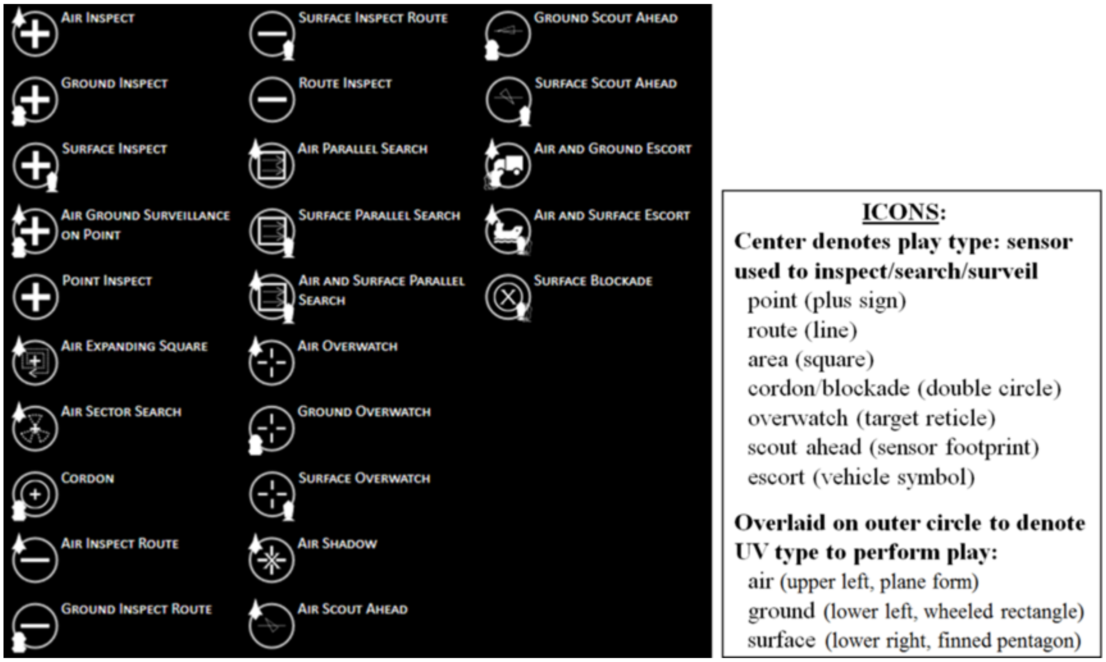


Figure 15: Icons that Specify Plays and UxV Types

3.6.4.2 Interfaces to Call Play Type and Location

After extensive analysis, it was determined that only a few operator commands are required for the majority of plays likely to be called for the targeted base defense mission. At a minimum, the operator needs to communicate what type of play needs to be accomplished as well as the location of the play. Given these two pieces of information, the autonomy can recommend which one or more UxVs should accomplish the play, as well as other play details (route, speed, etc.). Selection of the location for the play is accomplished either by making a designation directly on the map or identifying the location in a pull-down menu of the base's buildings and other landmarks. To specify what type of play, three dedicated play-calling interfaces were designed and implemented for mouse/touch manual input (Figure 16), in addition to the speech-based interface. With two of these, the operator interacts directly with the map to select either a location or a certain vehicle. This prompts a radial menu to appear on the map consisting of only the play options relevant to that location or vehicle (e.g., no ground based plays if a sea surface vehicle is selected). In a third interface, all available plays are available for selection. Play calling could also be achieved via control functionality integrated into the Task Manager that listed the pre-determined steps for most mission events communicated via chat (e.g., intruder at a certain gate).

Figure 16: Three Interfaces to Specify Play and UxV Type

3.6.4.3 Interface to Specify Play Details

Operator specification of play type and location supports a goal-based approach, with the operator expressing intent rather than directing the actions of individual UxVs. However, especially in light of dynamic missions, it is useful at times for the operator to communicate additional play requirements to the autonomy, either during play calling or after the play has started. Thus, interface mechanizations are needed whereby the operator can efficiently input any play related detail through a “Play Workbook Interface” (besides utilizing speech commands).

To accomplish this, the most likely details to be specified by the operator when calling a play, and also the most useful to the autonomy in terms of constraining the candidate solutions for the current mission situation, were identified. These were designated as “pre-sets” and were made the most accessible details in the Play Workbook. As shown in Figure 17, each of these was made available via a selectable concise icon in the right page of the Workbook. Selection options were grouped in rows as follows: size of the target, current environment around the target, optimization factor(s) to consider when proposing a plan for a play (e.g., minimize fuel usage or arrival time), and the play’s priority. The Workbook also provides quick access for the operator to specify a required payload (sensor and/or weapon), which are hard constraints that drive the autonomy’s asset assignment.

Besides the pre-sets described above, other play-related details are available on other Workbook pages. By changing between different pages of details via the tabs at the bottom of the right page, the operator can, for example, specify the loiter shape or change the allocated asset(s) that the autonomy recommended. Utilizing other control functionality, multiple plays can be chained together (e.g., each uses the same asset, with the second play commencing when the first one terminates or specifying the sequence for chained plays using different assets). Also, other temporal details can be specified like scheduling a play’s start time, end time, and/or duration.

Once the operator has designated play type and location, the autonomy determines and recommends at least one plan for the play (unless the operator’s constraints cannot be satisfied, such as there is no available asset with the specified payload). Via a Workbook selection or speech command, the operator can initiate the play. Alternatively, the operator can specify additional details either before or after the play has initiated that will prompt the autonomy to generate an updated play plan(s).

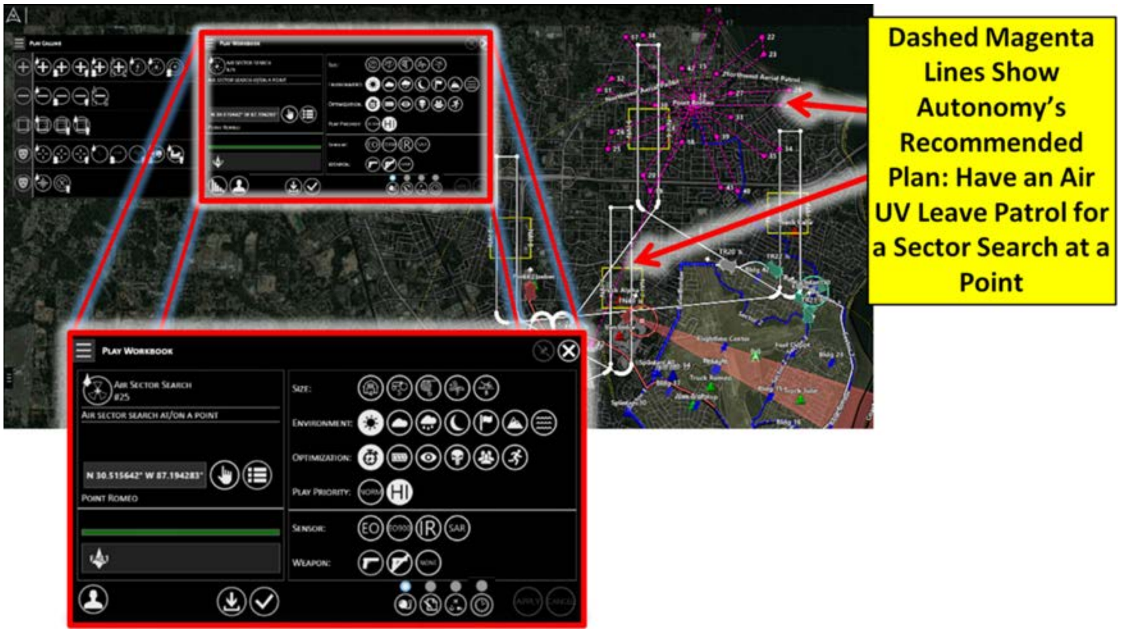


Figure 17: Play Workbook and Proposed UV Route on Map

3.6.4.4 Interfaces to Review Play Plans and Autonomy's Rationale

The HMI design enables the operator to review the basis of the autonomy's proposed play plan(s) and rationale. The icons highlighted in the Workbook communicate what constraints the autonomy considered in generating a play plan (Figure 17). The proposed asset(s) and route plans are also illustrated by uniquely colored symbology on the map (dashed until the plan is accepted) and rationale for autonomy's recommended plan can be accessed by opening a window adjacent to the Workbook (Figure 18). A Plan Comparison Interface can also be called up that illustrates trade-offs across multiple autonomy-generated plans as a function of several mission constraints (Figure 19; Hansen, Calhoun, Douglass, & Evans, 2016).

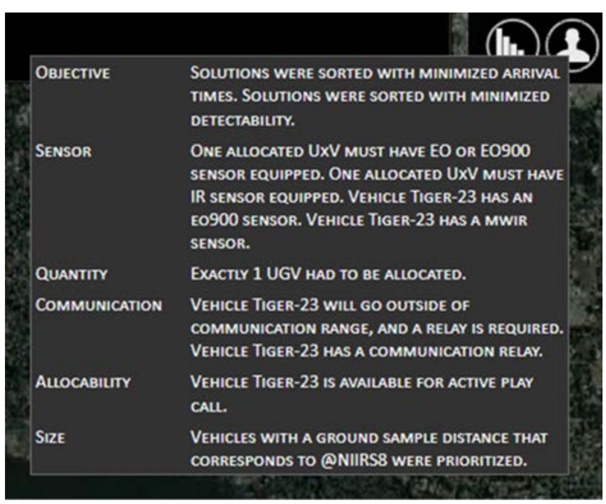


Figure 18: Window Showing Autonomy's Rationale for Proposed Plan

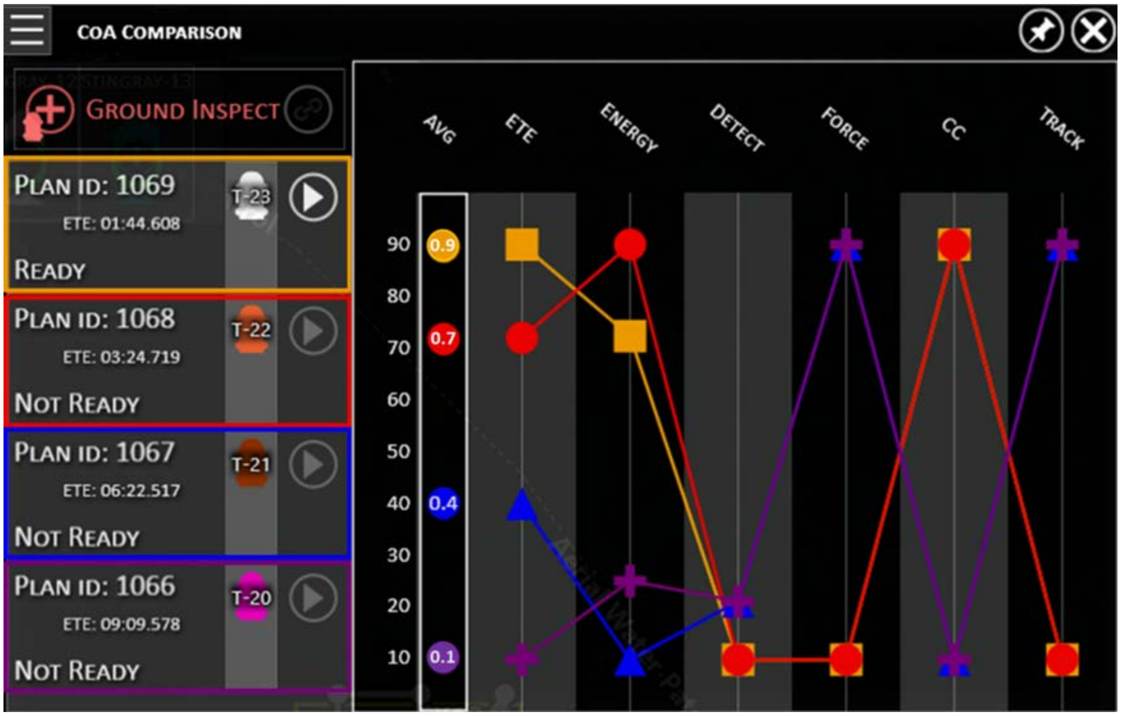


Figure 19: Plan Comparison Interface showing Trade-off of Several Candidate Plans

3.6.4.5 Interfaces to Monitor and Manage Multiple Plays

All active plays can be monitored by watching the movement of symbols representing each UxV along routes on the maps. Information is also presented for each play and ongoing patrol in a row within the Active Play Table (Figure 20), along with control functionality to cancel or pause each play. Selection of a row in the Active Play Table calls up the Workbook (Figure 15) associated with that play, as well as a Play Quality Matrix (Figure 21) that provides feedback on the ongoing play through autonomics algorithms. Deviation of each bar from the center of the matrix, as well as color (green, yellow, red) indicates whether the associated mission parameter is within, above, or below its expected operating range.

Figure 20: Active Play Table

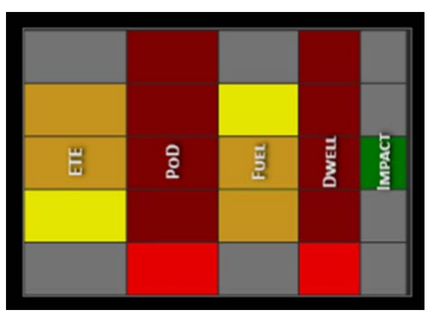


Figure 21: Play Quality Matrix

An Inactive Play interface consisting of two tables was designed to supplement the Active Play interface (Figure 22). The "Not Ready" table on the left contains plays that the operator has called, but the plays cannot begin yet because one or more constraints are not met (e.g., required sensor or UxV type not available). Plays that the operator has called with the intent of activating them later in the mission when resources are available are included in this table. In contrast, the “Ready” table on the right lists plays that have the required resources and are waiting for the operator to consent for the play to begin, or the operator has specified a specific time for the play to begin. The Ready Table also includes plays that were paused by the operator if the resources are still available.

The Active, Not Ready, and Ready Play Tables provide the operator with control functionality to quickly pause, initiate, and cancel plays, instead of calling up the Play Workbook to exercise these functions. The operator can also chain plays and designate a “Not Ready” play to automatically become active when an asset becomes available, without it first moving to the Ready Table to wait for the operator’s consent input.

Figure 22: Inactive Play Table Showing ‘Not Ready’ and ‘Ready’ Plays

3.6.5 Component Testing

A more detailed explanation of the HMI design process is available (Calhoun, Ruff, Behymer, & Frost, 2017). Early notional HMI concepts were illustrated in PowerPoint to support reviews accomplished by interface specialists trained in human factors and ecological design principles. These discussions resulted in some early concepts being discarded or refined. Next, a subset of control and display designs were mocked up in low-fidelity test apparatus and evaluated by participants without UxV or security mission experience. These experiments typically employed a single-task paradigm focusing on one aspect of play management. For example, there were individual experiments addressing: methods for communicating UxV play status (Behymer, Ruff, Mersch, Calhoun, & Springs, 2015), visualizations for operator comparison of autonomy recommended plans (Behymer, Mersch, Ruff, Calhoun, & Spriggs, 2014), and the design options for a video game inspired interface for calling plays (Mersch, Behymer, Calhoun, Ruff, & Dewey, 2016). Typically, candidate display formats were briefly presented and participants’ accuracy and speed in retrieving information or making a control input were measured.

The results of these “component tests” drove the interface designs implemented in the IMPACT virtual lab station and evaluated by subject matter experts (see Section 6). For the first full scale evaluation, four play-based interfaces were included (two play calling interfaces (speech and one manual approach), one interface for specifying play details, and one table showing active plays with the option of calling up additional play and vehicle status information). The design supported management of 6 UxVs with 13 plays besides the normal full coverage patrol. In the second full scale evaluation, there were four means of calling plays with manual inputs (in addition to speech-based control), a refined method to specify play details, a visualization depicting play status/progress, multiple tables showing the status of inactive, ready, and active play states, as well as two interfaces that provided additional insight into the intelligent autonomy agent's reasoning. These twelve interfaces were designed to support management of 12 UxVs, with 25 base defense related plays and two types of patrols. Section 6 provides additional detail on the methodology and results of these evaluations. In general, results were very positive, demonstrating that the utilized design approach helped ensure that the HMI reflect recommended human factors principles with the goal of better supporting how the operator and autonomy can jointly manage UxVs responding to mission events. In fact, one base defense expert commented that "each piece of the [interface] suite serves a purpose and is value-added for assisting the operator to accomplish the mission." The results also showed, however, that mouse and keyboard inputs were far more efficient manual inputs than use of the touch input modality (with the exception of zoom/map view manipulations; Calhoun, Ruff, Behymer, & Rothwell, 2017).

4 DETAILED TECHNICAL APPROACH: ADDITIONAL RESEARCH ACTIVITIES

4.1 Agent Transparency Studies

4.1.1 Motivation and Challenges

Past research has demonstrated that human operators sometimes question the accuracy and utility of intelligent agents when operators lack insight into the intelligent agent's rationale; this can lead to reduced use of the intelligent agent and subsequent loss of performance (Linegang et al., 2006). Researchers have suggested that, to support operator SA of the intelligent agent within a tasking environment, the agent needs to be transparent about its reasoning process and projected outcomes (Lee & See, 2004). To guide the development of agents that communicate transparency, Chen et al. (2014) proposed a model of agent transparency (Figure 23) to support operator SA: SA-based Agent Transparency (SAT). The SAT model uses insight from the theory of SA (Endsley, 1995), the BDI (Beliefs, Desires, Intentions) Agent Framework (Rao & Georgeff, 1995), Lee's 3P's (Purpose, Process, Performance; Lee, 2012), and other previous work (Chen & Barnes 2012a, 2012b) to guide the structuring of transparency information offered by intelligent agents. The first SAT level (L1) stipulates that the interface provide the operator with the basic information about system capabilities and limitations, current state, mission goals and intentions, and the agent's proposed actions (i.e., proposed plans or "plays" that can be executed to fulfill mission goals). At the second SAT level (L2), the operator is provided with the agent's rationale for recommending a particular play, including the weighing of capabilities and limitations, and the perceived trade-offs between different plays. At the third SAT level (L3), the operator is provided with information regarding the projection of future states and the certainty, or uncertainty, with which these projections are made. For the purposes of this project, transparency was operationalized and tested as existing at each of these three levels.

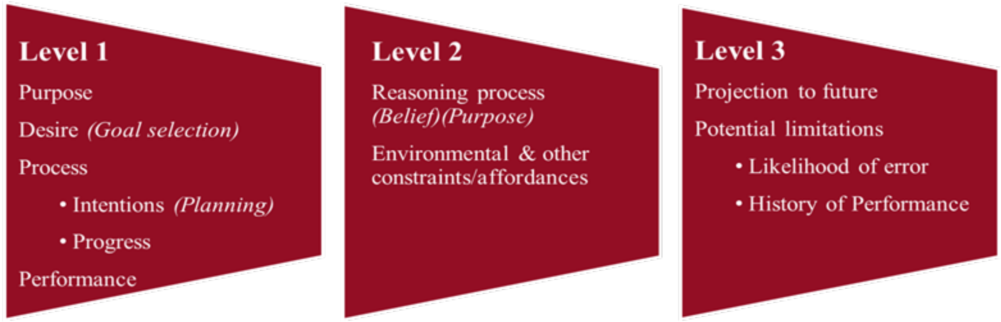


Figure 23: SA-based Agent Transparency (SAT) Model
(Chen et al., 2014)

4.1.2 Software Acquisition and Development

To isolate the effects of transparency at its various levels, a testbed was developed which emulated the Fusion testbed, but with hard-coded, modular components for manipulating the type and amount of transparency information shared with users. This testbed was used for all three years of experimentation, and allowed for yearly upgrading in order to meet requirements for testing the effects of transparency in a multi-UxV management task consistent with that studied by the IMPACT project as a whole. The final testbed can be seen in Figure 24.

Figure 24: Emulated Fusion Interface used for the ARL Experimental Studies

4.1.3 Development and Implementation

A total of three studies were completed to examine the effectiveness of agent transparency in facilitating performance and appropriate trust calibration. The first two of these studies manipulated only transparency according to the SAT model (discussed above), while the final study examined transparency framing in addition to a transparency level manipulation. Each of these studies are discussed below.

Studies 1 & 2: Methods. To investigate the three questions introduced in the "Background" of this section, we designed two experiments, each with three within-subjects conditions of transparency administered in blocks of mission scenarios (conditions were counterbalanced to prevent order effects; see Table 1 for conditions implemented). During the these experiments, participants assumed the role of a multi-UxV system operator whose task was to monitor and direct vehicles to carry out missions assigned to them by a simulated commander. Operators managed a team of six UxVs: two UAVs, two UGVs and two USVs, in collaboration with an intelligent agent which communicated play options to the operator for completing the mission. To complete missions, operators had to interpret their commander's intent, understand vehicle and environmental constraints, and ultimately, decide whether to follow the intelligent agent's play-calling suggestions.

During each of these decisions, operators' performance (based on the criteria in Table 2) and response time were monitored by the simulation. After each block of events, we surveyed participants for information including their perceived workload, perceived interface usability, and their trust in the intelligent agent.

Table 1. Transparency SAT Levels (SAT levels are additive).

Study 1		**Study 2**	
SAT Level	**Display Components**	**SAT Level**	**Display Components**
L1	Map icons, plan details icon, and path show basic information	L1+2	Map icons, path, line graph, and text show basic information
L1+2	Sprocket pie graph and text add reasoning information to display	L1+2+3	Sliding points on line graph and extra text add reasoning and projection
L1+2+3(+U)	Opacity of sprocket pie graph varied and extra text add projections including uncertainty	L1+2+3+U	Opacity of map icons and graph points varied, and extra text add assumptions and uncertainty

Table 2. Performance according to intelligent agent suggestion and operator choice of plans.

Performance Criterion	Correct Plan	IA Suggestion	Operator Choice
Proper IA Use	A	A	A
Correct IA Rejection	B	A	B

Study 1: Results. Results from study 1 (Mercado, et al., 2015; Mercado, et al., 2016) indicated that proper intelligent agent use and correct rejection were both significantly greater when participants were presented with SAT L1+2+3(+U) and L1+2 compared to L1. The greatest rates of proper intelligent agent use and correct rejection were found in L1+2+3(+U), suggesting that calibration of intelligent agent reliance is better when operators are presented with all three levels

of transparency by the agent. We found no significant differences for response time or workload, allaying concerns that higher levels of transparency might result in longer decision making time and greater operator effort.

Operator trust in the intelligent agent was analyzed after the first block of interactions as a between subjects variable, and examined it across two levels: the intelligent agent's analysis of the information, and the IA's ability to suggest and make decisions. There were no significant differences across SAT level for trust in the intelligent agent's ability to analyze information. However, we found that operator's trust in the intelligent agent's ability to suggest and make decisions significantly increased as transparency increased. Specifically, participants felt the intelligent agent made decisions that were more accurate when presented with L1+2+3(+U) as compared to L1+2 or L1. We also found a significant effect of SAT level on the perceived usability of the intelligent agent's interface; the intelligent agent was perceived to be the most usable when presented with L1+2+3(+U).
This study differentiated basic information (L1), reasoning (L2), and future projections (L3) in accordance with the SAT model. As such, we examined communication of the agent's projections and the agent's uncertainty in its projections as part of SAT L3 information level. However, due to this combination, the unique role of uncertainty in affecting operator decision was unclear. Study 2 filled this gap by parsing out uncertainty from other Level 3 information in the L1+2+3(+U) condition and adding another condition that included projection without uncertainty information: L1+2+3. For study 2, the L1 condition was eliminated (see Table 1 for condition listing).

Study 2: Results. Results from study 2 (Stowers et al., 2016; Stowers et al., 2017) indicated that proper intelligent agent use and correct rejection were both significantly greater when SAT L1+2+3+U was presented compared to L1+2. The L1+L2+L3 condition did not significantly differ from either of the other conditions. As L1+L2+L3 did not significantly differ from the low transparency condition without the addition of uncertainty information (+U), these findings support the conclusion that operators were most likely to make correct decisions when they were presented with all three levels of transparency as well as uncertainty. As was the case in study 1, operators did not experience greater workload as the amount of agent transparency information increased. However, unlike study 1, there was a significant difference in response time between L1+2 and L1+2+3+U (which corresponds to L1+L2+L3 in study 1), with L1+2+3+U taking the longest for participants to complete. This was not unexpected, as an increase in information on the display should naturally take longer to process. Though significant, this response time increase between the lowest and highest conditions was somewhat small (around 5.5 seconds). Contrary to study 1, in which we only analyzed trust after a single interaction with the interface, for study 2 we analyzed operator trust per condition as a within subjects variable while also controlling for the effect of pre-existing implicit associations. Unlike study 1, there was a significant difference across SAT level for trust in both the intelligent agent's ability to analyze information and the intelligent agent's ability to suggest and make decisions. Specifically, participants trusted the intelligent agent's ability to analyze information most when presented with L1+2+3+U, while they trusted the intelligent agent's ability to suggest decisions most when presented with L1+2+3. We also found a significant effect of SAT level on the perceived usability of the intelligent agent, where the intelligent agent was perceived to be the most usable when displaying L1+2+3 and the least usable when displaying L1+2+3+U. This perception is consistent with the participants' trust in the intelligent agent's ability to make decisions, where their trust and perceived usability peaked at L1+2+3 and decreased when uncertainty was added to the interface. This finding adds further support to the idea that usability may impact trust, or that there is at least a relationship between usability and trust regarding perceptions of intelligent agents.

Study 3: Transparency Framing. Agent transparency communication that draws attention to certain types of information could be thought of as a type of framing, or structuring of information. This framing may induce a bias in the operator's perception of the agent, which could be used to calibrate operator reliance on the agent. An example of this is attribute framing, in which an attribute of an object is described in either positive or negative proportions (e.g. the glass is half empty or half full). Historically, research has found that framing affects evaluations of objects, but it is not understood how the framing of something abstract, such as choice parameters, affect decision making. In the context of our study, where a human operator is making a decision (i.e. "play-calling" in the context of multi-UxV management) based on a set of parameters presented by an intelligent agent, positive or negative framing of the operator's decision by the agent may affect the human operator's trust and perception of the agent.
The overall goal of the final study was to understand the interaction between level of agent transparency communication, according to the SAT model, and the agent's framing of communication. We expected trust and evaluation of the agent to be higher with a high transparency interface than with a low transparency interface. When the agent is more transparent, and critical of the participant's plan decisions (critical framing), it should be perceived better and trusted more than a complimentary agent because it highlights reasons for error. On the other hand, we expected that when the agent is a more opaque, a complimentary agent would increase trust in the agent more than a critical agent would. An opaque agent provides less insight into possible shortcomings of its recommendations, which may result in lower perceptions and trust of the agent, but the agent's complimentary nature may help to offset negative evaluation of the agent.

Study 3: Method. To date, twenty-nine students from an American university were recruited for cash payment. Data were analyzed for 26 (17 men, 12 women, M_{age} = 20.03, SD_{age} = 2.09). Three were omitted from analysis due to technical issues.

This experiment involved a 2x2 mixed design with agent transparency as the within-subjects independent variable and communication framing as the between-subjects independent variable. Agent transparency was tested at two levels: (a) L1+2: containing reasoning information, and (b) L1+2+3+U containing reasoning and projection with projection uncertainty information. Communication framing was tested as two contrasting attitudes from the agent: (a) Critical: highlighting a parameter of the chosen plan that is not satisfied, and (b) Complimentary: highlighting a parameter of the chosen plan that is optimal. The HMI varied per condition by showing corresponding pieces of SAT-level information on a map display, in text, and on a sliding bar scale. Prior to the experimental trials, participants received about 1 hour of training. The experiment was divided into 2 blocks of 8 missions. Transparency order and communication framing were counterbalanced within sets of four participants, within which the scenarios where the agent's recommendations were correct and incorrect were held constant. The choice of correct and incorrect scenarios was randomized for each set but kept the 5 correct and 3 incorrect ratio. Performance and trust were recorded as done in studies 1 and 2. Additionally, perceived agent aptitude was recorded as a subset of trust.

Study 3: Preliminary Results. There were no significant task performances differences. However, for agreement with the agent, there was a main effect for transparency, as well as an interaction between transparency and framing. Agreement with the complimentary agent was consistent between transparency conditions. In contrast, agreement with the critical agent was higher in the low transparency automation condition than in the high transparency condition.

With regard to trust in the agent's ability to integrate and display analyzed information, survey results revealed a significant main effect for transparency. There were no other significant findings, but a trend suggests the possibility of an interaction between transparency and framing were there more statistical power. Participants were relatively distrustful of the low transparency complimentary agent. There were no significant findings from the trust survey with regard to the agent's ability to suggesting and making decisions.

There were main effects of both transparency and framing, and a non-significant interaction trend between the variables for perceptions of agent aptitude with regard to integrating and displaying information. Regarding perceptions of agent aptitude in suggesting plan decisions, there were also main effects for transparency and framing but an interaction was not significant ($p > .10$). In both cases, participants perceived the agent to be more apt when transparency was high and when the agent framed the update plan critically. For suggesting decisions, the interaction was driven by the difference between the low transparency complimentary condition and the other three conditions. There were no significant (or trends of) differences between condition in perceived automation reliability ($p > .10$).

4.1.4 Lessons Learned and Next Steps

Results from the 3 studies completed as part of this project yielded several insights to the utility of transparency, as well as best practices for implementing information transparency as part of an intelligent agent's interface. Primarily, it was found that agent transparency, as operationalized and implemented according to Chen et al.'s SAT model, is useful for improving performance in complex decision making such as that done in multi-UxV management tasks. Additionally, this performance is increased without a cost to workload. However, it should be noted that response time does increase for a few seconds, which may or may not create an issue depending on the mission environments.

The final study reported yielded insights to the possible importance of framing of transparency information. Participant agreement with the critical agent being higher in the low transparency automation condition than in the high transparency condition suggests that framing of information may be particularly important in situations when an agent is less transparent about its projection and uncertainty. Further examination of this effect in a separate study that isolates framing can yield insights to the effects of agent behavior on human decision making.

Overall, increased levels of transparency led to partially increased trust (in specific capabilities of the intelligent agent) in the first two studies, while there was an interaction between transparency and framing regarding trust in study 3. The findings in study 3 are particularly important to consider, as they show that a critical agent may be more trusted, perceived more positively, and agreed with more frequently. This shows that the way in which transparency information is presented can have an impact on trust calibration. Further examination of critical versus complimentary framing of information may show why this is the case.

The results of studies highlight the importance of considering variables in addition to performance (e.g. trust, workload) when studying human interaction with intelligent agents; especially as these variables may be useful for predicting the success an operator may have when interacting with intelligent agents. For example, if operators are over- or under-trusting, they may over- or under-rely on intelligent agents, to the detriment of the mission.

These studies have also highlighted that, in addition to considering transparency when designing intelligent agents, it is important to consider both the usability and behavior of intelligent agents in order to increase the likelihood of appropriate use and prevent undue burden from being put on operators. More research is needed to make refined design recommendations that incorporate usability and agents behaviors in this way. Furthermore, additional research is needed which explores human-agent teaming in a more bidirectional manner.

Future studies will examine human-agent teaming with bidirectional communications to evaluate the utility of SAT-based interfaces in a dynamic manner (Chen et al., 2017). This can be used to inform the design of field research being done with finalized intelligent agents that are capable of behaving independently. Additional efforts will also focus on developing a repository of HMI design elements (e.g., visualizations) to support SAT-based interfaces.

4.2 Human Workload and Attention Model Development

4.2.1 Motivation

The overall goal was to develop a real-time, online, predictive model of human operator automation monitoring. As automation capabilities increase, human operators need to be able to understand what the automation is doing while monitoring automation progress. This monitoring process can become quite boring when automation is performing well, but is a necessary job for humans who supervise automation systems. There are many instances of human operators not monitoring automation, which can lead to disasters.

The overall approach was a mix of experimental data collection and model building. A complex automated system, Research Environment for Supervisory Control of Heterogeneous Unmanned Vehicles was modified for these studies. In these experiments, the automation was helpful and successful much of the time (80% or more in some cases), but at times it behaved in an unexpected manner (e.g., it sent a UAV to an undesired location). Data was then collected on how human supervisors used the automation when it performed in an unexpected manner, how they dealt with automation failure, and how long it took them to identify and correct any automation failures. A critical aspect to this modeling work was the ability to measure where supervisor's visual attention was, so an eye-tracker was used to collect visual information while they were performing the task.

4.2.2 Software and hardware acquisitions

A series of models were produced that were able to predict when operators were likely to miss automation failures. These models have been implemented in both formal and computational forms.

4.2.3 Development and Implementation

A series of experiments were planned that explored how long it would take human supervisors to notice an automation failure, and then how they would correct the automation itself. A large amount of data was analyzed, including action protocols, eye-movement data, decisions, and reaction time.

From previous work in this area, it takes approximately three years to build a model of this complexity. Data collection, analysis, model building, and model validation are time consuming yet critical to the success of any model building effort. Given this expectation, the implementation was extremely realistic and effective.

4.2.4 Capability Developed

A model was successfully developed of when a human supervisor may be performing poor visual scanning. Poor visual scanning leads directly to missed automation failures, which was operationally defined as critical for this effort. This model is predictive, it runs in real-time, and has high accuracy. While a variety of different models were developed, one in particular will be highlighted. That model, called the meta-knowledge model, used three different predictors (last look, wait time queue, and available time; details available in separate publications). These three different predictors were able to predict 84% of missed automation failures (TPR) while only 3.8% were incorrectly categorized. These results reveal a *c* statistic of .97 (excellent) and a d' of 2.7 (excellent). These results suggest that the model is a viable model for automated system, since most automated systems need a d' of at least 2.0 to be functional.

4.2.5 Component Testing

Several individual experiments were conducted throughout the project. Over 200 participants were run across 6 different experiments.

4.2.6 Lessons Learned and Next Steps

There were several obstacles that were overcome. One obstacle consisted of the large variability of eye-tracking data. There is a known lack of research in how to account for eye-movements during dynamic tasks (e.g., where should a fixation be recorded as an object is moving?). Second, a real-time system to track and follow eye-movements needed to be created. Finally, determining how to create automation failures that were experimentally reasonable for the participants required some solutions. All of these problems were addressed through computational, modeling, and pilot testing methodologies.

4.3 Play Synthesis from Temporal Logic Specifications

4.3.1 Motivation and Challenges

The primary goal of the ARPI was to encourage development and deployment of high levels of autonomy in DoD systems. However, a practical barrier to deployment of such systems is a lack of feasible verification & validation (V&V) approaches. Current approaches often amount to exhaustively testing all possible system behaviors, which is intractable for highly autonomous systems. Therefore new approaches for V&V are required, e.g. based on mathematical analysis of system requirements and designs at various levels of abstraction, including the level of autonomous reasoning. One such class of V&V approaches focuses on automated synthesis of "correct-by-construction" system designs from specifications.

Toward this end, early work focused on automated verification and synthesis of mission plans for teams of unmanned vehicles. In this work, methods were developed to express mission requirements as temporal logic specifications, check that specifications are satisfiable (Kim & Humphrey, 2015), then either automatically verify human-generated plans against them or synthesize plans guaranteed to meet them (Humphrey, 2014). With IMPACT, these concepts were extended to address several new challenges including:

1. Increasing the reactivity of plays by synthesizing decision logic to change play behavior in response to mission events, with desired behaviors encoded in formal specifications.
2. Increasing synthesis ease of use by providing a template-based approach for developing formal specifications.
3. Implementing synthesis in IMPACT by integrating with the intelligent agent and UxAS.
4. Developing a design approach that supports V&V of autonomy.

4.3.2 Software and Hardware Acquisitions

SLUGS (SmalL bUt Complete GROne Synthesizer) was used to perform synthesis from GR(1) specifications. SLUGS is available at https://github.com/VerifiableRobotics/slugs and is free to use.

4.3.3 Development and Implementation

The IMPACT ARPI brought together several groups with different approaches for implementing autonomy. During the early stages of IMPACT, we considered formalizing specifications for plays in temporal logic and synthesizing plans for play-based missions using an approach similar to the one developed for planning multi-vehicle surveillance missions (Humphrey, 2014). This approach encodes mission goals and constraints in linear temporal logic (LTL) and uses model checking to synthesize a feasible plan. LTL extends propositional logic with temporal operators according to the grammar

$$\varphi \coloneqq true \mid a \mid \varphi_1 \wedge \varphi_2 \mid \neg\varphi \mid \bigcirc \varphi \mid \varphi_1 \cup \varphi_2$$

where a is an atomic proposition that evaluates to $true$ or $false$. LTL formulas include standard and derived propositional operators, e.g. $\wedge$ "and", $\vee$ "or", $\neg$ "not", and $\rightarrow$ "implies". They also include standard and derived temporal operators $\bigcirc$ "next", $\cup$ "until", $\square$ "always", and $\lozenge$ "eventually", where $\bigcirc \varphi$ holds if φ holds in the next state, $\varphi_1 \cup \varphi_2$ holds if φ_2 holds in the current state or some future state and φ_1 holds in all states until then, $\square\, \varphi$ holds if φ holds in the current and all future states, and $\lozenge\, \varphi$ holds if φ holds in the current state or some future state. LTL formulas can easily specify tasks that must be performed, constraints on the relative ordering of tasks, and conditions that should always hold or never hold, e.g. remaining inside "keep-in" zones or staying out of "keep-out" zones.

While synthesis from LTL specifications works well for certain applications, other groups had approaches that were better suited to the early needs of IMPACT. In particular, the intelligent agent framework provides a more elegant solution for determining which vehicles are most appropriate for a given play call, and UxAS is better able to account for vehicle dynamics in path planning and in plays that require multi-vehicle trajectory coordination. We therefore shifted in the second year toward alternative applications of synthesis in IMPACT, including synthesis of plays that react to human operator inputs during play execution (Feng, Wiltsche, Humphrey, & Topcu, 2015) (Feng, Wiltsche, Humphrey, & Topcu, 2016). However, such approaches require relatively detailed models of particular types of human behavior, which were not forthcoming.

In the middle of the second year, efforts shifted toward synthesizing plays that react to mission events. R*eactive synthesis* approaches were employed, focusing on synthesis from generalized reactivity (GR(1)) specifications. In general, reactive synthesis approaches focus on automatically generating decision logic that guarantees correct system operation in dynamic environments. Reactive synthesis specifications take the form

$$\varphi_e \rightarrow \varphi_s$$

where φ_e specifies possible behaviors of the environment, and φ_s specifies system behaviors that must hold given φ_e. In IMPACT, we took the environment to be certain unexpected mission events that were tentatively planned for the third year demo, e.g. vehicles running out of fuel, losing communication, and finding enemy targets. We then synthesized decision logic to change a vehicle's behavior in response to these types of events. The result is a "reactive play," which is in some sense implemented as an event-triggered "play of plays".

4.3.4 Capabilities Developed

In creating an approach for synthesizing reactive plays from temporal logic specifications, several new capabilities were developed. These include reactive plays, a template-based approach for developing temporal logic specifications, and an IMPACT-compatible implementation that makes use of the intelligent agent framework and UxAS. Furthermore, the overall approach contributes to the broader goal of V&V of autonomy. These points are described in greater detail in the following subsections.

4.3.4.1 Reactive Plays

As previously mentioned, reactive plays were synthesized from GR(1) specifications. Such plays react by automatically changing vehicle behaviors in response to events in the environment. Consider the situation in which an air vehicle should explore a region, track a target if it is found, and refuel if its fuel runs low, as depicted in Figure 25 and described in (Apker, Johnson, & Humphrey, 2016). We synthesized a play to implement this behavior and simulated it along with other more complex reactive plays in AMASE using the IMPACT framework.

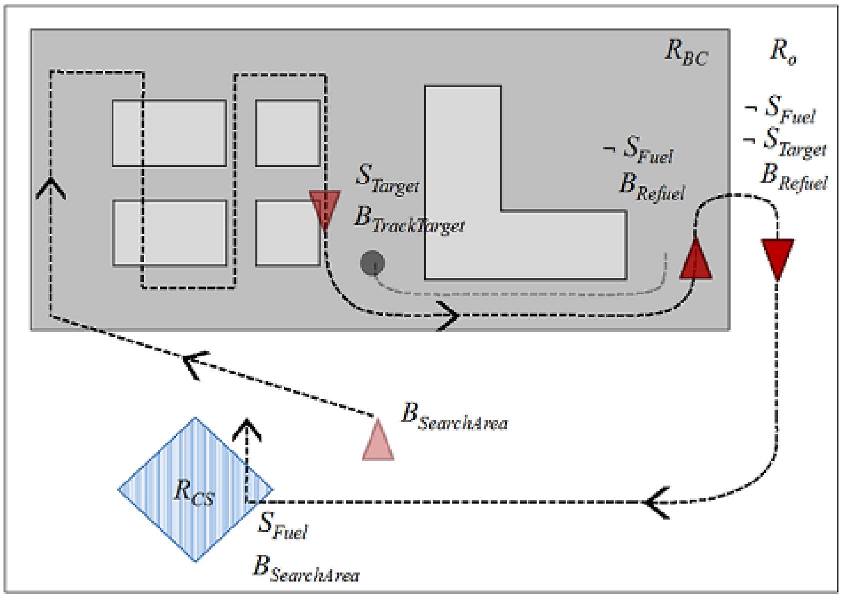


Figure 25: Behavior of an Example Reactive Play

4.3.4.2 A Template-Based Approach for Temporal Logic Specifications

A common criticism of temporal logic specifications is that it requires significant expertise to understand them. However, we found that many specifications of interest in IMPACT follow relatively simple patterns, leading us to formulate a pattern-based approach for developing GR(1) specifications. In particular, a common specification pattern includes a primary play that is executed in nominal conditions, a secondary play executed in response to a particular mission event, and a contingency play executed if something goes wrong. In (Apker, Johnson, & Humphrey, 2016), we developed a template-based approach for specifying the primary, secondary, and contingency plays and the events and conditions that trigger them, as in the example from the previous subsection.

4.3.4.3 Implementation with the Intelligent Agent and UxAS

The result of synthesis from GR(1) specifications is a control protocol that describes actions a system should take in response to events in the environment. A control protocol is a specific case of the underlying formalism used in the intelligent agent's behavior models. Simple routines were written to translate synthesized reactive plays into agent behavior models and we implemented supplementary behavior models to detect relevant mission events. When a relevant mission event is detected, a synthesized behavior model automatically changes the behavior of vehicles involved in the corresponding reactive play by calling the necessary route planning or inter-vehicle coordination tasks in UxAS.

4.3.4.4 Toward V&V of Autonomy

Many traditional approaches for verifying system safety involve identifying system hazards that could result in an operational environment and ensuring they have been sufficiently mitigated, e.g. as in fault tree analysis. Reactive synthesis provides an elegant method for mitigating identified hazards at the level of autonomous reasoning, as demonstrated by reactive plays that respond to events in the environment, including certain types of faults. This is only one example of ways in which reactive synthesis and other formal methods-based approaches can supplement existing verification approaches for V&V of autonomy. The use of formal methods also enables the use of automated verification tools that can, e.g., verify that components with formal specifications on their behavior will interact correctly in a larger “system-of-systems.” Given the complexity of autonomous systems, such automated approaches will be necessary to keep verification tractable.

4.3.5 Lessons Learned and Next Steps

There are many approaches for implementing autonomy. In a program like IMPACT, where several groups with different approaches come together to solve a concrete problem, it takes time to learn the strengths and weaknesses of each approach and determine which approaches are best suited to different aspects of the problem. To remain productive, some teams may have to shift away from their planned focus. Here, this team had to shift focus because the intelligent agent and UxAS were able to better provide many of the originally planned capabilities.

In the end, we were able to find a better fit for our approach, which in turn has given us ideas for future research. For instance, while our synthesis approach produces control protocols that are “correct-by-construction,” they must be translated into some implementation framework in order to actually execute. Full verification would then require verification of both the implementation framework and the translation processes, and methods to efficiently perform verification in such a situation are still needed. Another challenge is in debugging specifications for reactive synthesis, i.e. checking that specifications are realizable and that they capture the designer’s intent. This is much more challenging than debugging traditional system specifications, since reactive synthesis specifications involve both the system and its environment. Both of these areas would be interesting next steps for future research.

4.4 Machine Learning of Autonomous Vehicle Tactics through Human Evaluation

4.4.1 Motivation and Challenges

IMPACT as a whole is focused on tools and autonomy to aid a human supervisor in managing large teams of autonomous vehicles. Autonomy was designed to help the supervisor resource their plays, and to plan the routes for the vehicles. The underlying autonomy of the vehicles is assumed to be robust and efficient. This assumption however, is in doubt as we move towards a future where swarms, and dynamic environments require the use of machine learning techniques to develop the underlying autonomy of the vehicles. As the number of vehicles grows the workload on an individual operator will become ever more burdensome. To alleviate this, more robust autonomy needs to be developed, especially in the case of large numbers of drones acting in concert, such as a swarm. Often pure machine learning techniques can produce efficient behaviors, but those behaviors might seem foreign to the supervisors who must make the decision on whether to allow the vehicles to continue to operate. This research sought to examine the effect of Interactive Machine Learning on the trust of a supervisor by creating team behaviors that are more recognizable to a human operator.

The importance of understanding what algorithms are capable of doing is obvious when you are co-located with a potentially dangerous device. Thus for human-robot interaction, physical proximity creates a demand for high trust between the humans and the machines

(Groom & Nass, 2007; Lee & Nass, 2010; Nass, Fogg, & Moon, 1996). Less intuitively, trust in unmanned systems and autonomy is still needed when these systems are operated from a distance through command abstractions, such as supervisory control. Moreover, supervisory control is precisely where machine-learning algorithms should be leveraged in helping to determine the best mixtures of tasks, vehicles, and operator performance for mission success.

If implemented, machine learning will be difficult to supervise (Sheridan & Parasuraman, 2005), and calibrated trust will be nearly impossible to achieve as it relies critically on understanding the intentions and behaviors of the system (transparency – see (Sanders, Wixon, Schafer, Chen & Hancock, 2014)). Trust in automation is a complex research area, well summarized across several reviews (Lee & See, 2004; Sanders, Oleson, Billings, Chen & Hancock, 2011). Lee and See (Sanders, Oleson, Billings, Chen & Hancock, 2011) outlined three general bases for development of trust for automation in humans: performance of the automation (does it fail unexpectedly), process (whether the automation is understandable and fits well into the users workflow), and purpose (the automation functions as intended). Though purpose, process and performance can form the basis for trust, trust is still different from reliance (the choice to use the agent or automation.) For example, one can choose not to use a robot to perform a task, even though it could be very trustworthy; or vice versa, distrust a system but have no choice but to rely on it under certain circumstances, such as cognitive overload (Wickens, Hollands, Banbury & Parasuraman, 2013).

Often humans must rely on their perception of an automated system or robot's ability and behavior. The more obvious these abilities and behavioral intentions are, the more obvious failure states become. It is not that a system has to be perfect in order to be trusted, but it must be somewhat predictable; trust is more calibrated if one can "trust" automation to make certain kinds of mistakes (e.g. (Freedy, Devisser, Weltman & Coeyman, 2007)) but not others. With an opaque system, the operator cannot compensate for these faults (risking mission performance), in part because the expectancies surrounding failure conditions are not obvious. Calibrated and high-resolution trust is less likely because automation mistakes are not observable. Many have suggested increasing automation transparency is needed to improve teaming here; but the tradeoff with transparency in this case is that opaque systems may provide more optimal solutions. Neuroevolutionary computation (Gauci &Stanley, 2007; Stanley, D'Ambrosio & Gauci, 2009; Stanley & Miikkaulainen, 2002)) is one such method; the serious downside to neuroevolutionary computation is that it can result in "black boxes" from the human operator's point of view, which can make its application unsuitable for the real world. When applied to robotic plans, it may have the user asking questions like "What is this robot doing? What is it going to do? Why did it do that?"

The focus on increasing the optimality of these systems, largely performed in the domains of computer science and mathematics, generally ignores the need for user interaction. We attempt to mitigate the notable downside of generating black box solutions with new methods, as explained below, seeking to make their behavior more tolerable to the human supervisors who might oversee their operation.

4.4.2 Development and Implementation

4.4.2.1 Interactive Machine Learning (IML)

To improve the comprehension between the user and the evolved team behaviors we implemented an interactive evolutionary system. This system develops underlying team tactics that can be incorporated into plays by evolving neural network controllers for teams of vehicles. This neuroevolutionary approach allows for the creation of team behaviors that scale with the size of the team, while maintaining team symmetries and dynamics. The interactivity of the training process is accomplished by intermixing human choices into the evolutionary process periodically throughout the team's training. Our central hypothesis is that Interactive Machine Learning (IML) will develop behaviors (plans in this experiment) that adhere more closely to user goals and expectations. Plans should be more identifiable and trustworthy as a result. We focused on three questions: (1) does the incorporation of humans in deriving ML algorithms, through IML, lead to more human trust in the plans that are generated? (2) Do participants, who helped generate plans, recognize, and are they able to differentiate between IML and black box plans (which used neuroevolution, but no human involvement). Finally, (3) does the amount of neuroevolution that occurs, represented as steps, affect either trust or plan recognition?

To test this, we developed a simple 2-d kinematic simulator that allows a human subject to interactively train a small team of robots in the process of maintaining coverage over target areas. A research protocol was then developed, detailed below, that focused on addressing the three questions.

4.4.2.2 Experiment

Sixty participants (between the ages of 18 and 40) recruited from the University of Central Florida performed in the experiment. They received payment ($15/hr.) as compensation, in compliance with all Institutional Review Board statutes. The study lasted approximately 2 hours. Participants completed a trust in automation pre-experiment survey (Jian, Bisantz, Drury & Llinas, 2000); then they performed in three phases of experimentation: training, comparison, and labeling.

Training Phase: Participants were taught about the goal of 3 robots trying to search two areas effectively, and that the human role was to help train automated behaviors to maximize the amount of the area searched. A set of robot search agents in a virtual environment were shown exploring a space (Figure 26). Agents were autonomous and left signal decay trails in their wake, allowing participants to view how much of the targeted area had been searched.

Figure 26: Learning Phase: Multiple Teams in Action were Shown. A single choice was made.

Participants responded by choosing from these options a good behavior to evolve further. Participants were counter-balanced across the frequency of user input to be provided in IML (a decision, as in Figure 26, every 10 or every 25 steps of evolution). With fewer steps of evolution, the human has more "say" in the outcome. After making their selection, the algorithm evolved, and then new "plans" were presented as the next stages in evolution. Each plan had a fitness score associated with the algorithm, thus we were able to compare IML plans against black box plans. Participants responded through approximately 410 steps of evolution due to time constraints (about 40 points of interaction for 10-step, and only about 16 points of interaction for 25-step).

Comparison Phase: After training, participants were shown two teams in action. One of the two teams was IML and the other was black box, with the location of each team on the screen randomized (left or right). Plan pairs were chosen on the backend to equate fitness between them. When plans stopped participants selected the plan they believed would best cover the designated areas, and then made a response, 1-100, on a sliding trust scale indicating 1 for no trust and 100 for complete trust in the team plan they had chosen.

Labeling Phase: Following comparison, participants were shown a single team in action, and asked whether the team was IML, or black box. The interactive evolution teams were drawn from the specific individual's set of IML plans. Approximately 50% of each type of plan was shown randomly over 80 trials. Participants were given immediate feedback on their answers. At the end of the phase, participants were asked for their decision criteria for determining whether the teams in action had human IML, or were the evolved plans. Responses to the last question ended the experiment. Following, participants were debriefed and thanked for their participation.

Initial Results: Overall, participants chose the IML plans 66% of the time over the purely evolved plans in the head to head comparison. The IML and evolved plans had similar fitness so the user's choices must be based on characteristics imparted during the IML training phase. During the labeling phase, the participants are able to correctly label the plans as IML or Evolved 77% of the time. This suggests that there is something imparted from the interactive training that users are able to recognize.

4.4.3 Capabilities Developed

In the future where swarms, and dynamic environments will require the use of machine learning techniques to develop the underlying autonomy of the vehicles especially as the workload on an individual operator will become ever more burdensome. A system was built to meet this goal by the observation of machine evolved swarm actions and those that involved interactive machine learning where human feedback was injected into the generational cycles of evolutionary computation. It was shown via human subject testing that pure machine learning techniques can produce efficient behaviors, but those behaviors do seem foreign to human supervisors by a factor or two-thirds to three quarters of the time. This research illustrated the effect of IML on the trust of a supervisor by noting that human subjects more often selected IML over evolved results.

The hypothesis is weighted more heavily towards supervisory control with interaction by the human input into machine-learning algorithms in terms that should support the leveraging of the best mixtures of tasks, vehicles, and operator performance for mission success. The interactive evolutionary system via human subject testing results supports the intended notion that our hypothesis tends to develop behaviors (plans in this experiment) that adhere more closely to user goals and expectations. Plans should be more identifiable and trustworthy as a result.

4.4.4 Lessons Learned and Next Steps

Involving humans in generating neuroevolutionary behaviors for teams of agents (IML) resulted in behaviors that participants chose more often, and could be recognized. The importance of this first step is key, as it suggests that IML imparts traits to ML behaviors, which could be tuned to increase the expectancy and alignment of teams of machines. As mentioned in the introduction, this is a key limitation to employment. Despite their preferences, participants trusted IML plans slightly less than black-box plans, despite generally good trust of plans (M= 61).

From a methodological standpoint, the IML methods appears to have been effective even with small amounts of user involvement. Users may be imparting traits, correcting early, common "odd" behaviors of the algorithms, or possibly, it was their active involvement in the behavior development that made it familiar to them. No matter the explanation our work shows a hopeful avenue for exploration toward making otherwise opaque algorithms useful, and creating expectancies or familiarity for the user.

Although machine learning offers required advantages, it can be opaque to users and reduce their awareness, confounding C2. We have shown there is promise in interactive machine learning techniques that increase user selection of team behaviors compared to pure evolution alone.

4.5 Machine Learning for Task Generation Capability

4.5.1 Motivation and Challenges

The C2 of unmanned vehicles is a cognitively intensive task for human operators. The efficiency and success of the operator's performance often depends on a multitude of parameters, such as training, human abilities, timing and situational awareness. Humans are required to multitask in an uncertain environment, process situational data, and be able to efficiently utilize autonomous agents in multiple regions of interest. To improve operator's performance in complex C2 operations within the IMPACT environment, a machine learning model was developed that addresses these challenges.

This is accomplished by reviewing of incoming data from sensor feeds, chat messages and environmental events to find a "common denominator" for both the human and the autonomous agents. They all operate in the space-time domain; thus, it is important to know time, location, duration and the assets involved in the tasking of the autonomous agent or person. The data were compiled to a "human-agent interaction" (HAI) database as illustrated in Figure 27. In addition the modeling included a conversion of deterministic variables into a set of "soft" human based descriptors where the ranges are defined by the user. The tasking is then machine and put into a model to be used as inference rules.

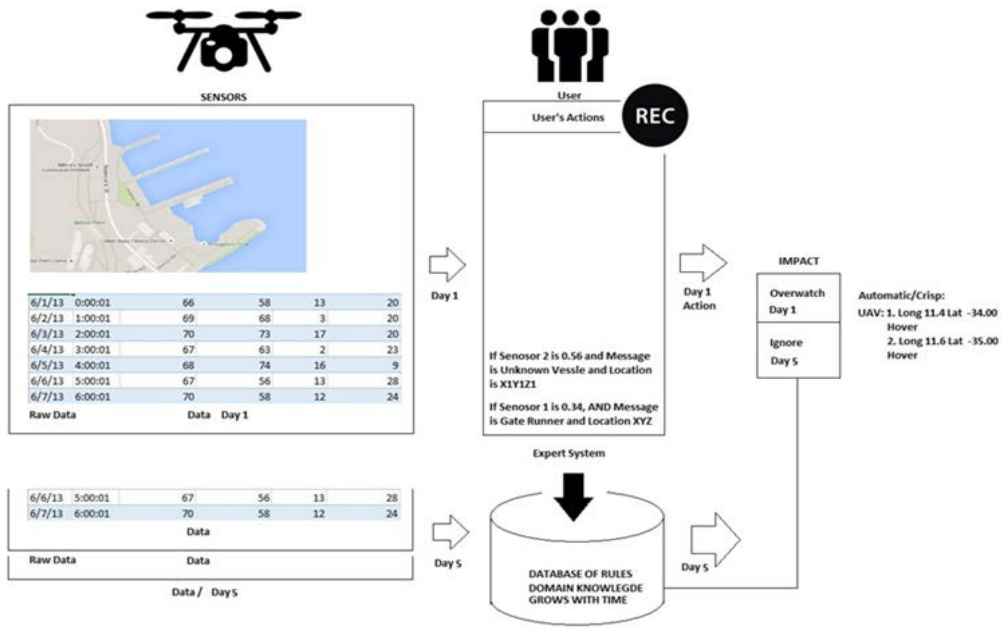


Figure 27: Machine Learning Model for Task Generation

The motivation for this model is to (1) optimize information in the time and space domain, (2) provide understanding to a human and autonomous agent, (3) make the process transparent to the human, and (4) improve the IMPACT system capabilities.

4.5.2 Software and Hardware Acquisitions

This research leveraged the IMPACT system and utilized the existing simulations. MATLAB, a software tool for technical computing with regards to algorithm development, modeling, simulation and prototyping was utilized. Two toolbox modules within the MATLAB, Machine Learning and Neuro-Fuzzy, were utilized in performing the machine learning aspects for the project.

4.5.3 Development and Implementation

Data from IMPACT simulations were utilized in the process and parsed to provide the tasking information as illustrated in Figure 27. Keywords from this and chat messages data are clustered as shown in Table 3. This approach measures effectiveness during the machine learning phase of the project as it makes sense of this data.

The model collects data from 4 inputs: sensor feeds, chat message feeds and events feeds as inputs and uses “tasking type” as an output. The data has previously been optimized for the time-space domain. A subsample of rules, IF-THEN rules, models the behavior environment in MATLAB in order to set the dynamics for the training method. The rules all have been modeled by a user to show possibility of optimization and machine learning capability for this approach. Once IMPACT simulations have expanded data seeds and the scope within and above simulations effort, and more data is provided, it will be possible to re-train the model and validate this approach for accuracy, consistency and computational power.

The developed model allows the use of either a “soft” linguistic term or deterministic data. This allows for a “neuro” part used to train the model on deterministic data, and the “fuzzy” use of soft-linguistic terms (Petrosyuk, 2016).

4.5.3.1 Foreseen/ Unforeseen Challenges

A few challenges due to missing and repetitive data sets that were encountered which included ill-defined or missing start and end points.

A separate issue arises from the question: how can we process and extract meaningful information form natural (chat) language data? This will most likely require development of a consistent method for processing chat messages. Experimentation with clustering algorithms and data feeds may provide information on statistics tasks metrics. These metrics are important in getting a clear understanding of how inference rules are cleared and can be investigated separately.

4.5.3.2 Capability Technical Approach

We have attempted to create a machine learning model based on the operator, handing of deterministic data from the sensors, the environment and the decision–making process of the IMPACT user.

All data in the IMPACT simulation are stored in states, which store the live data for all of the vehicles in the simulation. Other data comes from the sensors of the Unmanned Vehicles (UxVs). These data are stored as camera images, video streams, and radio state variables. Each vehicle state is comprised of many variables such as: current location, velocity, acceleration, current heading, available energy, energy usage rate, list of payloads, and current tasks. These data were used by machine learning techniques to determine what IMPACT based “play” should be generated.

The initial machine learning approach taken is the K-Nearest Neighbor algorithm (Fix et al., 1951) based on its simplicity and applicability to many problems. The high level approach is comprised of three steps:

1. Record the states of the IMPACT simulation when a task is created.
2. Continuously monitor the IMPACT simulation states.
3. If the current simulation state matches a state previously recorded when a task was made, then generate this task for the user of the Task Manager.

4.5.3.2.1 Task Optimization

The complexities of the IMPACT system can result in human operator information overload. The model used monitors queuing of tasks in IMPACT with the aim of reducing the operator’s cognitive load.

4.5.3.2.2 Data Collection

All vehicle raw data collection from the IMPACT system yields two types of data: user generated chat messages and sensor data which can be broken into four elements: time of message, location, duration and asset. Thus the data represented the time stamp when the message was issued, the region of interest (ROI) where an asset is planned to appear, the duration of completion after the trigger initiation and asset with sensor that can complete the task.

4.5.3.2.3 Complexity in IMPACT

For the user who simultaneously controls a number of autonomous agents, complexity means higher rates of multitasking. In this case, complexity reduction happens when many simpler tasking states are grouped together in a relevant sequence with proper timing.

4.5.3.2.4 Environmental Events

Environmental events take place outside of the user's control. These events trigger a user's reaction which will require actions in the IMPACT system to respond to the environmental events. Example environmental events include: a gate runner, mortar fire, and a user's observation of a chat message.

4.5.3.2.5 Sensor Data

Some of the variables in the IMPACT system include data that is supplied by a sensor from the unmanned vehicles (UxV). UxV's operate in a time and space domain and carry variable sensor performance characteristics (airspeed, energy rate, altitude, latitude/longitude coordinates, etc.).

4.5.3.2.6 Proposed Optimization Model for Machine Learning

Common attributes of the data presented in the summary above are space and time. Both the sensors and the IMPACT operators see information in the space and time domain. All events and tasks occur at a specific ROI and a point in time. The data-task optimization problem of the IMPACT system can thus be stated as follows: What is the least complex sequence of tasks that needs to take place to satisfy success of the outcome within a specified completion time? Three main variations of complexity settings designated as high, medium and low were utilized to categorize complexity. In the occurrence of a single event, the set of rules is straightforward but in the occurrence of simultaneous events the situation becomes rather complex. This led to being able to properly control and evaluate states under different complexities as a minimization-maximization problem. This led to an optimized tasking table in order to observe behavior of the separate UxVs and their task load in the time and space domain. Clustering the sensor and chat message status is the first step that can be taken towards reducing the operational complexity for the user and to investigate if such data can be used to model the control system under different complexity levels.

4.5.4 Lessons Learned and Next Steps

An optimization approach has been developed that would allow the IMPACT system to perform tasking under different levels of complexity. The level of complexity was shown to depend on the number of users using the IMPACT system, the number of random events happening during scenario and frequency of such events. All of these factors contribute to operator overload. Minimization of complexity can be achieved by optimizing the IMPACT input- output space in the time and space domain.

5 TECHNICAL EVALUATIONS

5.1 Spiral 1 Evaluation

This section will briefly describe IMPACT's Spiral 1 evaluation that was designed to solicit feedback from UxV operators and base defense subject matter experts on the Spiral 1 IMPACT system (for a more in-depth treatment see Behymer, Rothwell, Ruff, Patzek, Calhoun, Draper, Douglass, Kingston, & Lange, 2017). For this evaluation, participants were trained on IMPACT's autonomous technologies and asked to manage six UxVs (three UAVs, two UGVs, and one USV) in support of a simulated base defense mission. Feedback was sought on the HMI candidate display formats, symbology, and input modalities (mouse, touchscreen, and speech recognition) as well as UxAS, IA, and autonomics framework. Subjective data were recorded via questionnaires and analyzed and additional data were collected on the modality participants used to call plays. The results of this evaluation informed the development of the Spiral 2 IMPACT system.

5.1.1 Method

5.1.1.1 Participants

Seven current or former United States Airmen participated in the study. Three participants had UxV operational experience (Predator, ScanEagle, Global Hawk, and Shadow) and four had experience in conducting base defense operations in deployed environments (Afghanistan, Germany, Iraq, Kuwait, and Saudi Arabia). All participants were male and reported normal or corrected-to-normal vision, normal color vision, and normal hearing.

5.1.1.2 Equipment.

The Spiral 1 IMPACT test bed consisted of six computers (a Dell T5610 & five Dell R7610s running Microsoft Windows 8.1). One computer ran IMPACT and the AMASE (AVTAS: Aerospace Vehicle Technology Assessment and Simulation - Multi-Agent Simulation Environment) vehicle simulation (used to simulate the UxVs). One computer ran the TOC and simulation for simulated entities in the sensor videos (Vigilant Spirit Simulation; Feitshans & Davis, 2011), three computers ran two simulated (SubrScene) sensor videos, and one computer ran an XMPP Chat server for simulated communications. The IMPACT test bed used four 27" touchscreen monitors (Acer T272HUL), a headset with a boom microphone (Plantronics GameCom Commander), a foot-pedal (for push-to-talk speech control), and a mouse and keyboard.

An overview of the IMPACT test bed used for the Spiral 1 evaluation is shown in Figure 28. Starting with the top screen and moving clockwise, the Tactical Situation Display provided a geo-referenced map with UxV locations as well as UxV-specific information (e.g., a UxV's current play, error indicators, a UxV's planned route, etc.). The Payload Management display showed available sensor feeds on demand. The Sandbox display was a workspace for the participant to call and edit plays without obscuring the current state of the world (which was always available in the Tactical Situation Display). Finally, the System Tools display contained chat windows as well as help documentation (e.g., list of voice commands).

Figure 28: IMPACT's Test Bed Interface

5.1.1.3 Procedure.

After completing a background questionnaire, participants were given an overview of IMPACT that described the project's goals and introduced the concept of play calling. Next, participants were seated at the IMPACT test bed and given an overview of their mission that included:

- A description of the UxVs they would be controlling, how each UxV, its route, and its sensor footprint were represented on the map, and the tasks that each UxV was responsible for performing in support of base defense operations.
- An overview of the base they would be defending including the base's perimeter, sectors, critical facilities, patrol zones, and the named areas of interests in the area immediately surrounding the base.
- An overview of their role as a multi-UxV operator supporting base defense operations that described that they would be assigning high-level tasks to the UxVs while the autonomous system components flew, drove, and operated the UxVs. Also, that they would be assigned tasks from their commander in a chat window and that they would have access to the UxV sensor feeds but it was not their responsibility to monitor them.

After the general overview of the IMPACT simulation, mission-related tasks, and input modalities available for play calling, participants received a detailed briefing on the play-related interfaces available in Spiral 1. Next, training focused on providing participants with experience with each input modality. Participants received 12 chat messages asking them to call a play using a specific modality (e.g., "Using speech, call an air surveillance at Point Alpha"; 4 plays for each modality).

Participants were then trained on how to specify constraints, vehicles, and details when calling and/or editing a play. For all three input modalities, Participants were instructed via chat messages to call a specific play (e.g., "Using speech, call an air surveillance on Point Alpha, set sensor to EO, and optimize for low impact"), then make edits to the ongoing play (e.g., "Change the loiter type to a figure 8"). If a participant made a mistake, the experimenter provided feedback and the participant tried again until he had successfully completed the correct action. On average, training lasted one hour and was followed by a short break before the experimental scenario.

The goal of the 20-minute experimental scenario was to provide participants with the opportunity to exercise all of IMPACT's capabilities within a realistic base defense scenario. Participants were informed that the scenario would begin with an UAV investigating a suspicious watercraft with all other UxVs on a high alert patrol. Table 3 lists the exact sequence of mission events that occurred. Participants were instructed to respond to each chat message by calling one or more plays that best addressed the event. Participants were free to choose which of the three input modalities to employ in completing each step of the play calling process.

Once the experimental scenario was completed (approximately five plays called), participants completed paper questionnaires on the overall IMPACT system and its components. Then a semi-structured interview was conducted to capture additional feedback on IMPACT and its associated technologies including the three different input modalities. The entire procedure lasted approximately 3.25 hours.

Table 3: Sequence of Mission Events.

Event	Description
1	Participant receives chat message from Sensor Operator that unidentified watercraft is a fishing boat.
2	Participant receives a chat message from Commander to resume normal base defense operations.
3	Intelligent Agent recognizes a serendipitous surveillance opportunity (a UAV is near a critical facility) and recommends a play (air surveillance at the critical facility) to participant.
4	Participant receives chat message from Commander to send a UAV to Point Charlie and to instruct other UxVs to go highly mobile in response to a patrol reporting smoke at Point Charlie.
5	Participant receives chat message from Sensor Operator to send UGV to Point Charlie for "eyes-on".
6	Participant receives chat message from Commander to provide UGV headed to Point Charlie with a UAV over watch.

5.1.2 Results

Of the seven individuals who participated in the study, six completed the training and mission in the allotted time. Due to unanticipated time restrictions, the seventh participant was unable to complete the study and was eliminated from the data analysis. Due to the small number of participants, UxV operators and Security Force personnel data were not analyzed separately.

5.1.2.1 Overall System.

Participants used a 5-point Likert scale (ranging from 1: No Aid to 5: Great Aid) to rate IMPACT's potential value for future UxV operations as well as the ability of IMPACT to aid operator workload and SA (see Figure 29; note that the vertical line is the scale's midpoint, so all ratings to the right are above a 3). Participant responses indicated that they had a positive opinion on the potential value of IMPACT for future UxV operations, to aid workload, and to aid SA, with no ratings less than a 4.

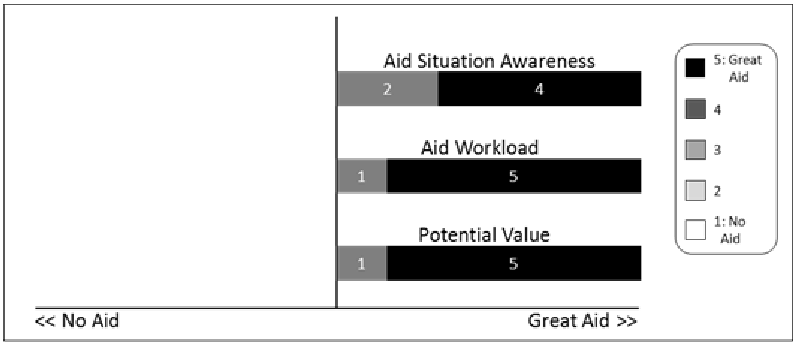


Figure 29: Participant Ratings for IMPACT's Ability to Aid SA, Workload, and Potential Value *(Every participant rated IMPACT at a 4 or higher for each of the three measures. The numbers inside the bars indicate the number of participants who provided a rating at that scale value.)*

The overall usability of IMPACT was assessed using the SUS (Brooke, 1996). The SUS asks participants to evaluate 10 items related to system usability using a 5 point Likert scale ranging from Strongly Agree to Strongly Disagree (see Figure 31), and these 10 items contribute to an overall SUS score. Overall mean SUS score for IMPACT was 73.75, placing it in the 70th percentile of SUS scores.

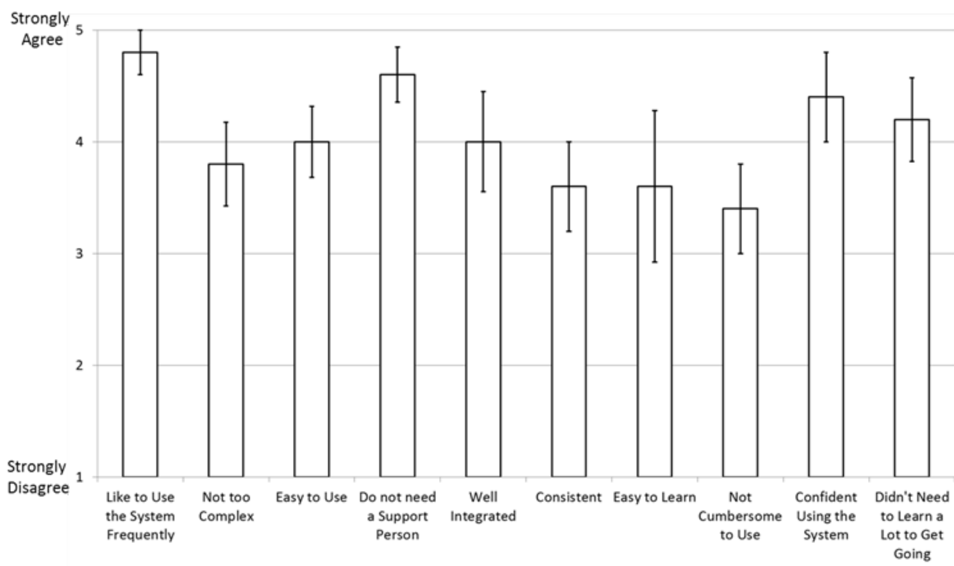


Figure 30: System Usability Scale Results (Error Bars = Standard Errors of the Means)

Participants were also asked what they most liked, what they least liked, what was most confusing, and what they would improve in regards to the overall IMPACT system (see Table 4).

Table 4: Participant Comments on the Overall System.

Most Liked	Least Liked	Most Confusing	Improvements
• Multiple modalities (e.g., speech, touch, keyboard and mouse) • Agent recommending plays when a UxV is near a critical facility • Play Creation tile's intuitive symbology • Ability to maintain big picture on top screen while zooming in and planning on bottom screen (sandbox)	• Touch screen wasn't precise enough; difficult to select correct button • Cannot manually draw routes for UxVs • Displaying all UxV routes made map cluttered • Requiring confirmation to execute a play	• Difficult to determine where a specific UxV was going due to map clutter • Challenge to learn how play icons were organized in Play Creator tile	• Ability to call plays by clicking locations/ vehicles on map • A single ear headset would be more comfortable and help maintain SA • Expand voice commands to more than play calling • Forecasting capabilities (e.g., what are things going to be like in 10 min.)

In addition to rating the overall IMPACT system, participants were asked to rate four system components (Play Calling, Autonomy, Feedback, and Test Bed) on five parameters (Potential Value, Ease of Use, Integration, Consistency, and Ease of Learning) and provide any comments they had about each component. Overall, 88% of ratings were either a 4 or 5 (the top two categories) and only a single component (Ease of Learning) was rated less than a 3 (by a single participant).

5.1.2.2 Play Calling Modality.

The feedback on touch and speech was mixed; in general, participants seemed to like the idea of being able to execute plays via touch and speech. However, participants expressed concerns about the touchscreen's calibration and lack of precision (a participant might touch an icon three times before the system registered it) and the speech system's poor accuracy (the word error rate was 21.95%). Objective data was also collected on the modality (mouse, touch, or speech) that participants used to call plays during the mission (when participants could choose the modality). Though several participants had positive comments about speech and touch, participants tended to use the mouse more than touch or speech (see Figure 32 – note that speech is labeled speech/mouse because when participants used speech during the mission they always used it in conjunction with the mouse. For example, a participant would initiate a play call with a speech command but execute the play by clicking the checkmark with the mouse instead of saying "Confirm" to execute the play by speech command). In fact, only one participant tried to use the touchscreen to call plays during the mission and only two participants tried to use speech. Participants also made a higher percentage of major errors (defined as failing to complete a play correctly) when using touch than mouse or speech.

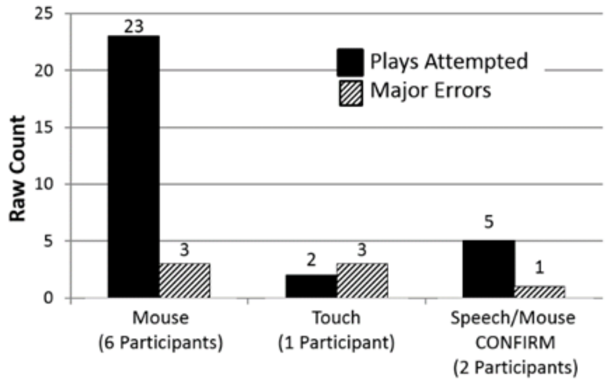
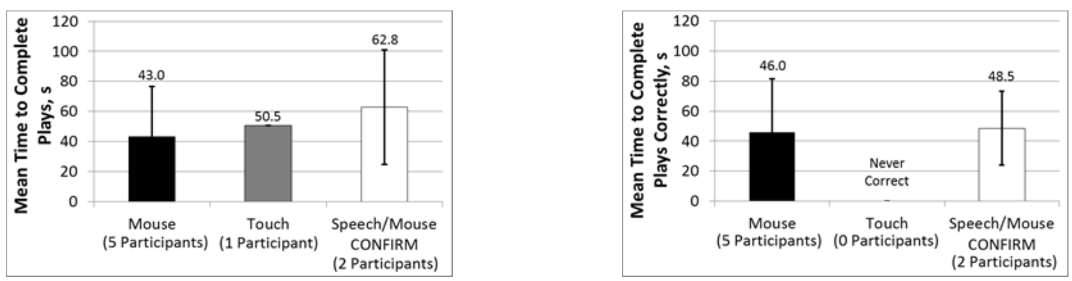


Figure 31: Number of Plays Attempted & Number of Errors by Modality

Participants were also faster at completing plays using the mouse as compared to using touch or speech (see Figure 33a). However, this difference most likely reflected the specific problems participants had with the touch (not precise enough) and speech (not accurate enough). In fact, when only correctly completed plays (i.e., no major errors) were examined the difference between time to complete a play with the mouse and speech was only 2.5 seconds (see Figure 33b – note that participants never correctly completed a play using touch).



(a) *(b)*

Figure 32: Mean Time to Complete Play Call by Modality
a) all plays b) plays called correctly (Error bars = standard deviations)

5.1.3 Discussion

This evaluation examined the usability of the IMPACT Spiral 1 system. Even though the Spiral 1 feature set was a subset of the Spiral 2 IMPACT system, five out of six participants strongly agreed that IMPACT has the potential to be a great aid in future UxV operations. Additionally, all participants agreed that IMPACT has the potential to improve operator SA and reduce operator workload. Participants rated both the overall IMPACT system and system subcomponents including play calling, autonomy, feedback, and testbed positively.

This study also examined the modality that participants used when calling plays. Participants overwhelmingly used the mouse compared to the touchscreen or speech recognition, and were faster and more accurate with the mouse. Several factors may have contributed to these results. Multiple participants had difficulties with the touchscreen registering their inputs. For example, it would often take a participant multiple attempts of touching a play icon before the system responded. In fact, some participants suggested in their comments that if the touchscreen had worked better, they would have been more likely to use it. Several participants spoke favorably of speech in their comments, especially the security force personnel, who mentioned that the speech commands were very similar to the dispatch calls they make during security force operations. However, this preference was not reflected in performance, as participants used the mouse/keyboard to call plays more than the speech. Several participants commented that they weren't completely familiar with the speech vocabulary, suggesting that training may not have been sufficient. In the end, participants may have chosen to use mouse/keyboard due to its reliability; clicking a play icon with the mouse consistently resulted in the desired action, while touching a play icon or issuing a voice command often failed to support task completion.

The biggest limitation of this study was the lack of objective measures; though participants provided positive subjective feedback in regards to IMPACT, the extent to which IMPACT improves participant performance was not ascertained in the Spiral 1 evaluation. The mission duration was also short, limiting the opportunity to investigate the degree to which participants could seamlessly transition between plays. Additional limitations included the small number of participants and the length of time (only ~3.25 hours) participants were exposed to IMPACT. Participants with greater experience with IMPACT may have been more comfortable using the touchscreen and/or speech recognition.

Participant feedback informed and improved IMPACT's Spiral 2 development. For example, participants expressed a desire to directly manipulate UxVs and call plays from the map, features that were implemented in Spiral 2. Participant feedback also generated research questions that led to additional empirical studies. For example, several participants felt the Play Creator tile could be improved by organizing play icons by vehicle type rather than by play type. A study was conducted examining the effects of icon organization and the results supported participant opinions; icons organized by vehicle type may improve a participant's ability to locate the correct icon (Mersch, Behymer, Calhoun, Ruff, & Dewey, 2016). Additionally, improvements were made to IMPACT's input modalities. For touch, the diameter of the play icon's selectable area was increased slightly (7.94 mm diameter compared to 6.35 mm in Spiral 1) and the touchscreen was replaced with a slightly larger one positioned at a lower tilt angle. For the speech modality, the finite grammar was dramatically expanded to allow hundreds more ways to say things, resulting in a large increase in flexibility and naturalness. Commands were also added to support a more complex mission (i.e., more UxVs, larger variety of play types, and ability to specify play details with speech).

5.2 Spiral 2 Evaluation

This section will briefly describe IMPACT's Spiral 2 evaluation (for a more in-depth treatment see Draper et al., 2017). Participants managed twelve simulated UxVs to support base defense operations. In order to demonstrate the effectiveness of IMPACT's autonomous system capabilities, this research compared IMPACT to a Baseline condition that represented the current state-of-the-art at the beginning of the IMPACT project. The Baseline condition had a subset of IMPACT's capabilities including the UxAS to assist in route planning and a HMI to interact with the UxAS. However, the Baseline condition lacked intelligent agents to support vehicle recommendations, the autonomics framework for plan monitoring, the task manager, and the voice recognition system. Operator performance and overall mission effectiveness were hypothesized to be significantly improved with IMPACT as compared to Baseline. Additionally, participants were hypothesized to prefer IMPACT over Baseline.

This research was also designed to investigate the extent to which IMPACT aids performance as the complexity of the mission increases. To this end, two levels of complexity were examined in the experiment, a low complexity mission and a high complexity mission. Operator performance was hypothesized to be worse in the high complexity missions as compared to low complexity, but the performance decrement was hypothesized to be less with IMPACT than with Baseline.

5.2.1 Method

5.2.1.1 Participants.

Eight volunteers with relevant military experience participated in this study, four active duty and four who had previously served. Six participants had prior experience piloting UAVs (Global Hawk, Predator, Reaper, Scan Eagle, Raven), one participant was a former Predator/Reaper SO, and one participant was an experienced security force and base defense expert. Seven participants were male (one female) and all participants reported normal or corrected-to-normal vision, normal color vision, and normal hearing. The average age of participants was 43.6 years (SD = 10.84).

5.2.1.2 Design.

A 2 X 2 within-participants design was used, with each participant experiencing both Baseline and IMPACT at two different levels of task complexity. The order of conditions was counterbalanced by tool and task complexity. In the Baseline condition participants had access to the UxAS and a HMI to work with the UxAS. The IMPACT condition had these features as well as an intelligent agent to support plan recommendations, plan monitoring, task manager, voice commands, and associated HMIs (Table 5).

Table 5. Differences Between IMPACT and Baseline.

Tool	Human Operator	HMI	UxAS	IAs	Monitoring	Task Manager	Voice
Baseline	X	Subset of IMPACT					
IMPACT	X	X	X	X	X	X	X

Task complexity was varied by a combination of increasing the number and complexity of RAMs the participant needed to complete during the shift, increasing the number of commander queries the participant needed to respond to, increasing the amount of noise radio and chat chatter (i.e., messages that didn’t require participant action), and increasing the number of events (normal base defense, intruder, environment, UxV faults) the participant encountered.

5.2.1.3 Equipment.

The experimental configuration used in this study consisted of four stations, the C2 Operator Station, the Sensor Operator Station, the TOC, and the Simulation Station (see Figure 34). The Simulation Station used a Dell Precision T5400 running Microsoft Windows 7 and OneSAF, a simulation tool that generated all friendly, neutral, unknown, and hostile forces during the experiment, with the exception of the UxVs. The C2 Operator Station and TOC each used a Dell

Precision T7910 while the Sensor Operator Station used a Dell Precision T5600; all three ran Microsoft Windows 8.1. The C2 Operator Station, Sensor Operator Station, and TOC had identical monitor setups, with one Sharp PN-K322B 4K Ultra-HD LCD Touchscreen (3840 x 2160 resolution) and three Acer T272HUL LED Touchscreen (2560 X 1440). Three Dell Precision R7610 running Microsoft Windows 7 located in a different room provided the sensor feeds for the UxVs (four feeds per machine). SubrScene, an in-house simulation visualization toolkit was used to provide the sensor feeds.

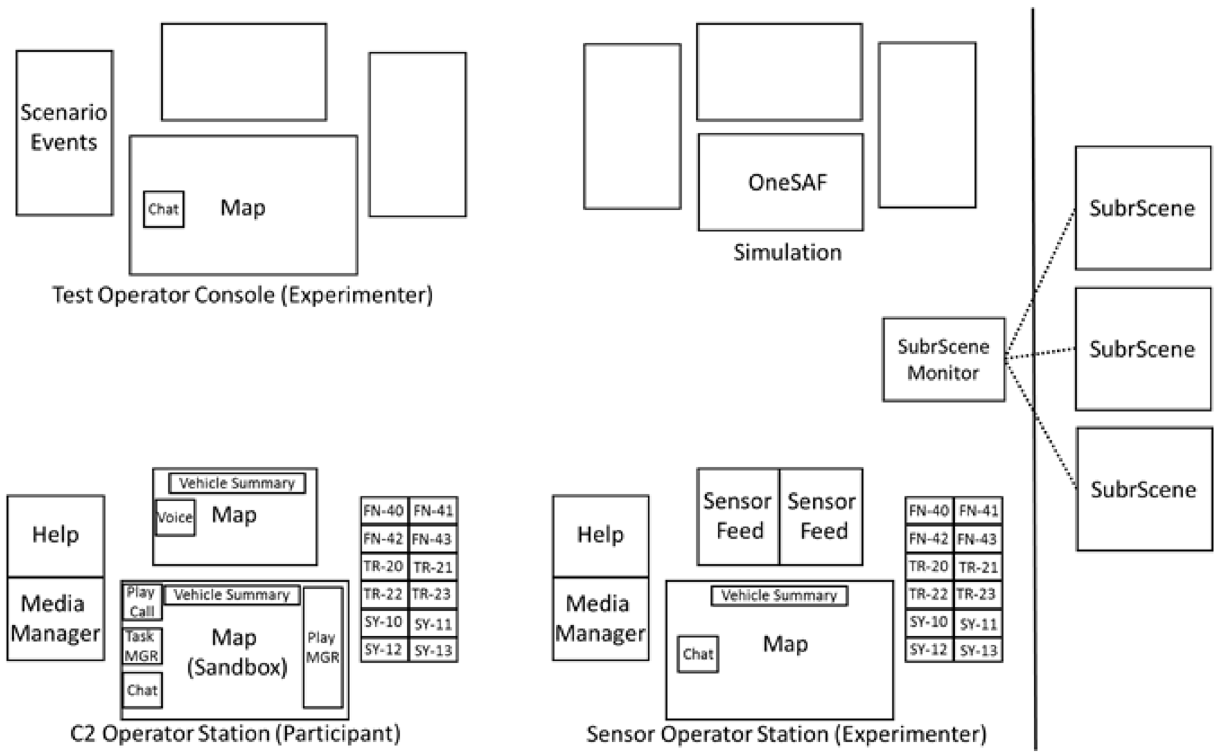


Figure 33. Experimental Configuration

5.2.1.4 Scenario.

During the mission participants were placed in the role of an operator managing twelve UxVs (four UAVs, four UGVs, and four USVs) to support base defense operations. The participant's job was to use the UxVs to accomplish tasks in response to requests by his or her commander which were generated from the TOC via pre-scripted chat messages. The participant had access to the UxV sensor feeds but it was not his or her responsibility to monitor them; that role was performed by the Sensor Operator, who was played by one of the members of the experimental team. The participant's main task was directing and monitoring the UxVs in response to various events. For each event participants had a quick reaction checklist available in the Help file that listed the correct response for that event. Events that could occur included patrols, RAMs, normal base defense events (e.g., responding to alarms, investigating suspicious vehicles), and intruder events (e.g., gate runner, mortar fire). In addition to these events, participants also had to respond to queries from their Commander via chat. Example queries include: What's FN-42's

Altitude? How long would it take to get a Show of Force at Gate 3 in place? How many RAMs have been completed? Participants also had to respond to vehicle failures (e.g., sensor malfunctions, engine failures) and environmental events (e.g., restricted operating zones, dense smoke).

Four experimental scenarios were used, two low complexity scenarios and two high complexity scenarios. Each type (low and high) were matched, so that if an event occurred in the first low complexity scenario (Scenario A), an equivalent event happened in the second low complexity scenario (Scenario B) at the same time. For example, at seven minutes into the mission the participant was asked to investigate an unidentified watercraft in Scenario A and a suspicious vehicle in Scenario B. Each experimental scenario was 60 minutes long and had an initial period of normal base defense operations lasting about 30 minutes, followed by an intruder event that lasted about 15 minutes, followed by a resumption of normal base defense operations for the final 15 minutes.

During the mission, the SO (played by a member of the research team) acknowledged participant actions and took images from the sensor feeds as required. For example, if the participant called a point inspect play to investigate an unidentified watercraft, he or she would radio the SO and inform him or her of the play and the SO would acknowledge this play via the radio and then take an image of the watercraft once the UxV arrived and send it to the participant. The SO, depending on what the script called for, either gave the all clear after the image was taken (thus implicitly instructing the participant to return the asset to patrol) or stated that the all clear had not yet be given (thus implicitly instructing the participant to keep the UxV on task). If the participant asked the SO about the status of non-SO related task, the SO would advise the participant to check his or her chat. For example, if the participant asked the SO if the unidentified watercraft had been imaged, the SO would reply with a yes or no. If the participant asked if the gate runner had been apprehended the SO instructed the participant to check his or her chat window.

The TOC operator, played by another member of the research team did not interact with the participant during the mission. However, the TOC was responsible for ensuring pre-scripted events occurred and injecting pre-scripted events as needed. For example, if the script called for the UxV that was conducting a point inspect to lose its sensor feed, it was impossible to know which UxV the operator would assign to the task a priori. Once the operator had selected the UxV for the task, the TOC would disable that UxV’s sensor feed on the fly.

5.2.1.5 Procedure.

The study took place over two days. On day one participants were trained on the base defense mission as well as how to use Baseline and IMPACT. Participants completed experimental trials on day two.

Day 1: Training. Participants were briefed on the goals and purpose of the study, signed an informed consent form, and were given a safety briefing. Next, participants completed a background questionnaire that collected basic demographic information (age, gender), unmanned vehicle operations experience, manned flight experience, and base defense experience. Participants were then seated at the C2 Operator Station to begin training. Training consisted of the lead experimenter instructing the participant how to perform specific actions using IMPACT and Baseline. Once a particular capability or function had been trained, the participant was sent a series of questions/tasks to accomplish via chat to ensure that he or she had been sufficiently trained before moving on to the next topic. For example, once a participant had been trained on how to manipulate the map, chat messages were sent asking him or her to pan, zoom, and rotate the map using the mouse and the touchscreen.

Participants were first trained on the base defense mission they would be supporting, beginning with a description of the base and their role in the upcoming mission. Participants were also given a paper map of the base that they had access to for the duration of the experiment. The experimenter then provided a briefing on the capabilities of the UxVs and instructed participants how to compare across the UxVs to determine which UxV to use in specific environmental conditions, when a specific optimization (e.g., optimize for crowd control) was required, or when a specific payload was needed. Next, participants were trained on the types of tasks they were responsible for performing (e.g., patrols, RAMs, normal base defense events, intruder event, commander queries, vehicle failures, environmental events) during the mission and the correct response for each. Participants were also told how they would be evaluated for each task (e.g., "For commander queries, your performance will be evaluated on the time it takes you to respond as well as the accuracy of your response").

Once participants had been trained on the mission and the UxV capabilities, they were trained on the functionality shared across IMPACT and Baseline including map movement, map decluttering, chat, vehicle dashboard, vehicle summary panel, media manager, and help. Participants were then given a high-level overview of the autonomous systems, including the UxAS, the intelligent agent, the Plan Monitor, the Task Manager, and the voice recognition system.

Next, participants were trained on Baseline, and how to use the system to respond to each possible type of mission event. After a break, participants completed a sixty minute Baseline capstone mission, equivalent in complexity to a low complexity experimental scenario. During the capstone mission the lead experimenter answered any questions the participant had, pointed out any errors the participant made, and suggested better methods for accomplishing tasks. After the mission, participants filled-out a digitized version of the NASA-TLX in order to understand what to expect during data collection trials.

After a break for lunch, the participant was trained on IMPACT, including the voice recognition system, the Play Calling interface, the Play Workbook, the Active Play Manager, the COA Planner, and the Task Manager. Participants were then given an opportunity to respond to each possible mission event using IMPACT. After a break participants completed a sixty minute IMPACT capstone mission, equivalent in complexity to a low complexity experimental scenario. Just as during Baseline capstone, the lead experimenter answered any questions the participant had, pointed out any errors the participant made, and suggested better methods for accomplishing tasks. Once again, participants completed the NASA-TLX after the mission.

Day 2: Experimental Trials. On the second day, participants completed four sixty minute experimental trials blocked by system. Participants were given refresher training before each block. Refresher training consisted of the participant being asked to respond correctly to chat requests for each RAM, each normal base defense event, each intruder event, each commander query type (time to get a specific vehicle to a specific location, time to get a specific capability to a specific location, time to get a specific task in place, vehicle speed, vehicle altitude, what a vehicle was doing, and what vehicle was doing/had done a specific task), sensor and vehicle failures, environmental events, and ROZs.

Once refresher training was completed participants were given a paper copy of the RAMs they were responsible for conducting in the first trial and given as much time as they needed to develop a plan for executing the RAMs. When the participant was ready, the lead researcher counted down ("Three, two, one, GO!") and the mission began. During the mission the lead researcher sat alongside the participant and observed his or her actions. The lead experimenter did not intercede unless the participant encountered a software bug or to prevent a participant from crashing the system due to a known bug. Both the lead experimenter and the TOC operator recorded how well participants did on each mission task to supplement Fusion's data logs. A software tool called Camtasia was used to record the Sandbox screen and well as all voice commands and radio calls the participant made during the mission.

5.2.2 Performance Measures

Subjective Measures. After each trial participants completed the NASA-TLX. After each block participants completed a tool specific overall questionnaire, a tool specific usability scale, and a tool specific component questionnaire. After the participant had completed both blocks, he or she filled out a questionnaire comparing IMPACT to Baseline across mission tasks.

Objective Measures. Performance data for each type of mission event (RAMs, Normal Base Defense Events, Intruder Events, Vehicle Failures and Environmental Events, and Commander Queries) were collected. For RAMs, participants were scored on how many RAMs they accomplished correctly (i.e., met all the constraints for) during the course of the mission. For Vehicle Failures and Environmental Events, participants were scored on how many they responded to correctly. Both accuracy and response time (the time the query was sent to the participant's chat window until the participant replied via chat) data was collected for Commander Queries.

For Normal Base Defense Events and Intruder Events both outcome (i.e., was the response to the event accomplished correctly) and process (i.e., did the participant select the correct location/target, correct play, the optimal vehicle, and meet the event's constraints) data were collected. For example, imagine that a participant, in response to a task to provide an escort for Convoy Kilo before Convoy Kilo left the gate at 22:00, called an overwatch for Convoy Kilo that wasn't in place until 23:00. In this case the outcome score would be 0 because the participant called the wrong play (an overwatch instead of an escort) and was late getting the play in place. However, the process score would be 0.5 because the participant called the play on the right target (Convoy Kilo) and picked an appropriate UxV (each component of the process score was equally weighted). Analyzing the data in this fashion provided information that was both mission relevant (i.e., did the mission get accomplished?) and diagnostic of where the process may have broken down (e.g., participants often failed to respond correctly to a specific event because they had trouble identifying the optimal vehicle to use).

Response time was not analyzed for mission events because direct comparisons between conditions were not possible. For example, in the Baseline Low Complexity scenario a participant may not have even attempted to respond to a particular event, while responding to the same event in the IMPACT Low Complexity scenario. Instead, the time and number of mouse clicks from when a participant began to call a play (e.g., click a play icon) until the play was executed (e.g., hit the check mark) was analyzed.

5.2.3 Results

5.2.3.1 Subjective Measures

Overall Ratings. Participants used a 5-point Likert scale (ranging from No Aid to Great Aid) to rate IMPACT and Baseline across three measures: potential value to future UxV operations, ability to aid operator workload in future UxV operations, and ability to aid SA in future UxV operations. The data was analyzed using a paired samples t-test. IMPACT was rated significantly higher than Baseline for both potential value to future UxV operations, $t(7) = 3.99, p = .005, d = 1.99$ and ability to aid workload $t(7) = 5.35, p = .001, d = 5.86$ (Figure 34). In fact, all eight participants gave IMPACT the highest possible rating when asked about IMPACT's potential value to future UxV operations and seven out of eight participants gave IMPACT the highest possible score when asked about IMPACT's ability to aid workload in future UxV operations. No significant difference was found for the ability to aid SA, $t(7) = 1.49, p = .18, d = 0.54$ (Figure 34).

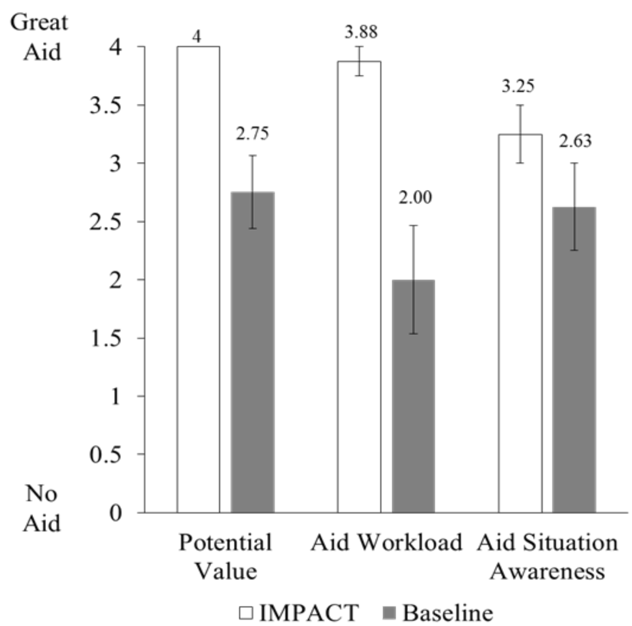


Figure 34: Overall Ratings for IMPACT and Baseline

System Usability. The overall usability of each tool was assessed using the SUS (Brooke, 1996). The SUS asks participants to evaluate 10 items related to system usability using a 5 point Likert scale ranging from Strongly Agree to Strongly Disagree, and these 10 items contribute to an overall SUS score. Participants rated IMPACT higher than Baseline on every single SUS item. The overall SUS scores were compared using a paired samples t-test. IMPACT's overall SUS score was significantly higher than Baseline's overall SUS score, $t(7) = 2.73, p = .03, d = 0.97$ (see Figure 35).

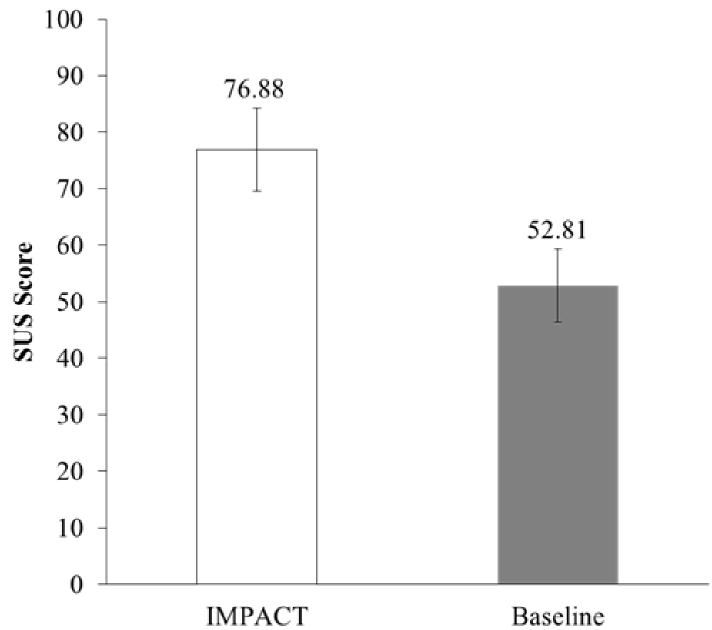


Figure 35: Mean SUS Score for IMPACT and Baseline

IMPACT vs. Baseline. After the experimental trials were completed, for each mission task, participants were asked to rate whether they performed the task better with IMPACT or better with Baseline. Participants rated their performance as better with IMPACT as compared to Baseline for every single mission task. Participants were also given the opportunity to comment on the differences between IMPACT and Baseline. Two participants elected not to comment. Of the remaining six participants, five were very positive about IMPACT as compared to Baseline. A single participant gave IMPACT a mixed review stating that although his or her performance was better with IMPACT, he or she had better SA and confidence with Baseline.

NASA-TLX Workload. Participants completed the NASA-TLX to assess their perceived workload after each experimental trial. Data were analyzed with a repeated measures Analysis of Variance (ANOVA). No significant interaction between tool and complexity was found, ($F(1,7) = 2.57, p = .15, \eta_p 2 = .27$). The results indicated a main effect of complexity ($F(1,7) = 17.06, p = .004, \eta_p 2 = .71$), with participants rating workload lower in the Low Complexity condition ($M = 36.72$) than in the High Complexity condition ($M = 58.54$). The results did not indicate a main effect of tool ($F(1,7) = 4.08, p = .08, \eta_p 2 = .27$, *IMPACT* $M = 43.28$, *Baseline* $M = 51.98$).

5.2.3.2 Objective Measures

Participants performed significantly better with IMPACT across multiple performance measures including the number of RAMs completed and the process score for both Normal Base Defense Events and Intruder Events. Table 6 provides a summary of results comparing IMPACT to Baseline across objective performance measures.

Table 6: Evaluation Summary (IMPACT vs Baseline).

Measure	IMPACT > Baseline	
Objective Measures	**IMPACT > Baseline**	**High Complexity < Low Complexity**
Rams Completed Correctly	*	**
Normal Base Defense Outcome Measure		
Normal Base Defense Process Measure	*	**
Intruder Events Outcome Measure		**
Intruder Events Process Measures	**	**
Response to System Failures/Environmental Events		**
Commander Query Accuracy		**
Commander Query Response Time		
Time to Call Plays		
Number of Clicks to Call Plays	**	

*   significant at .05

** significant at .01

Despite failing to reach an alpha level of .05 for the Normal Base Defense and Intruder Outcome Measures, the overall pattern of the results indicates a similar trend as the process measures. In fact, the pattern of results indicated in Figure 36 for RAMs Completed Correctly was the same for Normal Base Defense Outcome and Process Measures, Intruder Events Outcome and Process Measures, and Commander Query Response Time. Additionally, the results indicated that participants performed better in the Low Complexity condition as compared to the High Complexity condition across almost all performance measures.

For Commander Query Response Time (shown in Figure 37) a significant interaction ($F(1,7) = 6.39, p = .04, \eta_p 2 = .48$) was found. In the Low Complexity condition participants were faster at answering queries with IMPACT as compare to Baseline. However, in the High Complexity condition participants answered commander queries faster with Baseline as compared to IMPACT.

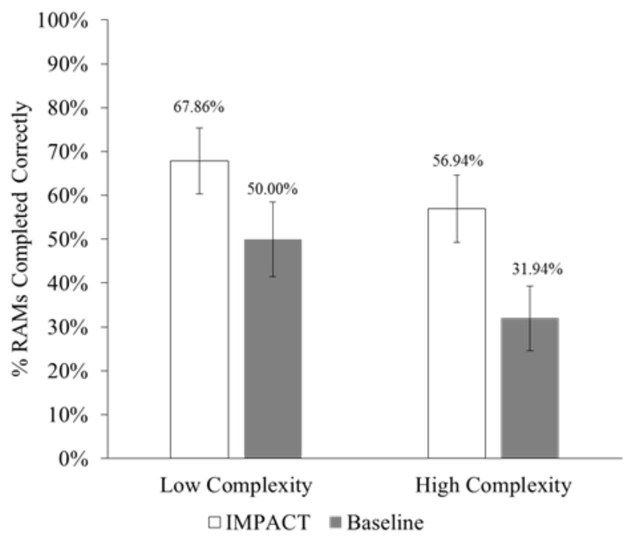


Figure 36: Percentage of RAMs Completed Correctly

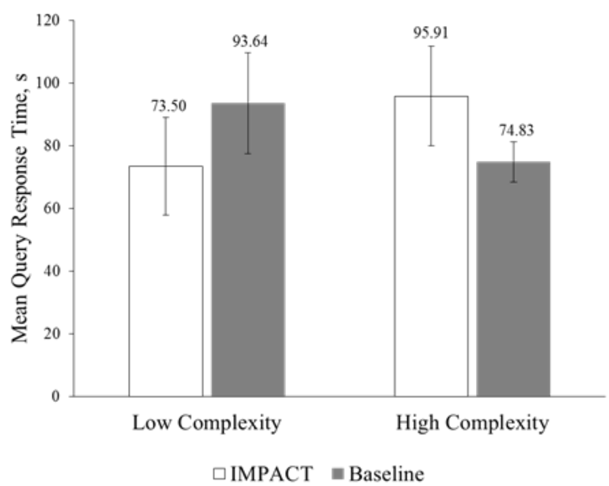


Figure 37: Commander Query Response Times

At first, the results of this analysis were perplexing—why were participants faster at answering queries with Baseline in the High Complexity condition? Upon examining the video recordings of the high complexity trials an interesting pattern of behavior emerged. In the high complexity scenarios, some participants using IMPACT would often set commander queries aside in order to focus on higher priority tasks (i.e., Normal Base Defense Events or Intruder Events) and return to the query later. In contrast, in the Baseline condition, these participants would immediately answer the query instead of responding to the higher priority tasking. It appeared as if some participants in the Baseline condition were relieved when a commander's query came in asking them, "What's TR-22's Speed?" in a high complexity scenario because it was an easy task that they knew how to answer. IMPACT, on the other hand, seemed to help participants prioritize tasks and enabled them to have discretionary control. The performance data supports this hypothesis. In Figure 38, each participant's average process score for Normal

Base Defense Events and Intruder Events is mapped on the y axis, while response time to commander queries is mapped on the x axis. Baseline data is coded blue and IMPACT data is coded red. Note, that of the Top 10 average process scores, 8 of them occur when the participant was using IMPACT. Also note that though three participants (1, 2, and 4) had noticeably slower mean query response times with IMPACT, all three had higher average process scores with IMPACT.

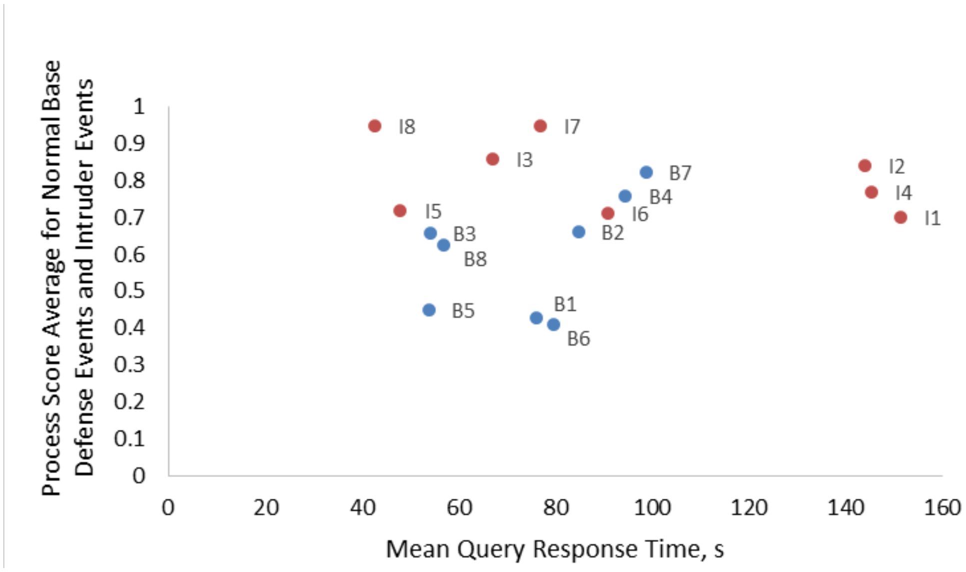


Figure 38: Commander Query Response Time vs. Process Score

5.2.4 Discussion

The hypothesis that participants would both prefer and perform better with IMPACT as compared to Baseline was supported. Participants preferred IMPACT as compared to Baseline on multiple subjective measures including usability, perceived value to future UxV operations, and ability to aid workload. Participants also performed better with IMPACT as compared to Baseline on multiple objective measures including number of RAMs completed and the process score for both Normal Base Defense Events and Intruder Events.

The hypothesis that Operator performance would be worse in the high complexity missions as compared to low complexity was supported, with participants performing better in the low complexity missions across almost all performance measures. However, the hypothesis that the performance difference between IMPACT and Baseline would be significantly greater in the High Complexity condition was not supported. Several factors may account for this including a lack of statistical power due to the small number of participants as well as limiting the experiment to two levels of complexity.

This research effort had multiple limitations. First and foremost, this study was limited to a small number of participants due to budgetary, time, and availability constraints. The small number of participants reduced the statistical power of the study. For example, the outcome measure difference between IMPACT and Baseline for both Normal Base Defense Events and Intruder Events was not significantly different despite a seemingly large difference in the means.

At the beginning of this research effort it was determined that the advantages of using participants with real world experience would outweigh the negatives. One of the negatives was the lack of time to train participants. In the operational world, a Warfighter would have far more time to learn the tool—two weeks instead of a single day—before needing to use the tool in a real-world mission. Unfortunately, it was extremely difficult to find active-duty participants who could donate two days of their time let alone two weeks. It is unreasonable to expect that participants could expertly wield all of IMPACT's functionality after a single day of training. In fact, certain results, such as participants not using the voice recognition system to call plays, may be directly tied to a lack of training. Valuable future research would include replicating this experiment with extensively trained participants, and it would be valuable even if these participants were not military professionals.

6 LOOKING BACK

Now that the IMPACT project has completed, it's appropriate to reflect on the approach taken and major decisions made. Many things worked and some things did not. One thing that very much worked was the time taken to meet face-to-face and plan the first year of the program. Although that was definitely the 'storming' phase of the effort, as we were from many different disciplines and had different areas of interest, we were able to work through these differences over a several day period to form the foundation of a workable architecture and plan for development. This process carried over to similar technology interchange meetings at the beginning of Year 2 and Year 3, all of which were valuable. An occasional mid-year meeting also occurred to keep the team in sync.

Communication truly is critical to working in large, interdisciplinary, distributed groups towards a unified product. And these communications were best facilitated when it occurred frequently, and when it occurred 'face-to-face'. This type of communication also fostered the building of deeper relationships that are foundational to many successful teams.

We had also set a goal for all sub teams to integrate their technology into the IMPACT functional testbed by the end of the project. This goal was only partially realized however. While most technology was indeed integrated with integration continuing to develop over the lifespan of the project, certain aspects (i.e., human models, machine learning) were not. This was due in large part to the comparative lack of maturity associated with those technologies; they were still at the foundational science level.

Additionally, it needs to be acknowledged that some scientists may simply be less inclined to tightly collaborate, likely for a myriad of reasons (past experience, dilution of focus, lower efficiency, etc.). It's best for the whole for that to be acknowledged early so that the integrated aspects of any project can continue forward with a coalition of only the willing.

The ARPI process, by and large, forced researchers out of their comfort zone and into interactions with scientists and technologies they were largely unfamiliar with but were critical to an overall systems solution to effective human-autonomy applications. And it forged relationships between researchers that will continue long after the ARPI process ends. Many IMPACT researchers commented that this project was the most rewarding project they had ever worked on, which is firm evidence that the process worked in this case.

Finally, the approach that ASD/R&E took towards managing the ARPI process should be lauded. They could have chosen a micro-management style that burdened the research teams with numerous reporting requirements. However the approach taken, that of quarterly reports, streamlined financial reporting, and detailed yearly reviews, enabled the ARPI teams to maximize time on technical matters.

7 RETURN ON INVESTMENT

The ARPI Program provided an excellent opportunity for Service lab researchers to collaborate on joint projects on autonomy. During the three years, these researchers gained significantly better understanding of each other's expertise and capabilities. Collectively, the IMPACT team has pushed the boundary of human-autonomy teaming science and technologies. With the knowledge gleaned from the IMPACT project and ARPI as a whole, the United States has been able to continue the trend of creating cutting edge technologies. Such technologies are particularly useful for military use, but may also provide a foundation for future research in civilian contexts as well. Below are some tangible examples on return on investment associated with the IMPACT project.

7.1 Delivering System-Level Innovations

The IMPACT project directly demonstrated human-autonomy teaming (HAT) innovations that were empirically proven to be superior to a state-of-the-art Baseline system via an extensive evaluation. Eight subject matter experts completed a variety of defense mission related tasks involving twelve simulated UxVs. Completion of these tasks was enhanced by our novel implementation of a play-calling approach, context-specific intelligent decision aiding, and advanced routing algorithms. Besides employing concise video-gaming type symbology throughout the interfaces, the approach was innovative in terms of providing a comprehensive suite of play-based interfaces that provided intuitive and efficient means by which the operator could team with C2 autonomy. Specifically, the interface supported capturing the operator's intent, allocating UxVs to tasks, routing UxVs, and editing on-going plays. The interfaces also provided visibility into the autonomy's reasoning, highlighted the tradeoffs of autonomous-generated plans and communicated ongoing play progress. With the intelligent aiding, cooperative routing, and well-integrated play workflow, participants' task performance was significantly improved on multiple mission performance metrics with the IMPACT system in comparison to the Baseline system. Participants were also able to execute plays using significantly fewer control inputs with IMPACT as compared to Baseline. Participants rated IMPACT higher than Baseline on all possible usability metrics. Participants also subjectively rated IMPACT significantly better than Baseline in terms of its perceived value to future UxV operations as well as its ability to aid workload.

This system innovation will continue at multiple research sites due to the creation of a 3-station DoD VDL for HAT-related research. Research is extending along several research fronts, all making maximal use of this important testbed.

7.2 New Software Products

This effort has produced several software products and numerous research results that are key for developing future human-autonomy systems. In terms of software, significant effort was put into developing Fusion. Fusion both implements a base set of capabilities that are necessary for building the core of any human-autonomy system, and it provides an extensible architecture that allows new autonomy capabilities to be rapidly incorporated. We believe Fusion therefore represents an invaluable framework for developing and testing new human-autonomy concepts and will pay great dividends when used by others in the DoD.

Similarly, other software components such as UxAS provide foundational autonomy capabilities that could serve as fundamental building blocks for autonomy research and future

DoD autonomy systems. For example, UxAS provides a core set of task routing and online execution capabilities needed by many physical, mobile systems, and like Fusion, it is designed to be extensible. UxAS is now publically available on GitHub and is currently the subject of AFRL's Summer of Innovation program. In this program, over 30 participants from NASA, Rockwell Collins, GE, CMU's Software Engineering Institute, and other industrial and academic partners are using UxAS as a case study for current and future V&V approaches. This will both spur future research and harden UxAS so that it can be confidently used in programs such the AFRL Loyal Wingman program, which aims to augment a manned fighter with unmanned teammates.

7.3 Research Results

In addition to new software that provides foundational capabilities for human-autonomy systems, new research results provide insights that are key for improving human-autonomy teaming. Over 60 publications, covering a range of topics from human autonomy interaction, artificial intelligence, computer science, human workload and attention modeling, etc. have been produced as a result of the IMPACT program (See Appendix A). These publications include conference papers and journal entries. In addition, an entire symposium was dedicated to IMPACT research at the 2017 International Symposium of Aviation Psychologists.

An innovative model of agent transparency (Situation Awareness of Agent Transparency-SAT) was developed and demonstrated to improve an operators' ability to reduce misuse and disuse of agents' planning and asset management decisions. By informing the human of the agent's intent, logic and predicted outcome uncertainty, the SAT model enabled a true synergy between the agent's ability to suggest solutions and the human's ability to adapt solutions to the current tactical situation.

A model was successfully developed of when a human supervisor may be performing poor visual scanning. Poor visual scanning leads directly to missed automation failures, which was operationally defined as critical for this effort. This meta-knowledge model is predictive, it runs in real-time, and has high accuracy.

Machine learning approaches were applied toward evolutionary learning of tactics with human inputs, and the automatic generation of new tasking. This foundational research can potentially result in major leaps in future IMPACT capabilities.

7.4 Reducing Cost

The IMPACT project demonstrated the potential to reduce future DoD costs by reducing the number of personnel required to manage multiple UxVs. This was directly illustrated for the application of future base defense by augmenting human management with IMPACT's agility-enhancing technologies. However, these technologies can easily be extended to other RSTA applications and even beyond.

7.5 Ensuring Trust

Trust was directly investigated through a series of experiments in the IMPACT project. We were able to show the ability of SAT information to calibrate trust (reduce misuse and disuse) and to improve subjective trust in the IMPACT system as well. The SAT model improved the operator's ability to correctly override the agent's misuse of multiple autonomous systems based on sub-optimal mission profiles while increasing trust in the agent's decisions when supported by the current tactical situation. In addition, HMI efforts focused on creating transparency throughout the IMPACT interface so that the operator could properly calibrate trust.

7.6 Disrupting Advanced Persistent Threats

Terrorist's threat against military and civilian installations are becoming ubiquitous. 24/7 use of autonomous systems informed by IMPACT related agility tools offers a cost effective and practical means of protecting large scale installations and urban terrain.

7.7 Collaborations/Extensions/Fostering New Opportunities

The technology development and collaborations initiated during the ARPI process are being leveraged for future efforts. The IMPACT Virtual HAT testbed has already proven to be a key enabler for jumpstarting new research and fostering new joint service collaboration. Below are a few examples of how the IMPACT project has extended into new research areas and advanced technology development, within the DoD, Industry, Academia, and internationally.
DoD:

- DARPA (including CODE and Explainable AI programs)
- Research on optimal human-autonomy teaming structures and communication requirements
- AFRL Autonomy Initiative on manned/unmanned teaming for air operations
- "Autonomy at Rest" framework for multi-domain C2 applications
- Dynamic operator workload prediction and augmentation strategies
- Multiple University Research Initiative to develop transparency related concepts with NRL, AFRL

Industry:

- Operator performance sensing, assessing, and augmenting to dynamically balance workload
- Augmenting team performance in distributed operations
- Assessing complex contextual attention and dynamic engagement
- Cognitive assessment model and enhanced workload models for HAT

Academia:

- Transparency interfaces to evaluate UAV swarms
- Causes and mitigations for decision biases
- Multiple University Research Initiative to develop transparency related concepts with NRL, AFRL

International:

- TTCP Autonomy Strategic Challenge: Multi-nation autonomy co-development, integration, and assessment in live-virtual exercises. IMPACT selected as the core C2 autonomy component and will combine its capabilities with those of other nations to produce a more robust and mature C2 capability.
- NATO-HFM-247: IMPACT used to inform HAT metrics and HAT design pattern development

Looking forward, there are many emerging opportunities for extending the technology and understanding developed within IMPACT. Many of these opportunities are documented within the “Next Steps” subsections within Sections 4.0 and 5.0. A few additional ideas follow.

- A systematic exploration of communication degradations and information uncertainty on HAT performance. Starting with IMPACT and the assumption of perfect knowledge and continuous communications, research is needed explore how communication links and mission information might be degraded, what the effects are on HAT performance, and what mitigation methods work the best to maintain overall system performance. The IMPACT testbed is a perfect baseline by which to study this problem area; data already exists on HAT performance in perfect/continuous communication environments, and mitigation exploration can entail countless combinations of machine reasoning, multiple cooperative mission planning methods, and advanced HMI solutions.
- A more in-depth implementation and study of the results of implementing HAT patterns identified in NATO HFM-247. For instance, IMPACT Agent might alter its behavior based on priority of plays. The Task Manager could adapt based on task due time. However a more robust change could occur as tasks become close to key decision deadlines, increasing the use of automation as the human operator is unable to address tasks in time. A wide variety of responses are possible and could be controlled via working agreements with the operator.
- A plug-and-play architecture for planners with hierarchical planning to increase the ability to handle diverse autonomous assets and handle multiple levels of detail. Hierarchical planning which is considered one of the best possibilities for handling increased complexity will add challenges to the operator that will need to be explored. Hierarchical planning will allow the combination of probabilistic and constraint-based planning at different levels. Including probabilistic planning will also require HAT studies relative to these planners.

8 LOOKING FORWARD AND CONCLUSION

The IMPACT project produced significant knowledge in a number of areas important to autonomy-related capabilities (see Appendix A for a listing of the many publications generated from this effort). Not only did the project spur advancements in component technology development, model development, and general design understanding/guidance, but much was learned from the integration of key autonomy-related technologies into a single multi-UxV control station application. IMPACT also produced a robust DoD “virtual lab” for continued human-autonomy teaming research. This was a key objective of the ARPI process. A three-station system (C2, sensor operator, & TOC) is available for organic wide-spectrum HAT evaluations with sites currently at AFRL, SPAWAR and ARL. Moreover, the IMPACT system breaks new ground in terms of enabling C2 of heterogeneous unmanned vehicles from the same control station. This is accomplished with the innovative autonomy teaming interfaces that enable the operator to seamlessly transition between many control states (from manual to fully autonomous), supporting the agility required for future Air Force missions.

This new vision for future human-autonomy systems was successfully conveyed to DoD senior leadership via many interactive demonstrations of the IMPACT system. This vision clearly illustrates that the human will continue to have a prominent role in interacting with increasingly autonomous technology, dynamically flexing between supervisor, teammate, or manual controller as conditions dictate. Finally IMPACT technologies have extended/transitioned in a myriad of ways. Other ARPI projects have leveraged IMPACT technology to advance their aims while new DoD projects (including JCTD and DARPA programs) and several industry contractors now utilize IMPACT generated capabilities in technology development efforts. Additionally, IMPACT has become the core C2 autonomy piece within the TTCP Autonomy Strategic Challenge which is a 3-year, 5-nation effort to integrate and assess promising allied interoperable autonomy capabilities in mixed live/virtual multi-UxV littoral environments.

The IMPACT project has enabled a deeper exploration into the critical issues that influence flexible and effective human-autonomy collaboration. Although the IMPACT evaluation demonstrated value in several aspects related to operator-autonomy teaming, several deficiencies and gaps in understanding were also identified and improvements are underway. These include research related to novel methods for enabling bi-directional communication and management of temporal constraints, more naturalistic dialogue and sketch interactions, and consideration of information uncertainty in decision-making tasks. Additionally, research investigating the effects of decentralized re-planning capability, real-time operator functional state assessment, and alternative team structures on overall human-autonomy teaming. The results of these follow-on efforts will provide a much richer understanding of this area.

9 REFERENCES

Air Force. (2015, June). Autonomous Horizons: System Autonomy in the Air force – A Path to the Future. *Volume 1: Human-Autonomy Teaming.* USAF Office of the Chief Scientist, AF/ST-TR-15-01.

Apker, T., Johnson, B., & Humphrey, L. (2016). LTL templates for play-calling supervisory control. AIAA Infotech@Aerospace (pp. 9-17). San Diego: AIAA.

Behymer, K.J., Mersch, E.M., Ruff, H.A., Calhoun, G.L., & Spriggs, S.E. (2014). Unmanned vehicle plan comparison visualizations for effective human-autonomy teaming. *Proceedings of the 6th International Conference on Applied Human Factors and Ergonomics (AHFE), Procedia Manufacturing*, 3, 1022-1029.

Behymer, K., Rothwell, C., Ruff H., Patzek, M., Calhoun, G., Draper, M., Douglass, S., Kingston, D., Lange, D. (2017). Initial evaluation of the intelligent multi-UxV planner with adaptive collaborative/control technologies (IMPACT). *Technical report AFRL-RH-WP-TR-2017-0011.* Wright-Patterson Air Force Base, OH: Air Force Research Laboratory.

Behymer, K.J., Ruff, H.A., Mersch, E.M., Calhoun, G.L., & Spriggs, S.E. (2015). Visualization methods for communicating unmanned vehicle plan status. *Proceedings of the International Symposium on Aviation Psychology*.

Brooke, J. (1996). SUS: a ‘quick and dirty’ usability scale. In P. Jordan, B. Thomas, I. McClelland, & B. Weerdmeester (Eds.), *Usability Evaluation in Industry.* Bristol, PA: Taylor & Francis Inc.

Calhoun, G., Draper, M.H., Ruff, H., Barry, T., Miller, C.A., & Hamell, J. (2012). Future unmanned aerial systems control: Feedback on a highly flexible operator-automation delegation interface concept. *AIAA Infotech @ Aerospace Conference*, AIAA-2012-2549, 1-16.

Calhoun, G.L., Goodrich, M.A., Dougherty, J.R., & Adams, J.A. (2016). Human-Autonomy Collaboration and Coordination Toward Multi-RPA Missions. In N. Cooke, L. Rowe, W.

Bennett, & D. Joralmon (Eds.), *Remotely Piloted Aircraft: A Human Systems Integration Perspective*, Chapter 5, 101-136. New York: Wiley.
Calhoun, G.L., Ruff, H.A., Behymer, K.J., & Frost, E.M. (2017). Human-autonomy teaming interface design considerations for multi-unmanned vehicle control. *Theoretical Issues in Ergonomics Science.* (Special Issue: Human-Autonomy Teaming).
Calhoun, G.L., Ruff, H.A., Behymer, K.J., & Mersch, E.M. (2017). Operator-Autonomy Teaming Interfaces to Support Multi-unmanned Vehicle Missions. In P. Savage-Knepshield & J. Chen (Eds.), *Advances in Human Factors in Robots and Unmanned Systems: Advances in Intelligent Systems and Computing 499* (pp. 113-126). Springer.
Calhoun, G.L., Ruff, H.A., Behymer, K.J., & Rothwell, C.D. (2017). Evaluation of interface modality for control of multiple unmanned vehicles. *Human Computer Interaction International (HCII*).
Chen, J.Y.C., and Barnes, M.J. (2014). Human–agent teaming for multirobot control: A review of human factors issues. *IEEE Trans. Human-Machine Systems, 44*, 13-29.
Chen, J.Y.C., & Barnes, M.J. (2012a). Supervisory control of multiple robots: Effects of imperfect automation and individual differences. *Human Factors, 54,* 157–174.
Chen, J.Y.C., & Barnes, M.J. (2012b). Supervisory Control of Multiple Robots in Dynamic Tasking Environments. Ergonomics 55, 1043-1058.
Chen, J.Y.C., Barnes, M.J., & Harper-Sciarini, M. (2011). Supervisory control of multiple robots: Human performance issues and user interface design. *IEEE Transactions on Systems, Man, and Cybernetics--Part C: Applications and Reviews, 41*, 435-454.
Douglass, S.A., & Mittal, S. (2013). A Framework for Modeling and Simulation of the Artificial. Andreas Tolk (Ed.), Ontology, Epistemology, and Teleology of Modeling and Simulation, Intelligent Systems Series, Springer-Verlag
Draper, M.H. (2007). Advanced UMV operator interface. In *Uninhabited Military Vehicles (UMVs): Human Factors Issues in Augmenting the Force*, NATO Report RTO-TR-HFM-078, Chapter 6. NATO RTO: Neuilly-sur-Siene, CEDEX.
Draper, M., Calhoun, G., Hansen, M. Douglass, Spriggs, S., Patzek, M., Rowe, A., Evans, D., Ruff, H., Behymer, K., Howard, M., Bearden, G., & Frost, E. (2017). Intelligent multi-unmanned vehicle planner with adaptive collaborative/control technologies (IMPACT). *Proceedings of the International Symposium on Aviation Psychology.*
Draper, M.H., Miller, C.A., Calhoun, G.L., Ruff, H., Hamell, J., Benton, J., & Barry, T. (2013). Multi-unmanned aerial vehicle systems control via flexible levels of interaction: An adaptable operator-automation interface concept demonstration. *AIAA Infotech @ Aerospace Conference,* AIAA-2013-4803.
Eggers, J.W., & Draper, M.H. (2006). Multi-UAV Control for Tactical Reconnaissance and Close Air Support Missions: Operator Perspectives and Design Challenges, *Proceedings of the NATO RTO Human Factors and Medicine Panel Symposium*, HFM-135, held in Biarritz, FR, 9-11 Oct. NATO RTO: Neuilly-sur-Siene, CEDEX.
Endsley, M. R. (1995). Toward a theory of situation awareness in dynamic systems. *Human Factors, 37,* 32–64.
Feitshans, G. L. & Davis, J. E. (2011) Advanced Net-Centric Combat Unmanned Air Vehicle *(Technical Report AFRL-RG-WP-TR-2011-0126)*, Air Force Research Lab.
Feng, L., Wiltsche, C., Humphrey, L., & Topcu, U. (2015). Controller synthesis for autonomous systems interacting with human operators. ACM/IEEE 6th International Conference on Cyber-Physical Systems (pp. 70-79). Seattle: IEEE.

Feng, L., Wiltsche, C., Humphrey, L., & Topcu, U. (2016). Synthesis of Human-in-the-Loop Control Protocols for Autonomous Systems. IEEE Transactions on Automation Science and Engineering , 13 (2), 450-462.

Freedy, A., DeVisser, E., Weltman, G., & Coeyman, N. (2007). Measurement of Trust in Human-Robot Collaboration. *International Symposium on Collaborative Technologies and Systems, CTS May 2007.* doi: 10.1109/CTS.2007.4621745.

Gao, J., Lee, J.D., & Zhang, Y. (2006). A dynamic model of interaction between reliance on automation and cooperation in multi-operator multi-automation situations. *International Journal of Industrial Ergonomics, 36*(5), 511-526.

Gauci, J., & Stanley, K.O. (2007). Generating Large-scale Neural Networks through Discovering Geometric Regularities. *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-2007), New York, NY; 997-1004*. doi: 10.1145/1276958.1277158.

Groom, V., & Nass, C.I. (2007). Can Robots be Teammates? Benchmarks in Human-Robot Teams. *Psychological Benchmarks of Human-Robot Interaction, Interaction Studies, October 2007, 8(3); 483-500*. doi: 10.1075/is.8.3.10gro.

Goodrich, M.A., & Olsen, D.R. (2003). Seven principles of efficient human robot interaction. *IEEE International Conference of Systems, Man and Cybernetics*, *4*, 3942-3948.

Hansen, M., Calhoun, G., Douglass, S., & Evans, D. (2016). Courses of action display for multi-unmanned vehicle control: A Multi-disciplinary approach. *Artificial Intelligence and Human-Robot Interaction (AAAI) Fall Symposium Series*.

Hooper, D.J., Duffy, J.P., Calhoun, G.L., & Hughes, T.L. (2015). A taxonomy for improving dialog between autonomous agent developers and human machine interface designers. *Artificial Intelligence and Human-Robot Interaction, AAAI Fall Symposium Series.*

Humphrey, L., Wolff, E., & Topcu, U. (2014). Formal Specification and Synthesis of Mission Plans for Unmanned Aerial Vehicles. AAAI Spring Symposium. Palo Alto: AAAI.

Jian, J., Bisantz, A., Drury, C., & Llinas, J. (2000). Foundations for an Empirically Determined Scale of Trust in Automated Systems. Air Force Research Laboratory Technical Report, HE-WP-TR-2000-0101, 2000.

Kilgore, R., & Voshell, M. (2014). Increasing the transparency of unmanned systems: Applications of ecological interface design. International Conference on Virtual, Augmented and Mixed Reality, 378-389.

Kim, B.U., & Humphrey, L. (2015). Satisfiability Checking of LTL Specifications for Verifiable UAV Mission Planning. AIAA SciTech Conference. National Harbor: AIAA.

Kingston, D., Rasmussen, S., & Humphrey, L. (2016). Automated UAV Tasks for Search and Surveillance. IEEE Multi-Conference on Systems and Control.

Kingston, D.B., Rasmussen, S.J., & Mears, M.J. (2009). Base defense using a task assignment framework. AIAA Guidance, Navigation, and Control Conference, AIAA-2009-6209

Klein, G., Woods, D.D., Bradshaw, J.M., Hoffman, R.R., & Feltovich, P.J. (2004). Ten challenges for making automation a "team player" in joint human-agent activity. IEEE Intelligent Systems, *19*(6), 91-95.

Lee, J.D. (2012). Trust, trustworthiness, and trustability. Presentation at the Workshop on Human Machine Trust for Robust Autonomous Systems. Jan 31-Feb 2, 2012, Ocala, FL.

Lee, J.D., & See, K.A., (2004). Trust in Automation: Designing for Appropriate Reliance. *Human Factors, Spring 2004, (46)1;* 50-80. doi: 10.1518/hfes.46.1.50.30392.

Lee, J.R., & Nass, C.I. (2010). Trust in Computers: The Computers-are-Social-Actors (CASA) Paradigm and Trustworthiness Perception in Human-Computer Communication. In *Trust* *and Technology in a Ubiquitous Modern Environment: Theoretical and Methodological Perspectives, Latusek D., & Gerbasi, A., Eds. Information Science Reference, 2010; 1-15*. doi: 10.4018/978-1-61520-901-9.ch001.

Linegang, M.P., Stoner, H.A., Patterson, M.J., Seppelt, B.D., Hoffman, J.D., Crittendon, Z.B., & Lee, J.D. (2006). Human-automation collaboration in dynamic mission planning: A challenge requiring an ecological approach. *Proceedings of the Human Factors & Ergonomics Society Annual Meeting*, *50*(23), 2482-2486.

Martinage, R. (2014). Toward a New Offset Strategy: Exploiting U.S. Long-Term Advantages to Restore U.S. Global Power Projection Capability. Center for Strategic and Budgetary Assessments.

Martin, Robert C., Martin, Micah (2012). Agile Principles, Patterns, and Practices in C#.

Mercado, J.E., Rupp, M.A., Chen, J. Y., Barber, D., Procci, K., & Barnes, M. (2015). *Effects of agent transparency on multi-robot management effectiveness* (No. ARL-TR-7466). Aberdeen Proving Ground, MD: ARL HRED.

Mercado, J. Rupp, M., Chen, J., Barnes, M., Barber, D., & Procci, K. (May 2016). Intelligent agent transparency in human-agent teaming for multi-UxV management. *Human Factors, 58*, pp. 401–415.

Mersch, E.M., Behymer, K.J., Calhoun, G.L., Ruff, H.A., & Dewey, J.S. (2016). Game-based delegation interface design for unmanned vehicles color coding and icon row assignment. *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, *60*(1): 122-126.

Miller, C.A., Hamell, J., Barry, T., Ruff, H., Draper, M.H., & Calhoun, G.L. (2012). Adaptable operator-automation interface for future unmanned aerial systems control: Development of a highly flexible delegation concept demonstration. *AIAA Infotech @ Aerospace Conference, AIAA-2012-2529,* 1-21.

Miller, C.A., & Parasuraman, R. (2007). Designing for flexible interaction between humans and automation: delegation interfaces for supervisory control. *Human Factors*, *49*(1), 57-75.

Nakamura, C., & Zeng-Treitler, Q. (2012). A taxonomy of representation strategies in iconic communication. *International Journal of Human-Computer Studies, 70*(8): 535-551.

Nass, C.I., Fogg, B.J., & Moon, Y. (1996). Can Computers be Teammates? *International Journal of Human-Computer Studies, December 1996, 45(6); 669-678*. doi: 10.1006/ijhc.1996.0073.

Opperman, R. (1994). Adaptive User Support, Hillsdale, NJ: Erlbaum.

Ososky, S., Schuster, D., Jentsch, F., Fiore, S., Shumaker, R., Lebiere, C., Kurup, U., Oh, J., & Stentz, A. (2012). The importance of shared mental models and shared situation awareness for transforming robots from tools to teammates. *SPIE* 8387, Unmanned Systems Technology XIV, 838710.

Oviatt, S. (1999). Ten myths of multimodal interaction. *Communication ACM, 42*(11), 74-81.

Petrosyuk, I. (2016). Neuro-Fuzzy Model for Image Processing in Electro-optical Application, *Modern Problems of Radio Engineering, Telecommunications, and Computer Science, 2016, TCSET 206 International Conference,* doi: 10.1109/TCSET.2006.4404501

Rao, A. S., and Georgeff, M. P. (1995). BDI Agents: From Theory to Practice. *Proceedings of the First International Conference on Multiagent Systems.* Palo Alto, CA: Association for the Advancement of Artificial Intelligence Press (pp. 312–319).

Rasmussen, J. (1990). Mental models and the control of action in complex environments. In D. Ackermann, D., & M.J. Tauber (Eds.), *Mental Models and Human-Computer Interaction*, pp. 41-46, North-Holland: Elsevier Science Publishers.

Rowe, A., Spriggs, S., Hooper, D. (2015). Fusion: A Framework for Human Interaction with Flexible-Adaptive Automation across Multiple Unmanned Systems. *Proceedings of the International Symposium on Aviation Psychology*.

Sanders, T., Oleson, K.E., Billings, D.R., Chen, J.Y.C., & Hancock, P.A. (2011). A Model of Human-Robot Trust: Theoretical Model Development. *Proceedings of the Human Factors and Ergonomics Society Annual Meetings, September 2011, 55(1); 1432-1436*. doi: 10.1177/1071181311551298.

Sanders, T., Wixon, T., Schafer, J., Chen, Y. C., & Hancock, P. (2014). The Influence of Modality and Transparency on Trust in Human-Robot Interaction. *2014 IEEE International Inter-Disciplinary Conference on Cognitive Methods in Situation Awareness and Decision Support (CogSIMA, 2014), March 2014; 156-159*. doi: 10.1109/CogSIMA.2014.6816556.

Sheridan, T.B., & Parasuraman, R. (2005). Human-Automation Interaction. *Reviews of Human Factors and Ergonomics, November 2005, 1(1); 89-129.* doi: 10.1518/155723405783703082.

Shneiderman, B. (1992). *Designing the User Interface: Strategies for Effective Human-computer Interactions*, 2nd ed. Reading, MA: Addison-Wesley.

Stanley, K.O., D'Ambrosio, D.B., & Gauci, J. (2009). A Hypercube-based Encoding for Evolving Large-scale Neural Networks. *Artificial Life, April 2009, 15(2); 185-212*. doi: 10.1162/artl.2009.15.2.15202.

Stanley, K.O., & Miikkulainen, R. (2002). Evolving Neural Networks through Augmenting Topologies. *Journal of Evolutionary Computation, Summer 2002, 10(2); 99-127*. doi: 10.1162/106365602320169811.

Stowers, K., Kasdaglis, N., Rupp, M., Chen, J., Barber, D., & Barnes, M. (2017). Insights into Human-Agent Teaming: Intelligent Agent Transparency and Uncertainty. In P. Savage-Knepshield & J. Chen (Eds.), *Advances in Human Factors in Robots and Unmanned Systems*, 149-160.

Stowers, K., Kasdaglis, N., Rupp, M., Newton, O., Wohleber, R., & Chen, J. (2016). Intelligent agent transparency: The design and evaluation of an interface to facilitate human and artificial agent collaboration. *Proc. Human Factors and Ergonomics Society 60th International Annual Meeting*, Washington, DC (19-23 Sept. 2016).

Vicente, K.J., & Rasmussen, J. (1992). Ecological interface design: Theoretical foundations. IEEE Transactions on Systems, Man, and Cybernetics, SMC-22, 589-606.

Wickens, C.D., Hollands, J., Banbury, S., & Parasuraman, R. (2013). Engineering Psychology and Human Performance, 4th Ed., Upper Saddle River, NJ: *Pearson, 2013*.

Woods, D.D. (1984). A concept to improve the cognitive coupling of person and computer. *I*nternational Journal of Man-Machine Studies, *21*(2), 229-244.

Woods, D.D., & Hollnagel, E. (2006). Joint Cognitive Systems: Patterns in Cognitive Systems Engineering. Boca Raton, FL: CRC Press, Taylor & Francis.

10 IMPACT RELATED PUBLICATIONS

Aguiar, N., Zish, K., & Trafton, J. (2016). Interruptions Reduce Performance across All Levels of Signal Detection When Estimations of Confidence are Highest. *Human Factors and Ergonomics Conference 2016.*

Altmann, E.M., & Trafton, J.G. (2015). Brief Lags in Interrupted Sequential Performance: Evaluating a Model and Model Evaluation Method. *International Journal of Human-Computer Studies*, 79, 51-65.

Apker, T., Johnson, B., & Humphrey, L. (2016). LTL templates for play-calling supervisory control. *AIAA Infotech@Aerospace* (pp. 9-17). San Diego: AIAA.

Atahary, T., Taha, T., & Douglass, S.A. (2016). The Parallelized Mining of Domain Knowledge on GPGPU and Xeon Phi Clusters. *Journal of Supercomputing*, *72*(6), pp. 2132-2156.

Atahary, T., Taha, T., Douglass, S.A., & Webber, F. (2015). Knowledge Mining for Cognitive Agents Through Path based Forward Checking. *16th IEEE/ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing* (SNPD 2015), Takamatsu, Japan.

Barnes, M. J., Chen, J. Y., Wright, J. L., & Stowers, K. (2016). Human-Agent Teaming for Effective Multirobot Management: Effects of Agent Transparency. *In International Conference on Engineering Psychology and Cognitive Ergonomics* (pp. 169-178). Springer International Publishing.

Behymer, K.J., Mersch, E.M., Ruff, H.A., Calhoun, G.L., & Spriggs, S.E. (2014). Unmanned vehicle plan comparison visualizations for effective human-autonomy teaming. *Proceedings of the 6th International Conference on Applied Human Factors and Ergonomics (AHFE)*, *Procedia Manufacturing*, 3, 1022-1029.

Behymer, K., Rothwell, C., Ruff H., Patzek, M., Calhoun, G., Draper, M., Douglass, S., Kingston, D., Lange, D. (2017). Initial evaluation of the intelligent multi-UxV planner with adaptive collaborative/control technologies (IMPACT). *Technical report AFRL-RH-WP-TR-2017-0011.* Wright-Patterson Air Force Base, OH: Air Force Research Laboratory.

Behymer, K.J., Ruff, H.A., Mersch, E.M., Calhoun, G.L., & Spriggs, S.E. (2015). Visualization methods for communicating unmanned vehicle plan status. *Proceedings of the International Symposium on Aviation Psychology*.

Breslow, L. A., Gartenberg, D, McCurry, J. M., & Trafton, J. G. (2014). Dynamic Operator Overload: A Model for Predicting Workload During Supervisory Control. *IEEE Transactions on Human-Machine Systems*, 44(1).

Calhoun, G.L., Goodrich, M.A., Dougherty, J.R., & Adams, J.A. (2016). Human-Autonomy Collaboration and Coordination Toward Multi-RPA Missions. In N. Cooke, L. Rowe, W. Bennett, & D. Joralmon (Eds.), *Remotely Piloted Aircraft: A Human Systems Integration Perspective*, Chapter 5, 101-136. New York: Wiley.

Calhoun, G.L., Ruff, H.A., Behymer, K.J., & Frost, E.M. (2017). Human-autonomy teaming interface design considerations for multi-unmanned vehicle control. *Theoretical Issues in Ergonomics Science.* (Special Issue: Human-Autonomy Teaming).

Calhoun, G.L., Ruff, H.A., Behymer, K.J., & Mersch, E.M. (2017). Operator-Autonomy Teaming Interfaces to Support Multi-unmanned Vehicle Missions. In P. Savage-Knepshield & J. Chen (Eds.), *Advances in Human Factors in Robots and Unmanned Systems: Advances in Intelligent Systems and Computing 499* (pp. 113-126). Springer.

Calhoun, G.L., Ruff, H.A., Behymer, K.J., & Rothwell, C.D. (2017). Evaluation of interface modality for control of multiple unmanned vehicles. *Human Computer Interaction International (HCII*).

Chen, J.Y.C., & Barnes, M.J. (2014). Human–agent teaming for multirobot control: A review of human factors issues. *IEEE Trans. Human-Machine Systems, 44,* 13-29.

Chen, J.Y.C., & Barnes, M.J. (2015). Agent Transparency for Human-Agent Teaming Effectiveness. *IEEE Conference on Systems, Man, and Cybernetics.* Hong Kong.

Chen, J.Y.C., Barnes, M.J., Selkowitz, A., & Stowers, K. (2016). Effects of agent transparency on human-autonomy teaming effectiveness. *Proc. IEEE Conference Systems, Man, & Cybern 2016*, Budapest, Hungary (9-12 Oct, 2016).

Chen, J.Y.C., Barnes, M., Selkowitz, A., Stowers, K., Lakhmani, S., & Kasdaglis, N. (2016). Human-Autonomy Teaming and Agent Transparency. *Proc. ACM Intelligent User Interface (IUI) Conference* (pp. 28-31), Sonoma, CA (7-10 March, 2016).

Chen, J., Croft, B., Lange, D., Lucero, C., Moreau, L., & Reeder, J., (2017). Transparency and Provenance for Man-Machine Autonomous Systems in Command and Control. *22nd International Command and Control Research & Technology Symposium (ICCRTS 2017). Unpublished script.*

Chen, J., Lakhmani, S., Stowers, K., Selkowitz, A., Wright, J., & Barnes, M. (2017). Situation awareness-based agent transparency and human-autonomy teaming effectiveness. *Theoretical Issues in Ergonomic Science (Special Issue on Human-Autonomy Teaming).*

Chen, J., Procci, K., Boyce, M., Wright, J., Garcia, A., & Barnes, M. (2014). Situation Awareness-based Agent Transparency. Technical Report ARL-TR-6905, Aberdeen Proving Ground, MD: Army Research Laboratory.

Coronado, B., Gustafson, E., Reeder, J., & Lange, D. S., (2016). Mixing Formal Methods, Machine Learning, and Human Interaction Through an Autonomics Framework. *The 2016 AAAI Fall Symposium Series: Cross-Disciplinary Challenges for Autonomous Systems.*

Draper, M., Calhoun, G., Hansen, M. Douglass, Spriggs, S., Patzek, M., Rowe, A., Evans, D., Ruff, H., Behymer, K., Howard, M., Bearden, G., & Frost, E. (2017). Intelligent multi-unmanned vehicle planner with adaptive collaborative/control technologies (IMPACT). *Proceedings of the International Symposium on Aviation Psychology.*

Dzieciuch, D., Reeder, J., Gutzwiller, R. S., Gustafson, E., Coronado, B., Martinez, L., Croft, B., Lange, D. S., (2017). Amplifying Human Ability through Autonomics and Machine Learning in IMPACT. *Proceedings Society of Photo-Optical Instrumentation Engineers (SPIE) 10194, Micro- and Nanotechnology Sensors, Systems, and Applications IX, 101941Y (May 18, 2017).* doi: 10.1117/12.2262849

Edmonds, M., Atahary, T., Taha, T., & Douglass, S.A. (2015). High Performance Declarative Memory Systems through MapReduce. *16th IEEE/ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing* (SNPD 2015), Takamatsu, Japan

Feng, L., Humphrey, L., Lee, I., & Topcu, U. (2016). Human-interpretable diagnostic information for robotic planning systems. *IEEE/RSJ International Conference on Intelligent Robots and Systems* (pp. 1673-1680). Daejeon: IEEE.

Feng, L., Wiltsche, C., Humphrey, L., & Topcu, U. (2015). Controller synthesis for autonomous systems interacting with human operators. *ACM/IEEE 6th International Conference on Cyber-Physical Systems* (pp. 70-79). Seattle: IEEE.

Feng, L., Wiltsche, C., Humphrey, L., & Topcu, U. (2016). Synthesis of Human-in-the-Loop Control Protocols for Autonomous Systems. *IEEE Transactions on Automation Science and Engineering, 13* (2), 450-462.

Gartenberg, D., Breslow, L. A., & Trafton, J. G. (2014). Situation Awareness Recovery. *Human Factors, 56(4)*, 710-727.

Gartenberg, D., Gunzelmann, G., & Trafton, J. G. (under review). The microlapse theory of fatigue explanation of memory effects in sustained attention tasks. *Acta Psychologica*.

Gutzwiller, R.S., Espinosa, S. H., Kenny, C., & Lange, D.S. (2017). A Design Pattern for Working Agreements in Human-Autonomy Teaming. *International Conference of Applied Human Factors and Ergonomics (AHFE), January 2017*.

Gutzwiller, R.S., & Lange, D.S. (2016). Tasking Teams: Supervisory Control and Task Management of Autonomous Unmanned Systems. *International Conference on Virtual, Augmented and Mixed Reality, (VAMR 2016); 397–405.* doi: 10.1007/978-3-319-39907-2_38.

Gutzwiller, R.S., Lange, D.S., Reeder, J., Morris, R.L., & Rodas, O. (2015). Human-Computer Collaboration in Adaptive Supervisory Control and Function Allocation of Autonomous System Teams. *Human Computer Interaction International, August 2015.*

Gutzwiller, R.S., & Reeder, J. (2017). Human Interactive Machine Learning for Trust in Teams of Autonomous Robots. *2017 IEEE Conference on Cognitive and Computational Aspects of Situation Management (CogSIMA, 2017).* doi: 10.1109/CogSIMA.2017.7929607.

Hansen, M., Calhoun, G., Douglass, S., & Evans, D. (2016). Courses of action display for multi-unmanned vehicle control: A Multi-disciplinary approach. *Artificial Intelligence and Human-Robot Interaction (AAAI) Fall Symposium Series*.

Hiatt, L. M., Narber, C., Bekele, E., Khemlani, S. S., & Trafton, J. G. (under review). Human Modeling for human-robot collaboration. *International Journal of Research Robotics.*

Hiatt, L. M., & Trafton, J. G. (2016). Familiarity, priming, and perceptual in similarity judgments. *Cognitive Science, 41*(6), 1450-1484.

Hooper, D.J., Duffy, J.P., Calhoun, G.L., & Hughes, T.L. (2015). A taxonomy for improving dialog between autonomous agent developers and human machine interface designers. *Artificial Intelligence and Human-Robot Interaction, AAAI Fall Symposium Series.*

Humphrey, L., Wolff, E., & Topcu, U. (2014). Formal Specification and Synthesis of Mission Plans for Unmanned Aerial Vehicles. *AAAI Spring Symposium*.

Kim, B.U., & Humphrey, L. (2015). Satisfiability Checking of LTL Specifications for Verifiable UAV Mission Planning. *AIAA SciTech Conference.* National Harbor: AIAA.

Kingston, D., Rasmussen, S., & Humphrey, L. (2016). Automated UAV Tasks for Search and Surveillance. *IEEE Multi-Conference on Systems and Control.*

Kirschenbaum, S.S., Trafton, J.G., Schunn, C D., & Trickett, S.B. (2014). Visualizing Uncertainty: The impact on performance. *Human Factors, (56(3)* 509-520.

Lange, D.S., & Gutzwiller, R.S. (2016). Human-Autonomy Teaming Patterns in the Command and Control of Teams of Autonomous Systems. *International Conference on Engineering Psychology and Cognitive Ergonomics; 179-188, Springer International Publishing.*

Lange, D.S., Gutzwiller, R.S., Verbancsics, P., & Sin, T. (2014). Task Models for Human-Computer Collaboration in Supervisory Control of Teams of Autonomous Systems. *2014 IEEE International Inter-Disciplinary Conference on Cognitive Methods in Situation Awareness and Decision Support (CogSIMA, 2014); 97–102*. DOI: 10.1109/CogSIMA.2014.6816547.

Lange, D.S., Verbancsics, P., Gutzwiller, R., & Reeder, J. (2012). Command and Control of Teams of Autonomous Units. *17th International Command and Control Research and Technology Symposium (ICCRTS 2012).* Accession Number: ADA570196.

Lange, D.S., Verbancsics, P., Gutzwiller, R., & Reeder, J. (2013). Trust in Sparse Supervisory Control. *The 2013 AAAI Spring Symposium Series: Trust and Autonomous Systems.*

Marsh, L., Dzieciuch, I., & Lange, D.S., (2017). Machine Learning Approach for Task Generation in Uncertain Environments. *The AAAI 2017 Spring Symposium on Computational Context: Why It's Important, What It Means, and Can It Be Computed? Technical Report SS-17-03.*

Mercado, J.E., Rupp, M.A., Chen, J.Y.C., Barber, D., Procci, K., & Barnes, M.J. (2015). Effects of Agent Transparency on Multi-Robot Management Effectiveness. Technical Report ARL-TR-7466. Aberdeen Proving Ground, MD: Army Research Laboratory.

Mercado, J. Rupp, M., Chen, J., Barnes, M., Barber, D., & Procci, K. (May 2016). Intelligent Agent Transparency in Human-Agent Teaming for Multi-UxV Management. *Human Factors, 58,* 401-415.

Mersch, E.M., Behymer, K.J., Calhoun, G.L., Ruff, H.A., & Dewey, J.S. (2016). Game-based delegation interface design for unmanned vehicles color coding and icon row assignment. *Proceedings of the Human Factors and Ergonomics Society Annual Meeting, 60*(1): 122-126.

Rasmussen, S., Kalyanam, K., & Kingston, D. (2016). Field Experiment of a Fully Autonomous Multiple UAV/UGS Intruder Detection and Monitoring System. *IEEE International Conference on Unmanned Aircraft Systems.*

Rasmussen, S., & Kingston, D. (2015). Development and Flight Test of an Area Monitoring System using Unmanned Aerial Vehicles and Unattended Ground Sensors. *International Conference on Unmanned Aircraft Systems.*

Reeder, J., (2015). Team Search Tactics through Multi-Agent HyperNEAT. *International Conference on Information Processing in Cells and Tissues. Springer International Publishing, 2015.*

Rowe, A., Spriggs, S., Hooper, D. (2015). Fusion: A Framework for Human Interaction with Flexible-Adaptive Automation across Multiple Unmanned Systems. *Proceedings of the International Symposium on Aviation Psychology.*

Schulte, A., Donath, D., & Lange, D.S. (2016). Design Patterns for Human-Cognitive Agent Teaming. *18th International Conference on Human-Computer Interaction (HCII 2016);* 231-243. doi: 10.1007/978-3-319-40030-3_24.

Stowers K., Chen J.Y.C., Kasdaglis N., Newton O., Rupp, M., & Barnes M. (in preparation). Effects of Situation Awareness-based Agent Transparency Information on Human Agent Teaming for Multi-UxV Management. Tech Report ARL-TR-TBD. Aberdeen Proving Ground, MD: Army Research Laboratory.

Stowers, K., Kasdaglis, N., Rupp, M., Chen, J., Barber, D., & Barnes, M. (2016). Insights into Human-Agent Teaming: Intelligent Agent Transparency and Uncertainty. *Applied Human Factors and Ergonomics (AHFE) Conference*.

Stowers, K., Kasdaglis, N., Rupp, M., Newton, O., Wohleber, R., & Chen, J. (2016). Intelligent Agent Transparency: The Design and Evaluation of an Interface to Facilitate Human and Artificial Agent Collaboration. *Human Factors and Ergonomics Society Annual Meeting,* pp. 1706-1710.

Tamborello, F. & Trafton, J. G. (2017). Human error as an emergent property of action selection and task-place-holding. *Human Factors*, *59*(3), 377-392.

Verbanscisics, P., & Lange, D. S. (2013). Using Autonomics to Exercise Command and Control of Networks in Degraded Environments. *18th International Command and Control Research & Technology Symposium, (ICCRTS 2013). Accession Number: ADA587015.*

Zhang, M., & Kingston, D. (2015). Time-Space Network Based Exact Models for Periodical Monitoring Routing Problem. *IEEE American Control Conference.*

Zish, K., Hassanzadeh, S., McCurry, J.M., & Trafton, J.G. (2015). Interruptions can Change the Perceived Relationship between Accuracy and Confidence. *Human Factors and Ergonomics Society Annual Meeting*, pp. 239-234.

Zuniga, J., McCurry, M., & Trafton, J. G. (2014). A Process Model of Trust In Automation: A Signal Detection Theory Based Approach. *Human Factors and Ergonomics Society Annual Meeting*, pp. 827-831.

LIST OF ACRONYMS

AFRL	Air Force Research Lab
AMASE	Aerospace Multi-Agent Simulation Environment
ANOVA	Analysis of Variance
API	Application Programming Interface
ARL	Army Research Lab
ARPI	Autonomy Research Pilot Initiative
ASD/R&E	Assistant Secretary of Defense for Research and Engineering
AVTAS	Aerospace Vehicle Technology Assessment and Simulation
BM	Behavior Model
C2	Command and Control
CDO	Cognitive Domain Ontology
CECEP	Cognitively Enhanced Complex Event Processing
CMASI	Common Mission Automation Services Interface
CMU	Carnegie Mellon University
COA	Course of Action
DARPA	Defense Advanced Research Projects Agency
DIS	Distributed Interactive Simulation
DoD	Department of Defense
DREN	Defense Research & Engineering Network
EO	Electro-Optical
ETE	Estimated Time Enroute
FSM	Finite State Machine
GIS	Geospatial Information Systems
HAI	Human-Agent Interaction
HAT	Human-Autonomy Teaming
HD	High Definition
HM	Highly Mobile
HMI	Human Machine Interface
HUD	Head Up Display
IA	Intelligent Agent
ICET	Intelligent Control and Evaluation of Teams
IML	Interactive Machine Learning
IMPACT	Intelligent Multi-UxV Planner with Adaptive Collaborative/Control Technologies
IR	Infrared
JCTD	Joint Capability Technology Demonstration
JSON	JavaScript Object Notation
LCD	Liquid Crystal Display
LED	Light Emitting Diode
LMCP	Light-Weight Message Control Protocol
LTL	Linear Temporal Logic
MVVM	Model-View-View-Model
NAIs	Named Areas of Interest
NFCP	Normal Full Coverage Patrol
NIIRS	National Imagery Interpretability Rating Scale
NRL	Navy Research Lab

OneSAF	One Semi-Automated Forces
PNG	Portable Networks Graphic
RAM	Random Anti-Terror Measure
RML	Research Modeling Language
ROI	Region of Interest
ROZ	Restricted Operating Zone
RPA	Remotely Piloted Aircraft
RSTA	Reconnaissance, Surveillance, and Target Acquisition
SA	Situation Awareness
SAT	SA-Based Agent Transparency
SLUGS	Small but Complete Grone Synthesizer
SO	Sensor Operator
SOLID	Single Responsibility, Open/Closed, Liskov Substitution, Interface Segregation, and Dependency Inversion
SPAWAR	Space and Naval Warfare Systems Command
SUS	System Usability Scale
TCP	Transmission Control Protocol
TOC	Test Operator Console
TPR	Task Priority Register
UAV	Unmanned Air Vehicle
UDP	User Datagram Protocol
UGV	Unmanned Ground Vehicle
USV	Unmanned Surface Vehicle
UxAS	Unmanned Systems Autonomy Services
UxV	Unmanned Vehicles (of any type)
V&V	Verification & Validation
VDL	Virtual Distributed Laboratory
XAML	Extensible Application Markup Language
XML	Extensible Markup Language
XMPP	Extensible Messaging and Presence Protocol