AFRL-RH-WP-TR-2015-0049

SUPERVISORY PRESENTATION FOR RESEARCH, INFORMATION, INTEGRATION AND TESTING (SPRINT)

Thomas Hughes
Clayton Rothwell
Sara Naderer
Daylond Hooper
Infoscitex Corporation

Allen J. Rowe
Warfighter Interface Division
Supervisory Control and Cognition Branch
Wright-Patterson AFB OH 45433-7511

March 2015
Final Report

Distribution A: Approved for public release; distribution unlimited.

STINFO COPY

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

NOTICE AND SIGNATURE PAGE

Using Government drawings, specifications, or other data included in this document for any purpose other than Government procurement does not in any way obligate the U.S. Government. The fact that the Government formulated or supplied the drawings, specifications, or other data does not license the holder or any other person or corporation; or convey any rights or permission to manufacture, use, or sell any patented invention that may relate to them.

Qualified requestors may obtain copies of this report from the Defense Technical Information Center (DTIC) (http://www.dtic.mil).

AFRL-RH-WP-TR-2015-0049 HAS BEEN REVIEWED AND IS APPROVED FOR PUBLICATION IN ACCORDANCE WITH ASSIGNED DISTRIBUTION STATEMENT.

//signed//
Allen Rowe
Project Engineer
Supervisory Control and Cognition Branch

//signed//
Jason Clark, Chief
Supervisory Control and Cognition Branch
Warfighter Interface Division

//signed//
William E. Russell, Chief
Warfighter Interface Division
Human Effectiveness Directorate
711 Human Performance Wing

This report is published in the interest of scientific and technical information exchange, and its publication does not constitute the Government’s approval or disapproval of its ideas or findings.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. REPORT DATE ***(DD-MM-YY)*** 29-03-2015	**2. REPORT TYPE** Final	**3. DATES COVERED** ***(From - To)*** 06 March 2012 – 17 March 2015

4. TITLE AND SUBTITLE
Supervisory Presentation for Research, Information, Integration and Testing (SPRINT)

5a. CONTRACT NUMBER
In-House

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER
62202F

6. AUTHOR(S)
Thomas Hughes*, Clayton Rothwell*, Sara Naderer*, Daylond Hooper*
Allen J. Rowe**

5d. PROJECT NUMBER
5329

5e. TASK NUMBER
09

5f. WORK UNIT NUMBER
H03J (53290901)

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Infoscitex Corporation*
4027 Colonel Glenn Highway, Suite 210
Dayton, OH 45433

8.PERFORMING ORGANIZATION REPORT NUMBER
IST-ATEA-13-002

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Air Force Material Command**
Air Force Research Laboratory
711 Human Performance Wing
Human Effectiveness Directorate
Warfighter Interface Division
Supervisory Control and Cognition Branch
Wright-Patterson AFB OH 45433

10. SPONSORING/MONITORING AGENCY ACRONYM(S)
711 HPW/RHCI

11. SPONSORING/MONITORING AGENCY REPORT NUMBER(S)
AFRL-RH-WP-TR-2015-0049

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

13. SUPPLEMENTARY NOTES
88 ABW Cleared 09/04/2015; 88ABW-2015-4233. Report contains color.

14. ABSTRACT

Unmanned aerial systems (UAS) are becoming an increasingly critical aspect of military operations. To meet this demand, the USAF is seeking ever more capable UASs, to include the ability of a single operator to simultaneously control multiple platforms, increased connectivity to net-centric information sources, and the ability to accomplish more complex, dynamic missions. To fulfill these missions, the USAF is exploring multi-vehicle UAS concepts to carry out tactical intelligence, surveillance, reconnaissance and combat missions. In many of these concepts there is an emphasis on managing UAS systems and conducting missions with minimal crew. UASs have evolved from being primarily remotely controlled systems to being pre-programmed or semi-autonomous, changing the role of the crew from flying to supervising. This change has opened the door to increasing the vehicle to operator ratio. While progress has been made in developing more capability with multi-vehicle systems (e.g., more simultaneous vehicle orbits managed from one control station), further research is needed to better understand the challenges associated with multi-vehicle control across a broad range of missions and operational situations while maintaining the highest level of mission effectiveness.. To achieve these levels of mission effectiveness crew performance and capability enhancements are needed. Technology development and advanced designs are required to facilitate more timely and effective operator situation assessment and decision-making. This report describes activities performed during the development of interface technologies designed to improve supervisory control of multiple information sources (711 HPW/RHCI), Work Unit 53290901, Supervisory Presentation for Research, Information, Integration and Testing (SPRINT).

15. SUBJECT TERMS
Supervisory Control, Multi-Modal Interfaces, Scalable Interfaces, Advanced Visualization, Augmented Symbology

16. SECURITY CLASSIFICATION OF:			**17. LIMITATION OF ABSTRACT:**	**18. NUMBER OF PAGES**	**19a. NAME OF RESPONSIBLE PERSON (Monitor)**
a. REPORT	**b. ABSTRACT**	**c. THIS PAGE**			
Unclassified	Unclassified	Unclassified	SAR	41	Allen Rowe **19b. TELEPHONE NUMBER** ***(Include Area Code)***

THIS PAGE INTENTIONALLY LEFT BLANK

TABLE OF CONTENTS

SECTION **Page**

TABLE OF CONTENTS iii

LIST OF FIGURES iv

LIST OF TABLES iv

Preface v

1.0 Introduction 1

2.0 Verifiable Task Assignment and Scheduling Controller (VTASC) 3

2.1 SPEC INTERFACE 4

2.2 VTASC Findings 8

3.0 VALUE OF INVORMATION IN COLLABORATIVE SYSTEMS (VICS) 8

3.1 UGS Symbols and TLE Design 10

3.2 VICS Findings 13

4.0 Human Behavioral Modeling for Stochastic Control (HBMSC) 14

4.1 Task 16

4.1.1 Test Design 17

4.2 Proof of Concept Demonstration 18

4.2.1 Task 18

4.3 HBMITR Findings 19

5.0 Monitoring and Controlling Multiple Assets in Complex Environments (MC-MACE) 21

6.0 Fusion 21

6.1 Fusion Framework 22

6.2 Virtual Distributed Lab 24

6.3 Fusion Keystone Program Application 25

6.4 Flexible Software Architecture 26

6.5 Cloud-based Simulation Architecture 26

6.6 Software Extensibility 27

6.7 Interface Instrumentation 28

6.8 Human-Autonomy Dialog through Retrospection 29

6.9 Summary 29

7.0 Conclusions 30

8.0 REFERENCES 31

LIST OF FIGURES

Figure 1. Diagram of VTASC system components that displays the connections between the operator, the Vigilant Spirit Control Station, the NuSMV model checker, and UAVs. 4
Figure 2. Screen shot of the Vigilant Spirit Control Station with a two monitor configuration. On the left is the Tactical Situation Display, vehicle status display, and route planning interfaces. On the right is the SPEC tool. 6
Figure 3. The translator interface and its three windows. A) the proposition editor, B) the token bank, C) the current specification list. 7
Figure 4. Examples of dynamic Unattended Ground Sensor (UGS) symbols and the Target Location Estimate (TLE). 11
Figure 5: Vehicle with Target(Y) and Non-Target(V) features. 16
Figure 6. Tactical Situation Display and 4 UAV Video Display 18
Figure 7. Fusion High Level Framework. 23
Figure 8. Fusion Layered Architecture. 23
Figure 9. Fusion Visual Framework Components 24
Figure 10. Fusion Virtual Laboratory Concept. 25
Figure 11. IMPACT instantiation of Fusion. 26

LIST OF TABLES

Table 1. Ranges of Independent Variables 17

Preface

This report describes activities performed during the development of interface technologies designed to improve supervisory control of multiple information sources (711 HPW/RHCI), Work Unit 53290901, Supervisory Presentation for Research, Information, Integration and Testing (SPRINT). The authors thank the entire SPRINT team to include Mr. Antonio Ayala, Dr. Thomas Carretta, Dr. Terry Stanard, Mr. Jason Roll, Mr. Brian Donnelly, Mrs. Sarah Spriggs, Mrs. Sarah Lampke, Mr. Austin Bangert, Mr. Adam Buchanan, Mr. Mike Bowman, Mr. Bruce Clay, and Mr. Mike Howard.

1.0 Introduction

Unmanned aerial systems (UAS) are becoming an increasingly critical aspect of military operations. To meet this demand, the USAF is seeking ever more capable UASs, to include the ability of a single operator to simultaneously control multiple platforms, increased connectivity to net-centric information sources, and the ability to accomplish more complex, dynamic missions. These capabilities include:

1) close collaboration with manned assets

2) find, fix, track, and target difficult targets in complex urban and difficult terrains

3) destroy or neutralize difficult targets

4) precisely deliver select effects to maintain controllable collateral damage

5) persistence in multiple areas of interest

To fulfill these missions, the USAF is exploring multi-vehicle UAS concepts to carry out tactical intelligence, surveillance, reconnaissance and combat missions. In many of these concepts there is an emphasis on managing UAS systems and conducting missions with minimal crew. UASs have evolved from being primarily remotely controlled systems to being pre-programmed or semi-autonomous, changing the role of the crew from flying to supervising. This change has opened the door to increasing the vehicle to operator ratio. While progress has been made in developing more capability with multi-vehicle systems (e.g., more simultaneous vehicle orbits managed from one control station), further research is needed to better understand the challenges associated with multi-vehicle control across a broad range of missions and operational situations while maintaining the highest level of mission effectiveness.. To achieve these levels of mission effectiveness crew performance and capability enhancements are needed. Technology development and advanced designs are required to facilitate more timely and effective operator situation assessment and decision-making.

The Supervisory Control Interfaces Branch of the Human Effectiveness Directorate (711 HPW/RHCI) has been tasked with researching the issues associated with effective vehicle interfaces. This effort has built on work that has been accomplished to date, as well as cooperative efforts with other elements of AFRL, AFOSR, and outside organizations such as ASC, the other Services, DARPA, NASA, industry, and international coalition allies.

Under this effort, Infoscitex (IST) in support of the Supervisory Control and Cognition Branch of 711th Human Performance Wing's Human Effectiveness Directorate (711 HPW/RHCI) analyzed, designed, developed and assessed controls, displays, and decision aids to support a reduced operator to UAS ratio in a supervisory role This effort focused on:

1) Developing and testing advanced interface concepts to support target detection, classification and prioritization for operators monitoring multiple type of sensors such as full motion video, auditory, and other mission specific sensors.

2) Developing and testing intuitive methods of flexibly interacting with automation that was designed to improve operator situation awareness and overall system performance.

3) Displaying net-centric information on intuitive displays, including seamless compression of information onto portable interfaces.

4) Exploring the utility of novel multi-modal input devices (e.g., multi-touch, gesture recognition, speech), and multi-sensory interfaces to support operator supervision of multiple vehicles to maximize performance and situation awareness while mitigating negative automation-induced problems.

5) Developing and testing advanced visualization concepts utilizing novel displays, mixed reality, and immersive multi-modal interfaces (full body gesture recognition, multi-touch contextual interactions) to support mission and sensor management for multi-UxS supervisory control.

6) Developing and testing novel information rich symbology (glyphs) in both a team and mission specific environment.

In support of these objectives IST engaged in a series of programs and projects with an emphasis on exploring various mission scenarios, automation concepts, operator interface design prototypes and control station frameworks. This research was conducted through a series of individual projects each designed to address a specific aspect or aspects of the overall program objectives.

Verifiable Task Assignment and Scheduling Controller (VTASC). The primary motivation of the VTASC research initiative as to investigate a means of implementing formal methods model checking to verify that UAV assignments, routes, and on-board dynamic mission planning conform with mission specifications (e.g., objectives, constraints, and rules of engagement) during mission planning and during mission execution, thereby increasing the chances of mission success in a way that was both intuitive and easy to understand for operators engaged in highly dynamic and complex operational environments.

Value of Information in Collaborative Systems (VICS). Was to support enhance human-autonomy interaction through the investigation and development of 1) novel automation display concepts extend current geo-spatial displays by adding geo-spatial information regarding the autonomous machine perception and decision making and 2) incorporating a new display that attempts to provide a window into the higher-order activities of automation by using state machine diagrams. Affording the opportunity for operators a view across levels of task abstraction is anticipated to quicken situation assessment of the systems' tasks and improve overall situation awareness. Furthermore, system behaviors can be viewed in real time or retrospectively evaluated

Human Behavioral Modeling for Stochastic Control (HBMSC). The HBMSC project was part of an ongoing collaboration with member of AFRL/RQQC and their attempt to refine a human behavioral model to accompany their stochastic control algorithm to support semi-autonomous ISR planning and collection of select targets. The project focused on developing a standard "operator" model based those characteristics of the image capture process that have the greatest impact on operator target detection performance, use those characteristics to build a model and use the model the drive the algorithms decision process during image collection.

Monitoring and Controlling Multiple Assets in Complex Environments (MC-MACE).

Fusion. A primary motivation for the development of the Fusion framework as to conduct developmental research that would enable natural human interaction with flexible and adaptive automation via the use of intelligent agents reasoning among disparate domain knowledge sources, machine learning providing monitoring services and intelligent aids to the operator, cooperative planners and advanced simulation thru an instrumented, goal oriented operator interface that empowers scientific experimentation and technology advancement across multiple systems

The following section provides a more detailed description of the research and development conducted on each of these projects as part of the ATEA Task Order 5 SPRINT effort.

2.0 Verifiable Task Assignment and Scheduling Controller (VTASC)

Recently, researchers have applied formal methods – mathematically based languages, techniques, and tools used to design and verify the safe and reliable operation of systems – to robotic systems, including UAVs (Humphrey, 2012; Doherty, Kvarnstrom, & Heintz , 2009; Humphrey, & Patzek, 2013.). Formal methods include model checking, a technique in which a finite-state representation of a system is automatically checked against a set of desired specifications expressed in temporal logic. In the UAV domain, model checking can be used to verify that UAV assignments, routes, and on-board dynamic mission planning conform with mission specifications (e.g., objectives, constraints, and rules of engagement) during mission planning and during mission execution, thereby increasing the chances of mission success. In particular, formal methods could be used to provide a better framework for communication between human and autonomous system members, a vital element for successful collaboration/teaming identified by researchers in human-automation interaction (Klein, Woods, Bradshaw, Hoffman, & Feltovich, 2004). However, formal methods and the associated temporal logics are difficult for a novice to understand, use, or learn (Dwyer, Avrunin, & Corbett, 1999). Therefore, in order to use formal methods to communicate the operator's specifications to autonomous UAVs, an interface is required that captures an operator's intended meaning and expresses it in a formal methods framework.

As part of the VTASC effort the Specification pattern Editor and Checker (SPEC) was developed to support such an interface. SPEC supports communication between an operator, a UAV ground control station, and model checking software by providing a set of tools to write specifications in English that can be converted to temporal logic. The aspiration is to equip an operator with a tool that helps achieve the mission objectives while complying with the rules of engagement and constraints of the mission. With model checking tools integrated into the control station, it is anticipated that operators can perform complex mission planning and management tasks for single- and multi-UAV operations effectively.

There is a long and varied list of potential mission specifications within the multi-UAV domain because of the widespread use of UAVs in many types of missions. In order to focus this initial interface work, a convoy escort mission was selected. In this mission, the primary goal is to assist a convoy of ground vehicles in quickly and safely reaching its destination by providing requisite surveillance and oversight of the primary route and surrounding areas: looking ahead along the route, scouting alternate routes, investigating traffic congestion and suspicious situations, and periodically checking possible bottlenecks. To establish detailed system requirements, in-house personnel with experience evaluating multi-UAV surveillance technology participated in a cognitive task analysis (CTA; Stanard, Bearden, & Rothwell , 2013) For the CTA, each participant stepped through a convoy escort scenario using a "think aloud" protocol in which the participants explained their thought processes, strategies, and decisions to an experimenter/recorder. The scenario was designed to be challenging to the extent that we could explore the reasoning processes of experienced personnel engaged in such a demanding task. It incorporated UAVs with heterogeneous capabilities, dynamic events that rerouted the convoy, and instances in which the vehicles available to the operator could not fulfill all the task demands of the situation. The CTA results informed the present work, providing insight into how operators plan and conduct these types of missions and how they would direct multiple semi-autonomous UAVs to meet mission objectives.

2.1 SPEC INTERFACE

The SPEC components and interface (i.e., the SPEC tool) described below are integrated within the VTASC system (Figure 1). The VTASC system has many components: a human operator, UAVs that act as semi-autonomous reactive systems, a UAV ground control station known as Vigilant Spirit, a piece of software called the model builder, and the NuSMV model checking software.

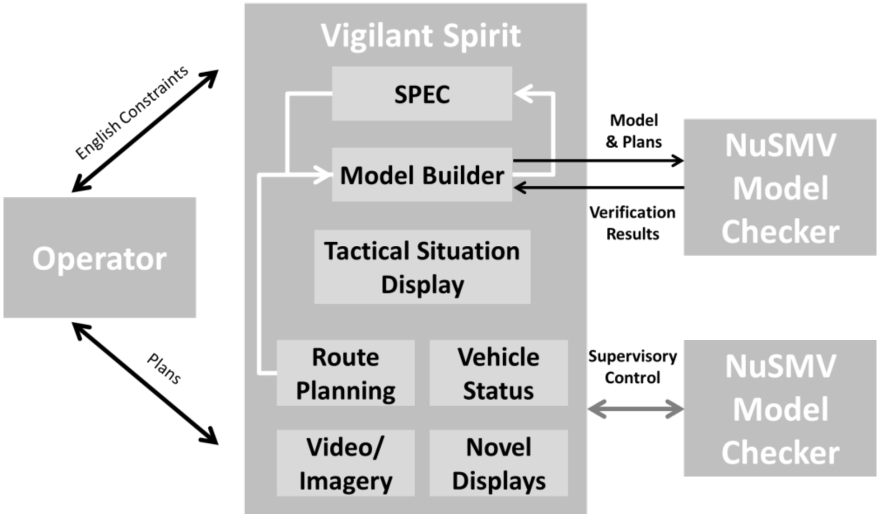


Figure 1. Diagram of VTASC system components that displays the connections between the operator, the Vigilant Spirit Control Station, the NuSMV model checker, and UAVs.

The human operator performs supervisory control of multiple UAVs, monitors their mission activities, views the images they collect, and has the ability to direct them through autopilot commands and waypoint-based flight plans. Within existing systems, UAVs respond exclusively to commands received from the ground control station. However, we envision a future when UAVs will have a flexible control scheme such that they can operate across a spectrum of varying levels of automation, ranging from teleoperated to fully autonomous. On the autonomous side of the spectrum, UAVs would behave as intelligent reactive systems, detecting dynamic elements of environments and missions and change their behavior in response. Therefore, our system has been developed to accommodate envisioned autonomous UAVs in subsequent tests. The Vigilant Spirit Control Station (VSCS) is a test bed designed by the Air Force Research Laboratory for studying multi-UAV supervisory control and associated interfaces and technology (Rowe, Liggett, & Davis, 2009; Feitshans, Rowe, Davis, Holland, & Berger, 2008). VSCS has tactical situation displays (i.e., geo-spatial maps), vehicle status displays, route planning interfaces for creating vehicle flight plans, and video/imagery displays, in addition to novel displays intended to improve human-automation interaction (the interface described in this paper is considered one of those novel displays; Figure 2). The model builder software was integrated into the VSCS that constructs a mission model that is sent to the model checker, that interfaces with the model-checking software (e.g., initiates the model checker, receives the results of the check). The model created by the model builder contains representations of the vehicles, their capabilities, and their flight plans as well as the environment. The specifications output by the SPEC interface are passed to the model builder, which adds them to the model and sends the model to the NuSMV model checker. NuSMV is a university developed symbolic model checker that accommodates many temporal logics (Cavada et al., 2010; Cimatti et al., 2002) including: linear temporal logic (LTL), computation tree logic (CTL), and real-time computation tree logic (RTCTL).

The Vigilant Spirit Control Station (VSCS) has user interfaces for UAV control/mission management and the SPEC tool (Figure 2). For VTASC, VSCS has a two monitor layout that has the tactical situation display, vehicle status, and route planning on the left screen and the SPEC tool on the right screen. No video/imagery displays are present in this layout because in pre-mission planning there is no live video to inspect.

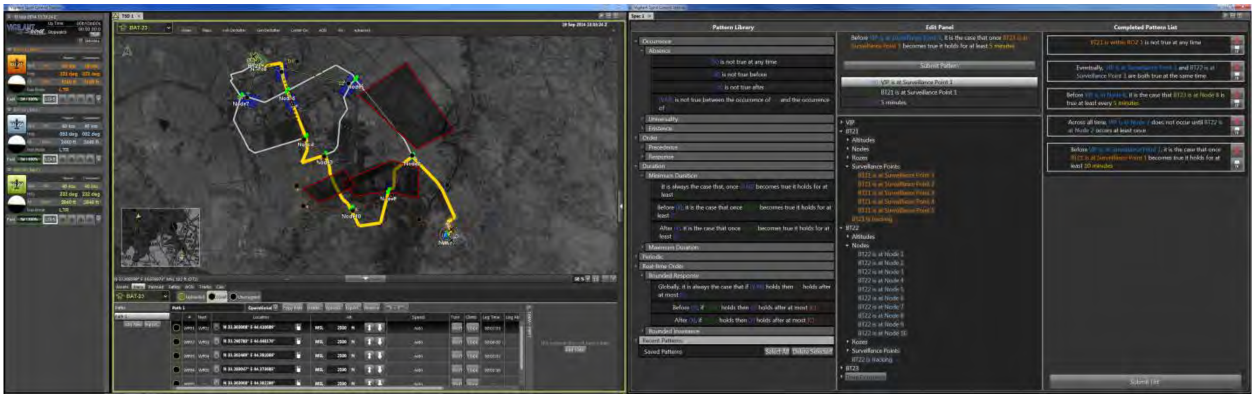


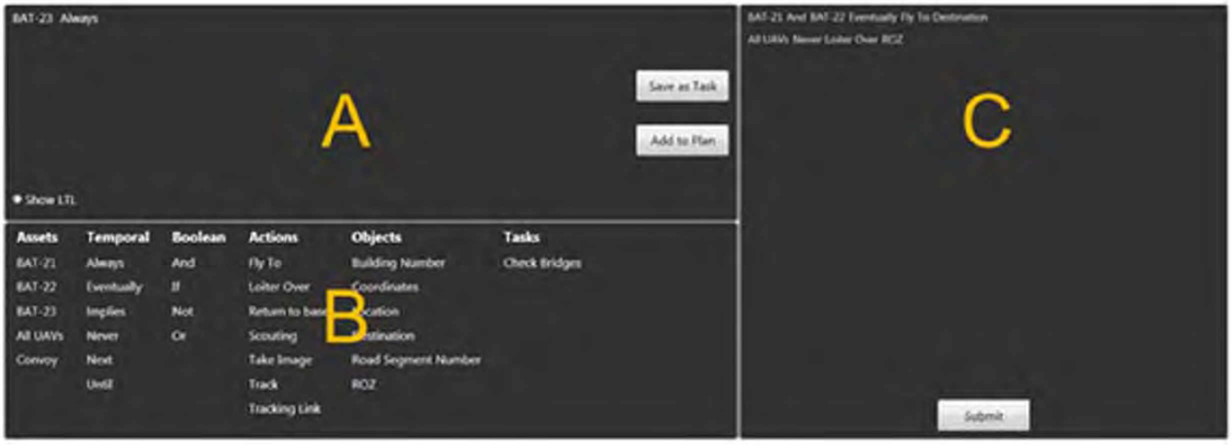
Figure 2. Screen shot of the Vigilant Spirit Control Station with a two monitor configuration. On the left is the Tactical Situation Display, vehicle status display, and route planning interfaces. On the right is the SPEC tool.

As noted earlier, the model contains specifications and vehicle plans. Both are created by the operator and added to the model. The long-term goal of this system is to have an automatic planning tool that uses the specifications to generate satisfactory vehicle plans (through a process known as synthesis), so that the operator can communicate desirable system behavior to the UAV's on-board controller in natural language and the UAV carries out the planning and re-planning as the mission unfolds. Automatic planning has not been implemented yet in SPEC. Consequently, the tool currently is used to verify vehicle plans created by the operator. To do this, the operators use the existing VSCS route planning interfaces to create flight plans for each UAV. A custom software method was written inside the model builder that converts the flight plans into the format needed for the model (i.e., a finite state transition system). These plans consist of legs of travel that are defined by waypoints. Each leg also has a speed and altitude associated with it. Each waypoint is able to have loiter operations associated with it. The model builder uses a simple air vehicle model to represent vehicles in the finite-state transition system which uses instantaneous turning and linear climbing/descending. Conceptually, the model checker verifies whether or not the vehicle plans the operator created will accomplish what the operator had intended them to (as conveyed by the specifications). The following section will detail how operators input specifications into the SPEC tool.

The SPEC interface is similar to the Specification Pattern Instantiation and Derivation EnviRonment (SPIDER) tool suite (Konrad & Cheng, 2005a). Both SPEC and SPIDER use a method for developing specifications that is based on temporal logic patterns rather than a translation-based approach (such as the approached used by the LTLMoP tool, Kress-Gazit, Fainekos, & Pappas, 2008; and our past work, Rothwell et al., 2013). A temporal logic pattern captures a temporal relationship that expresses a commonly-desired system property. Temporal relationships, such as *property P always happens at least once every t minutes*, are generic in nature such that *P* can be any property of a given system and *t* can be any one of the discrete time steps of that system. Patterns can be instantiated with variables specific to a particular system of interest and then used in model-checking. The next section will describe the features of SPEC for selecting a pattern, instantiating a pattern, and using the model checker to check the pattern against the current flight plans.

The interface is shown in Figure 3. It has three main windows: the token bank (B), the proposition editor (A), and the current specifications list (C). The token bank holds components (“tokens”) that are combined to build specifications as well as a library of saved specifications (“tasks”). The proposition window is where mission specifications are built and edited. Once built, the proposition can be added to the current specifications list. The current specification list is the list of mission specifications (propositions) that will be subject to the model checking/verification routines.

One of the unique features of this interface is the ability to build (communicate) constraints by selecting tokens. The tokens are fixed English words and phrases that cannot be edited, but were created in a way that allows propositions to closely follow natural language. Operators can drag and drop or double-click to add tokens to the proposition editor window and build mission specifications. Once in the proposition editor window, tokens can be reordered through drag and drop input as well. Token input was chosen for several reasons: 1) provide the operator some guidance as to the type of specifications that are currently supported by the translation and verification systems, 2) give flexibility to the word order of specifications (in contrast to form-letter with drop down boxes), and 3) minimize the nuisance typographical errors that can occur in free-typing input methods. The tasks column in the token bank window contains an operator-configurable library of mission specifications. Any proposition can be saved as a custom task, to be accessed later. Any task can be loaded into the proposition window and edited to fit the unique circumstances of the mission before it’s added to the current specifications list.



BAT-23 Always

Assets	Temporal	Boolean	Actions	Objects	Tasks
BAT-21	Always	And	Fly To	Building Number	Check Bridges
BAT-22	Eventually	If	Loiter Over	Coordinates	
BAT-23	Implies	Not	Return to base	Location	
All UAVs	Never	Or	Scouting	Destination	
Convoy	Next		Take Image	Road Segment Number	
	Until		Track	ROZ	
			Tracking Link		

BAT-21 And BAT-22 Eventually Fly To Destination

All UAVs Never Loiter Over ROZ

Figure 3. The translator interface and its three windows. A) the proposition editor, B) the token bank, C) the current specification list.

In addition to tokens, mission specifications can be built from spatial inputs as well. The map display of a ground control station can be used to make spatial references. Operators, through clicking on the map, can add latitude and longitude coordinates to a mission specification. Also, operators can select objects from the map: route waypoints, buildings, restricted operating zones, areas of interest, vehicles, and road segments with the mouse.

Upon completion of the develop effort a series of usability test were conducted to assess the overall utility of the capability as well as the usability of the resultant interface. Specifically the goals usability of SPEC tool within the context of a mission use case. We solicited feedback on the pattern concept, the method of creating specifications with the SPEC tool, and presentation of the model checker output. A total of six subjects participated in the assessment. Following a detailed overview of the goals of objectives of the project and an introduction to the interface, participants were given the opportunity to interact with the VSCS through a series and representative training scenarios. He training used demonstrations and hands-on expercieses with at least one pattern from each category in the pattern library to ensure participants were familiar with the technology prior to initiating data collection trials. The mission scenario was based on convoy escort in an urban environment using 3 UAVs to provide oversight and look ahead. Participants were instructed to use the interface to establish specifications for the task to be completed and develop a set of mission plans for each of the vehicles that satisfied those specifications. Our hypothesis suggested that the SPEC interface would provide an innovative and intuitive means of creating and checking resultant plans to ensure that that satisfy the set of constraints imposed by the demands of the surveillance mission.

2.2 VTASC Findings

Based on our observations and data captured during the usability testing accomplished the SPEC interface provides a way to use model checking tools for UAV mission planning. It was developed so that UAV operators could create mission specifications and use model checking software without having to learn temporal logics. It attempts to accomplish the mission specification writing through temporal patterns rather than machine translation. A usability study was conducted on an initial SPEC display. Participant feedback was generally positive, showing the promise of these kinds of technologies. The usability study also generated many ideas for further development and improvement of the SPEC tool, however; only a few topics approached or achieved a consensus. Participants expressed a desire for more feedback from the model checker, participants found the pattern wording difficult to understand at times, and participants found the Edit Panel’s variable tree hard to navigate. We recommend that future research address these issues through additions to the SPEC interface. Feedback could be increased through providing an explanation of where and how the vehicle plans violated the specifications. Pattern understanding could be improved through an additional graphical representation that would supplement the text representation of each pattern. The variable tree in the Edit Panel could be improved by altering the hierarchical expand and collapse behavior, providing alternate input methods to the tree, and automatic advancement once a variable has been selected. In addition, future research should experimentally evaluate the benefit, if any, model checking tools provide during mission planning.

3.0 VALUE OF INVORMATION IN COLLABORATIVE SYSTEMS (VICS)

To enable more robust and adaptive capabilities, the Air Force Research Laboratory (AFRL) initiated the Value of Information in Collaborative Systems (VICS) program with the purpose of developing technology that enables collaborative (i.e., de*centralized*) control of a flexibly autonomous system. The envisioned system is characterized as being composable and fractionated (United States Air Force Chief Scientist, 2010) where heterogeneous members, both human warfighters and unmanned systems, within the system can assemble and combine their respective capabilities and strengths to achieve mission objectives. The heterogeneous components, under varied communication conditions, can coordinate with each other and have the freedom to perform decentralized planning and decision-making as the situation dictates. The increased autonomous capability (i.e., on-board, dynamic mission planning technology) of the components allows the system to carry out missions without a C2 communication link, without a payload communication link, or without either link. Furthermore, the increased autonomy allows the operator the flexibility to offload mission management to the machine if he/she chooses, even when a C2 communication link is present. The capabilities developed represent a shift from centralized control to a collaborative and decentralized form of control. The ultimate goal of VICS is to enable agile and adaptive mission management and control for a team that is comprised of unmanned aerial vehicles (UAVs), unattended ground sensors (UGS), dismounted warfighters with mobile control stations, and an operator located in a central control station.

The challenge in supervisory control interfaces is two-fold. First, supervisory control interface concepts must display higher-order mission management and control information. This information should enable effective supervisory control of the collaborative, autonomous system. Displays should clearly show information pertaining to the vehicles' progress towards mission goals, tasks, and the underlying rationale for plan changes during mission execution. The goals, tasks, and rationale are intended to provide a window into the vehicle's perception and assessment of the situation. Second, in addition to providing more detailed information concerning the information processing and behavior of the autonomous system, there is a challenge to present the information in a manner that affords quick and accurate assessment of the multi-vehicle system. With this in mind, our goal was to build a design concept that uses symbols and patterns in an attempt to provide "at a glance" recognition of complex activities.

We proposed novel automation display concepts comprised of two main features. First, we extended current geo-spatial displays by adding geo-spatial information regarding the autonomous machine perception and decision making. This display layer sits on top of the map and shows the current search area of the UAVs and the outputs of their search algorithms (described further below). Second, we added a new display that attempts to provide a window into the higher-order activities of automation by using state machine diagrams (Patzek, et al, 2013). This display is a hierarchically arranged set of nested state diagrams called Layered Pattern Recognizable Interface for State Machines (L- PRISM). High-level goals and tasks are at the top-most tier and low-level sub-tasks or states at the bottom-most tier, providing different levels of abstraction. Individual tiers are designed to have unique layouts in an attempt to facilitate pattern recognition, which in turn, is anticipated to quicken situation assessment of the systems' tasks and improve overall situation awareness. Furthermore, system behaviors can be viewed in real time or retrospectively evaluated.

In conceptualizing these new interface concepts, we examined design visualization techniques used to illustrate and represent complex, multi-state systems and processes with the idea that these methods could be made dynamic and animated to represent real-time (and/or past) activities. Diagram and graph methods such as flowcharts, binary decision trees, goal graphs, finite state machine diagrams, and petri nets were assessed for their potential application for an intuitive dynamic display of the autonomous system. Finite state machine diagrams were selected given their efficiency to display multiple systems traversing multiple tasks and states, and the ability to nest states within tasks (Harel, 1987) to represent rules, constraints and overall mission decision logic. Furthermore, the arcs or directional lines between states represent the conditions that must be met by the systems to be able to move from one state into the next, which supports the notion of providing the operator the rationale for autonomous mission, task, and state changes. Another reason for choosing finite state machine diagrams as the basis for the interface was that they can be formed in a manner to produce unique layouts or patterns, supporting the goal to develop an interface that fosters efficient recognition of the activities. Finally, a form of a state diagram user interface has been demonstrated for robot mission planning and representing robot tasks (MacKenzie, Arkin, & Cameron, 1997; Endo, MacKenzie, & Arkin, 2004).

In addition to the state diagram concept, L-PRISM includes a control timeline and payload viewer to navigate and inspect the past events and associated details, including images, videos, audio recordings, and text messages. L-PRISM is designed to be integrated with the multi-UAV control station's tactical situation map and system status information to assist the operator to not only be aware of vehicles' locations and planned routes but also their mission goals, associated tasks, and states to achieve the mission goals. The remainder of this paper will describe the automation display concept, experimental assessments of the interface, and results.

3.1 UGS Symbols and TLE Design

Part of the UAV automation designed under VICS was intended to conduct road monitoring-the continual patrol of a road network for intruding ground vehicles using remote sensors called unattended ground sensors (UGS). The UGS were distributed on different segments of the road network and used radar to detect moving objects. These detections were relayed to the UAVs and used for a search decision algorithm that used Bayesian statistics and information theory to determine the possible current locations of the target vehicle and candidate UGS that might narrow the search space (Casbeer, Meier & Coa, 2013; Chen et al., 2014). We developed symbols for the control station's tactical situation display to represent the state of these sensors and the target location estimation algorithm. The UGS symbols had three states and changed color depending on state: the UAV had no information from the UGS (depicted in white), the UAV had information but there were no detections (depicted in blue), and the UAV had information and there was one or more detections (depicted in red). The UGS symbols also represented how old the UGS detection was by showing a solid colored square in for new detections and fading out to only colored outline as time progressed.

The estimate of the target vehicle's location (or target location estimate; TLE) was depicted as highlighted segments of the road based on the probability values output by the Bayesian calculation. Probability values were grouped into bins before being displayed to create discrete levels of probability. Highlighting represented probability following a redundant coding scheme using transparency and width. More probable segments were highlighted with opaque red with a thick width and less probable segments were displays as semi-transparent red with a thin width. Figure 4 shows red UGS symbols, blue UGS symbols, and the TLE (white UGS symbols not shown).

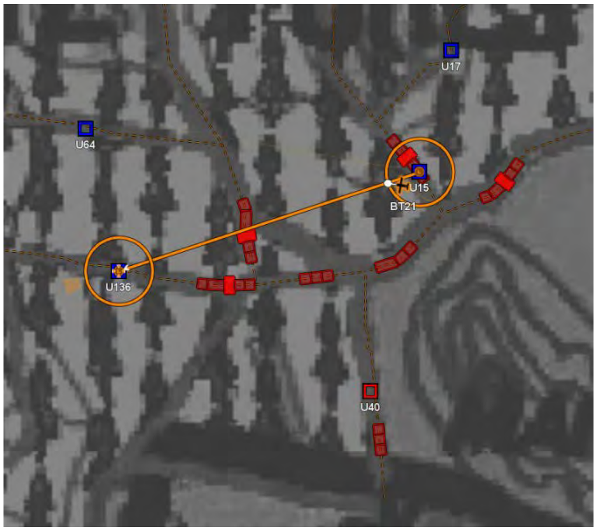


Figure 4. Examples of dynamic Unattended Ground Sensor (UGS) symbols and the Target Location Estimate (TLE).

Our first experiment utilized the road monitoring automation from the VICS program. As a developmental milestone, the simulation used simulated UGS and UAVs with software modules that contained the actual decision logic of those items. State diagrams that appeared in L-PRISM were developed to represent the road monitoring automation for this task in order to test L-PRISM's effect on supervisory control (See Appendix A). We hypothesized that with the assistance of the L-PRISM display, the UGS coloring, and the TLE, participants would experience higher situation awareness, lower workload, and higher performance while completing the task than without the novel automation displays. We also expected that the new automation displays would assist the participant more in the communication limited environments than situations with more persistent communication. More persistent communication would also increase workload and performance.

We found that communication level affected workload ratings. Control station configuration did not affect workload ratings, objective situation awareness (SAGAT scores), or capture times. We suspected that the automated behaviors of the UAVs were not sufficiently complex or unpredictable so there was no need for or benefit from the L-PRISM and automation displays. This suspicion was reinforced by the relatively low use of the timeline and retrospective modes of L-PRISM display. To address this limitation, we designed a second study that used other automated tasks in addition to the automated road monitoring as well as increased the decentralization of communication. In addition, the second study addressed the variability in trial length that led to missing SAGAT data and missing capture data.

The second study adapted the UGS technology to a combat search and rescue (CSAR) mission. The behavior of the UGS was changed so that they responded to the blue force warfighter who needed to be extracted ("Agent X"). Four different rescue mission scenarios were created that followed a similar progression: 1) UAVs searched for Agent X's position, 2) Agent X was located, 3) the convoy of ground vehicle's set out to reach Agent X while UAVs supported the convoy by providing overwatch and look ahead, 4) The convoy returned to base. Differences in missions were whether or not the convoy was able to reach Agent X, whether or not Agent X had to change locations before the convoy arrived, and whether or not Agent X was present when the convoy arrived. Additional differences related to the number of times the convoy had to reroute because of threats encountered along its way.

To increase the complexity of autonomous behavior that participants had to supervise, more automated tasks were created for this study. The tasks used were: road monitoring, overwatch, look ahead, scout, point surveillance (See Appendix A). Road monitoring was the visiting of UGS for detections of Agent X. Overwatch was a UAV loitering about the convoy's position to provide them with a real time imagery of their location and the immediate surroundings. Look ahead was imaging on the convoy's current route ahead of the convoy in order to detect threats on the route in advance. Scout was imaging the alternative convoy routes to assess their viability for reroute the convoy onto them, if necessary. Point surveillance was providing imagery on a stationary point, such as Agent X's position once he was located.

Participant interaction with the interface was altered from Experiment 1. In order to create the appearance of increased complexity of the autonomous behavior, UAV behaviors were scripted. Due to this scripting, participants were required to supervise the automation at work instead of being able to interact or direct the automation. Their task was to understand what was going on and be able to report back.

The communication levels in this study were different from the levels in Study 1. Communication levels were changed to increase decentralization and periodicity of communication relative to Study 1 with the hope that this would increase the need for the retrospective capability of L-PRISM. Communication levels are described in more detail in the method section below.

To directly investigate the monitoring of autonomous behavior, we created a fault detection task. Each mission scenario had 2 to 3 faults present in it that participants were to watch for and verbally report to the experimenter. Faults that appeared were: low fuel, vehicle off course, vehicle entering a no fly zone, and mechanical warnings (e.g., oil pressure, oil temperature). Some faults have a visual alert in the VSCS that appears in the top-right corner of the tactical situation display (TSD) but some did not.

We found that neither communication level or control station configuration affected situation awareness, workload ratings or fault detections. We speculated that the lack of significant data was caused by several factors. Primarily, we suspected that our experiment lacked the complexity that we had desired. While the experimental scenario had more vehicles performing more tasks, the scenario plan remained predictable. For example, the road monitoring vehicles always had to complete their task of finding Agent X before the overwatch and look ahead tasks could begin. This structured timeline of events possibly gave the participants a way to follow the sequence of events despite communication condition and control station configuration. This explanation was supported by the low use of the timeline and retrospective capability of the L-PRISM display.

Secondly, we speculated that the lack of significant data could have been influenced by the lack of direct interaction participants had with the scenario. Unlike Experiment 1, participants in

Experiment 2 were unable to influence the automation of the UAVs or control them directly. Instead the participants were put in a supervisory role and were told to understand what was going on and report any inappropriate behavior to the experimenter (fault detection). Having supervision and scenario awareness as the primary task, participants were able to stay at a consistent awareness though both communication conditions and control station configurations. Insignificant SAGAT and SART data was consistent with this idea.

3.2 VICS Findings

We initially hypothesized that when aided with the automation displays participants would experience higher situation awareness and performance than participants without the automation displays while also showing a decrease in workload. We also hypothesized that as communication levels got more decentralized, situation awareness, workload and performance would decrease. After two experiments we found little evidence to support our hypotheses.

We found that the control station configuration had no effect on situation awareness or workload. Participants experienced similar levels of situation awareness and workload in both control station configurations. Participants also achieved similar performances with both control station configurations. In Experiment 1 participants had similar capture times and in Experiment 2 participants achieved similar levels of fault detection.

Results from communication level data had a significant effect on overall workload. In Experiment 1, as the communication level became more decentralized, workload decreased. Specifically, when participants did not have long range payload link between the UAV and VSCS. Experiment 1 also showed significant results in perceived situation awareness, showing that participants felt more aware as the communication level became more centralized. Objective situation awareness did not vary significantly in either experiment, and Experiment 2 also showed no significance in subjective situation awareness or workload. Neither experiment found communication level having a significant effect on performance, however in Experiment 1 we saw a trend suggesting that performance increased as communication level became more centralized.

Participant comments seemed to correspond with the results we found. In Experiment 1, participants found the task easy enough to follow that they did not use the aid of the state diagrams or the timeline. Participants in Experiment 2 mentioned that they found the L-PRISM interface provided excessive information than what was needed for the task at hand.

After conducting the studies, we found weaknesses in our experimental design that could have affected the results. One issue that we looked at was the complexity of the experimental scenarios. In Experiment 1, we found evidence to suggest that the scenarios were not sufficiently complex enough for participants to find benefit in the L-PRISM control station design. This suspicion was confirmed in debrief interviews with the participants. Their comments suggested that they thought the task was straight forward and the L-PRISM display didn’t give them an added advantage to the task at hand. Some did mention that they thought maybe a more complex scenario would make the L-PRISM display more beneficial. Experiment 2 tried to address these issues, but ended up having weaknesses of its own. First we suspected that even though we increased the complexity of Experiment 2 by increasing the number of tasks and the amount of UAVs, we did not increase the uncertainty of the scenario. Each scenario still followed a basic timeline, making them predictable and possibly making components of the L-PRISM display, like the timeline, unnecessary. Secondly, due to time constraints, we chose to script the autonomous behaviors for Experiment 2 to increase capabilities over Experiment 1 with the trade-off of not having the ability to directly interact with it. Due to this drawback we had to remove the participant's ability to manually control any of the UAVs. By doing this we changed the experiment to a solely supervisory task. We believe that this could have caused an artificial inflation of the participant's situation awareness during the task and therefore caused the L-PRISM display to be unneeded.

After analyzing our results we concluded that our L-PRISM control station design requires more testing and we recommend a series of improvements to the experimental design for future study. First we would recommend spending more time working with the robustness of the UAV automation in the experimental scenario. By making the automation more robust we could allow for more direct interaction between the operator and the UAVs. A more interactive relation would increase the authenticity of the scenario and in turn provide a more realistic demand for the L-PRISM interface. We would also choose to update the complexity of the scenario and increase the amount of uncertainty. This change would give participants a more dynamic and realistic situation, causing them to need to increase their situation awareness to complete the mission successfully.

4.0 Human Behavioral Modeling for Stochastic Control (HBMSC)

Recent unmanned aerial vehicle (UAV) cooperative control technology efforts have focused on developing and demonstrating multi-vehicle assignment, scheduling and route planning for reconnaissance, surveillance and target acquisition (RSTA) tasks (Feitshans, et. al., 2008). These problem-solving capabilities can produce optimal or near-optimal solutions to minimize mission time or total distance travelled by a team of UAVs as they visit designated points of interest (POIs) to inspect potential targets. These capabilities have been shown to reduce operator workload and the time required to generate solutions compared to performing these tasks manually. The human-automation interaction, however, is fairly rigid, limiting flexibility in mission planning and execution. In an attempt to make the mission planning and management more agile and robust, efforts are underway to develop both adaptable and adaptive automation and interface technology. The adaptable automation provides more options for the human, enabling the operator to make major or subtle adjustments to the plan and associated calculations. The adaptive automation goes a step further with the machine having the ability to initiate changes to the plan based on the situation.

In pursuit of adaptive automation methods, the objective of this research was to develop a reliable Integrated Human Behavioral Model and Stochastic Controller (IHBMSC) (Chandler, et.al., 2014) that could be coupled to the vehicle assignment, scheduling and path planning automation. The goal was to demonstrate a proof of concept whereby the stochastic control automation would perform both a priori and real-time mission planning based on: 1) a dynamic and uncertain mission situation (modeled as a stochastic environment), 2) vehicle resources, and 3) an individual's capacity to perform the RSTA task effectively. The effort developed a human operator model that represents an individual’s performance tendencies and biases across a range of target acquisition and workload situations (e.g., arrival rate at POIs, dwell time over POIs, etc.). The idiographic-based model was integrated with the stochastic controller (SC) and cooperative control algorithm, leading to a more “closed-loop” form of problem-solving that not only accounts for the vehicles’ capabilities but also the individual operator’s behavior and performance in the uncertain environment.

The research approach leverages and extends current developments in stochastic control optimization (Holsapple, et.al., 2008, Pachter, Chandler, & Darbha 2006) by introducing a representation of an individual’s behavior to the stochastic controller. The goal of the embedded stochastic controller is to plan vehicle assignments, schedules and routing to reduce the probability of false alarms without adversely affecting the target detections using a priori and real-time situation and performance data. The initial stochastic controller design for sequential inspection of objects explored the decision of whether or not a UAV should revisit the point of interest to maximize the total expected information gain given there are multiple points to visit, limited resources, a set of corresponding conditions (position, time, and workload) to factor in, and the operator’s target assessment (Holsapple, et.al., 2008). The stochastic controller’s revisit decision criteria were optimized based on a Bayes rule information theoretic metric. For this research, the logic behind the SC was the following: On the first inspection (i.e., “first look”) of each POI, the operator makes a call as to whether or not the object is a target. For the first look, the stochastic mission environment may or may not have resulted in a poor view of the POI for the operator. The SC is designed to decide if and how the POI should be revisited. If the SC decides to revisit, it uses the operator model to determine an “optimal” revisit path that will result in a good view of the POI. Note that the SC may also decide not to revisit the object. To determine if the POI should or should not be revisited, the SC takes into account all of the following: the state for the first look, the operator’s response, the number of revisits left (e.g., fuel remaining) and the likelihood of the object being a target (or non-target), which is modeled by the a priori probability of target in this mission environment (also known as target density).

In pilot testing that helped identify the initial set of stochastic control parameters, and in another experiment with related multi-UAV tasks involving visual inspection of targets (Carretta, et.al., 2009) a wide variance in performance was observed across operators. Given this, a human behavioral model that is based on the group’s average performance may not represent a significant portion of the users very well, resulting in mismatched plans and UAV actions for a number of end-users. The research reported here takes a different approach by representing the individual’s behavior and performance in a computational human behavioral model (also referred to as the “operator error model” in this paper) which, in essence, will customize the problem-solving and resulting mission plans for the individual operator at the control station. During the mission, the controller can be used to monitor the actual states (e.g., ground-sampled distance, aspect angle, and dwell time) and compare it to the expected states, computing the probability of detecting a given target. When disparities exist between what was reported versus expected, or where conditions are beyond prescribed levels, the system potentially can adapt mission plans and actions, manipulating such things as the UAVs’ routes, automated sensor steering action points, and the employment of video inspection tools (e.g., digital video recording methods, mosaics).

4.1 Task

The RSTA sequential inspection task required the test subject to view an object in a video and classify the object as a target or a non-target. Two seconds before the object entered the field of view, an alert appeared on the interface to signal the test subject. The object was a white van with a black letter on the side, (Y vans were targets and V vans were non-targets) as presented in Figure 6. The test subject viewed the object and then responded either Target or Non-Target by pressing a button on the interface.

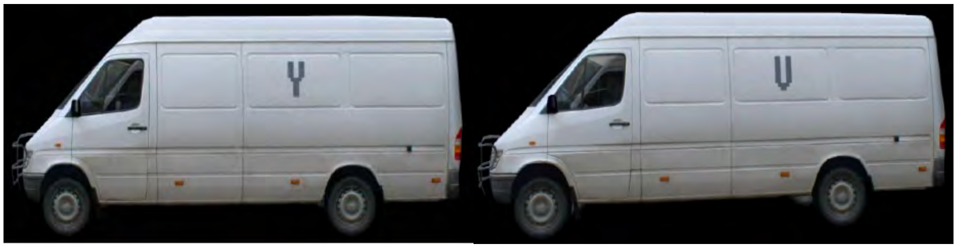


Figure 5: Vehicle with Target(Y) and Non-Target(V) features.

Confidence ratings were collected after each Target/Non-Target response. The confidence ratings were defined as: 1 = no confidence; 2 = low confidence; 3 = moderate confidence; 4 = high confidence; 5 = very high confidence. The test subject verbally reported his/her confidence rating to the test administrator, who recorded it. Confidence ratings were collected for a number of reasons: 1) to investigate the relationship between operator confidence and performance, 2) to enable some bias analyses with signal detection theory (not presented in this paper) that were not possible with only Target/Non-Target responses, and 3) to assess their potential to further improve system performance if they could be incorporated into the Stochastic Controller (SC).

In the task, the test subject was instructed to "do the best you can" (i.e., maximize accuracy) and no performance feedback was provided during training or testing. The test subject was instructed to sit a fixed distance from the monitor. No chinrest was used, but the distance was indicated with a marker and monitored by the test administrator. There were 20 POIs per run and each POI had one object to inspect. Half of the objects were Targets and half were Non-Targets. In addition, 12 POIs were revisited in each run because the SC was incorporated into the system during all testing. The initial views of POIs combined with the revisits resulted in 32 observations per run. Each run lasted approximately 13 minutes. Between runs, the test subject was given a short break that also allowed the simulation software to be reset with a different scenario. There were four scenarios used in the study. Each scenario had the same arrangement of POIs with a random arrangement of target objects. The test subject experienced scenarios in sequence to maximize the time between repeating a scenario. Test sessions never entailed more than 5 runs at a time, and typically one test session was conducted per day. On the few days that had two test sessions, one was held in the morning and one in the afternoon to allow a rest period of more than an hour.

4.1.1 Test Design

The study had four variables that defined the condition (i.e., "state") under which the Unmanned Aerial Vehicle (UAV) flew by an object. These variables were: Ground Sample Distance (GSD), Aspect Angle (AA), Dwell Time (DT), and Inter-Event Time (IET). GSD is a measurement of the resolution of a digital image referring to how much distance of ground is covered in one pixel. A smaller ground sample distance corresponds to a higher resolution image. GSD was manipulated by fixing the camera zoom of the UAV and changing the altitude of its flight path. GSD, in combination with the constants of display size and the test subject's fixed distance from the display, effectively manipulated the visual angle of the letters (i.e., their size on the retina). AA is the angle in two-dimensional space between the vertical side of the van displaying the task letters and the UAV's flight path. DT is a measurement of how long the task letter was presented within the simulated UAV sensor (i.e., the observation interval). DT arose from the time it took the letter to move across the sensor's field-of-view, which was a fixed distance on the display monitor. Therefore, DT co-varied with the letter's velocity on the display monitor. IET is a measurement of how much time elapsed from the end of one observation to the start of the next observation (i.e., inter-stimulus interval). IET was used as a way of affecting operator workload, analogous to the event rate in vigilance tasks.

Table 1. Ranges of Independent Variables

	GSD	AA	DT	IET
Minimum	0.1909 $\frac{ft}{pixel}$	45°	2.24s	0s
Maximum	0.2757 $\frac{ft}{pixel}$	90°	4.55s	500s

A planner was used to generate a UAV trajectory to visit all the objects. These variables were randomly sampled from uniform distributions to simulate the stochastic nature of realized flight path (Table 1). This random sampling procedure is different from most psychological methods (e.g., method of constant stimuli), but captured important variability that is present in the applied setting. For revisits, the premise was that once the UAV flew over the object and the object was in the image frame, then the true state could be determined. For example, if the size of the object was known a priori, then the UAV height above terrain could be computed as well as an altitude bias at that location. Also, knowing the ground speed, airspeed, and heading, the local wind vector could be estimated. Given the local altitude bias, the actual object pose angle, and the local wind vector, then a revisit state command could be computed that yielded the realized optimal revisit state computed by the stochastic controller.

The simulation approximated the previously described procedure and uncertainty. For the initial object visit state, or first look, the planner algorithm sampled the altitude, aspect angle, and speed (dwell) from uniform distributions to yield the commanded and realized state. For the revisit, or "second look," the optimum state commanded from the stochastic controller usually was achieved by the simulation, however a small amount of variability was present for a subset of second looks.

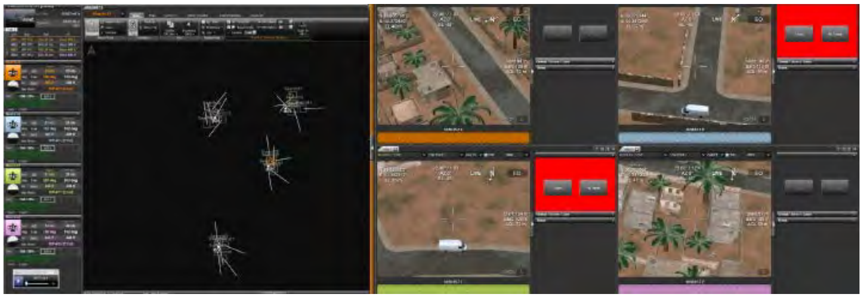


Figure 6. Tactical Situation Display and 4 UAV Video Display

The TSD is shown on the left side of Figure 7. The test subject could see the 4 UAVs and their point search tasks at a compressed scale. The video display is shown on the right side of Figure 7. It shows the simulated feeds streamed from each of the UAVs. The video images were generated using MetaVR©. There was a 3-dimensional terrain database and cultural objects (i.e., models) defined for the simulated video. The individual UAV dynamics and sensor motion were simulated on the VSCS. UAV positions and states were continuously streamed to the graphic computers to define a viewing state into the 3-dimensional database. The view at this state was then sent to the VSCS and displayed on the individual video panes. The Target/Non-Target buttons were located to the right of the corresponding video. The alert appeared behind the buttons as red fill and is illustrated in two of the video panes.

4.2 Proof of Concept Demonstration

The demonstration conducted was an assessment of the Stochastic Controller (SC) as well as the creation of an operator model. For the assessment, we expected that the operator and SC combined (performance with revisits) would perform better than the operator alone (performance without revisits). In order to conduct this assessment, we needed to conduct human-in-the-loop test using an operator error model of that individual in the SC. We had to create that model prior to conducting our test since no such model existed. The operator error model in the SC had to capture how an operator’s performance varies as a function of four variables of interest: Ground Sample Distance (GSD), Aspect Angle (AA), Dwell Time (DT), and Inter-Event Time (IET). We expected that operator error would be higher with higher values of GSD than with lower values and operator error would be higher with lower values of AA, DT, and IET than with lower values. No predictions were made concerning interactions between variables, however, we expected that model development could be made more efficient if no interactions were found between variables because future operator models could be developed by modeling the impact of each variable independently without needing to factorially combine variables. One member of the research team volunteered to participate as a test subject (S1). S1 had normal vision and was 21 years old. She was experienced with psychological research tasks and was informed of the purpose of the study.

4.2.1 Task

The test subject was introduced to the interface and the task prior to beginning data collection. The test subject was told that there were an equal number of targets and non-targets in each run. For the model creation effort, the subject performed 120 runs that each had 10 Target and 10 Non-Target trials (2400 trials overall). This number of trials would provide approximately 75 samples of each of the 16 modeled states (median-split combinations of all four variables) for target and non-target trials. Some technical issues with the data logging software resulted in losing a small number of trials (<1%), leaving 2,395 trials for creation of the operator error model.

Important to note, model creation runs also included the SC in the system loop, meaning that each run had 12 second look trials in addition to the 20 first looks. For the SC to function within the system loop during initial model creation runs, the SC used an operator model constructed from a preliminary dataset. This model was replaced by an interim model for S1 after 25 runs and updated after 75 runs. The preliminary model and the interim models were relatively similar and changing them did not appear to impact performance during testing. Once the complete 120-run operator model was created, an additional 10 runs were conducted using that model to assess the performance impact of the SC.

4.3 HBMITR Findings

Through this study we were able to create a model of an individual operator's performance during a target detection task. Then we were able to integrate this model into UAV control automation for a proof of concept demonstration. For the ranges sampled, the variables of GSD, AA, and DT affected the operator in the expected direction (as indicated by the direction of their slope). Low values of GSD increased performance compared to high values and high values of AA, DT, increased performance compared to low values. GSD had the most drastic effect on operator performance. The variable IET had little to no effect on performance. One specific IET that we were interested in was the occurrence of overlapping observations–instances in which the operator would have to split their viewing time between multiple video feeds. We expected that overlap instances would impact performance because they are related to how much time the operator has to view the POI. There were small sample sizes of 3 or 4 simultaneous letters onscreen (41 trials, or 0.34%) because overlap instances and IETs in general were not designed into the stimulus conditions, as previously mentioned. To increase the sample size, all overlap trials, including instances of 2 simultaneous letters onscreen, were grouped together (559 trials, or 4.66%). To further increase the sample size, we calculated a "generous" measurement of overlap, which began at the pre-observation cue that alerted operators to the upcoming inspection point and ended at the time of response (845 trials, or 7.05%). For all the ways we identified overlap trials, the difference in percent correct between overlap trials and all other trials was less than 1%. One difficulty in assessing the effect of overlap in this data set is that overlaps contained both first looks and second looks. Second looks were consistently very high in accuracy, and so reduced the variability in performance.

The results showed that the stochastic controller was able to aid the operator in the target detection task. The SC reduced the FA rate by nearly a factor of 8, reduced the MD rate by a third, and increased the PC by nearly 20%.

As was previously noted, previous work has shown individual differences in target acquisition which led us to investigate idiographic models of human performance. To assess the robustness of the idiographic modeling approach, we tested the performance of operators with the Stochastic Controller (SC) when the SC contained a model of a different person. If human behavior under these viewing conditions is sufficiently similar, then overall performance with the SC should still benefit from the ‘mismatched’ model compared to unaided performance.

Two members of the research team volunteered to participate as test subjects (S2 and S3). Both subjects were male with normal or corrected to normal vision and ages ranged from 24 to 28 years (mean = 26). Both subjects had previous experience in psychological research and had full knowledge of the research goals and task.

This test used the same test apparatus and task from the proof of concept demonstration. In addition, the SC used the operator model from S1 for both subjects. Both test subjects were trained in the task, one subject trained for 2 runs and the other trained for 5 runs, with sufficient performance for participation set to PC $\geq$ 70% correct. A total of 17 runs per subject were conducted, resulting in 340 trials per subject for analysis.

Data from the second study showed a large degree of similarity across test subjects. Unaided performance for the different subjects was similar in the task, as well as the direction and magnitude of the benefit of the SC. This data suggests that an operator error model for AA, GSD, DT, and Inter-Event Time (IET) may be robust and generalizable to different subjects that are similarly trained and selected.

This work tested the concept of representing human performance to control automation via an operator model for the purpose of increasing target detection performance. The demonstration created an operator performance model for values of GSD, AA, DT, and IET. The model was created using descriptive statistics and had 16 states that arose from high/low combinations of the four variables. This model was incorporated into the control automation, called the Stochastic Controller (SC), and validated. The operator aided by the SC outperformed the unaided operator in false alarms and missed detections. A second test was conducted to determine if the benefits of the SC and operator model would be robust to different operators that were not used to create the model. This study showed that different operators could still benefit from the SC when the SC has a mismatched model. One limitation of the robustness test is that only 2 subjects were used. Perhaps the model was similar to these subjects’ performance by chance. A future study should increase the number of subjects to better test the robustness of the model and use a more rigorous test procedure. One potentially rigorous procedure would be to create a complete operator error model for additional test subjects and then compare the differences between their models and test the performance with each other’s ‘mismatched’ models. Perhaps for variables of GSD, AA, DT, and IET operators aren’t similar when studied in this more thorough fashion. Moreover, other variables of interest mentioned above may show additional differences between operators (e.g., contrast sensitivity is known to vary between people and result in differences in performance on target ID tasks).

This research used UAV control automation to determine whether or not to revisit POIs. The revisit decisions from the SC are optimum from an information theory perspective. Future research is planned to compare the SC revisit strategy and resulting performance to a human’s revisit strategy (full manual revisit decision) and a human receiving revisit suggestions from the automation (management by consent). With the data from the previous two studies, subjects’ confidence ratings were highly correlated to their target detection performance. We expect that humans would choose to revisit when confidence was low and therefore there would be little to no difference between human revisit decisions and the theoretically optimum decisions of the SC. Moreover, fully automated revisit decisions are likely to face acceptance challenges during integration in a UAV control situation, especially when the revisit decisions contradict those of the operator.

5.0 Monitoring and Controlling Multiple Assets in Complex Environments (MC-MACE)

One of the challenges facing operators within current military operations centers such as Tactical Operations Centers (TOC), Combined Air Operations Centers (CAOC), Distributed Common Ground Station (DCGS) is that there is no intuitive, naturalistic way to visualize the huge volume of data and information representing an operational environment, particularly at the tactical level. The integrated the multiple frames of reference (e.g., spatial, temporal, conceptual) increases the difficulty in synthesizing the vast amount of data and information that allows operators to extract meaning and make decisions. Beleaguered operators often struggle to integrate information from different sources and must resort to the selection and display of subsets of shared databases to create user-defined operational pictures (UDOP) to perform their role.

This problem is not unique to the DoD. It's exactly what the first responders to the 9/11 terrorism faced as well. In their case it was compounded by each response agency working with their own stovepipe system. DHS has since contracted for systems to be created that better link first responders and give them a better picture of the operational environment.

6.0 Fusion

Autonomous systems are becoming an increasingly critical aspect of military operations. To meet this demand, the USAF is seeking ever more capable autonomy frameworks to enable interaction and task-based management of a multitude of systems. This framework should provide for increased connectivity to net-centric global information sources (GIS) and the ability to accomplish more complex, dynamic missions through goal-oriented interaction with autonomy based systems.

To realize such framework research is required to identify the issues related to developing a software-based framework (fusion) to foster the development of intuitive & effective human interaction with adaptive automation tools. By introducing cooperative planners, intelligent agents and machine learning into a comprehensive autonomy framework, this research has considered the following four fundamental software-based research questions:

1. How can a software system be instrumented to gather real-time user/machine interactions and system details for use in experimentation, machine learning, and software agents?

2. How can a software system 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 messages?

3. How can a software system record the state of each of its components at a rate near 30hz and make it user-accessible to enable discrete and continuous retrospection of the system in real-time?

4. How can a software system create a framework where every public element, regardless of its role as a model, user-interface element, etc., is customizable, extendable and override-able by any other software developer in the system?

Having a robust autonomy-based framework enables evaluation of cooperation and coordination among widely disparate platforms such as remotely piloted unmanned systems, autonomous unmanned systems, and adversarial components for ground, air, cyber, space, maritime, and submarine entities. Tying these interactions into an immersive user interface will help to better evaluate user behaviors and confidence in a low-risk environment.

Unification of user interactions, autonomous platforms, and intelligent aids provides a unique challenge. A common drive is to push towards more autonomy, where the user's involvement in the process is diminished. The intent is to provide a level of operator observability, transparency and directability that will support better human-autonomy coordination. This will allow evaluation of user comfort, trust, and confidence with autonomous components. A disciplined investigation of emergent human-autonomy coordination behavior will also serve to guide future requirements for user interface design and accessibility.

6.1 Fusion Framework

Fusion is a framework that enables natural human interaction with flexible and adaptive automation. It employs multiple components: intelligent agents, reasoning among disparate domain knowledge sources (Douglass, 2013); machine learning, providing monitoring services and intelligent aids to the operator (Vernacsics, 2013); cooperative planners (Kingston, 2009); and advanced simulation via an instrumented, goal-oriented operator interface (Miller, 2012). These empower scientific experimentation and technology advancement across multiple systems (see Figure 8). 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 API layer, (c) the module / messaging layer, and (d) the application layer (see Figure 9). The core framework layer provides foundational software classes and an application programming interface (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 and feedback, workspace management, asset management (vehicles, tracks, sensors, named areas of interest, etc.), global information services (GIS) data and earth mapping capability, and user interface elements. All software modules have 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 user interfaces, 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 test operator console. All code is written utilizing agile software development principles, namely the SOLID principles

(Single Responsibility, Open/Closed, Liskov Substitution, Interface Segregation, and Dependency Inversion) (Martin, 2012).

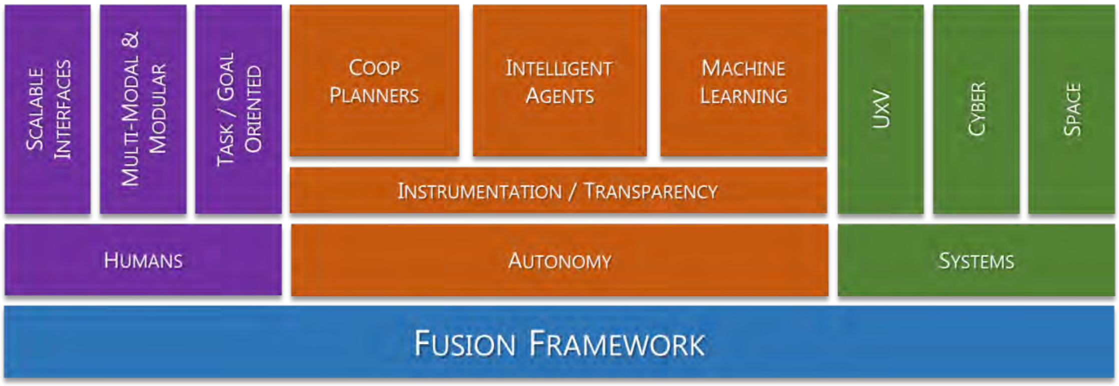


Figure 7. Fusion High Level Framework.

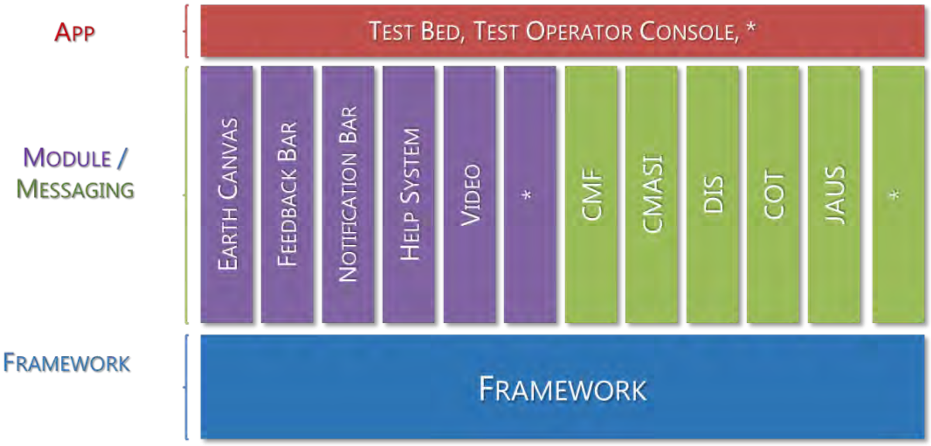


Figure 8. Fusion Layered Architecture.

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 10).

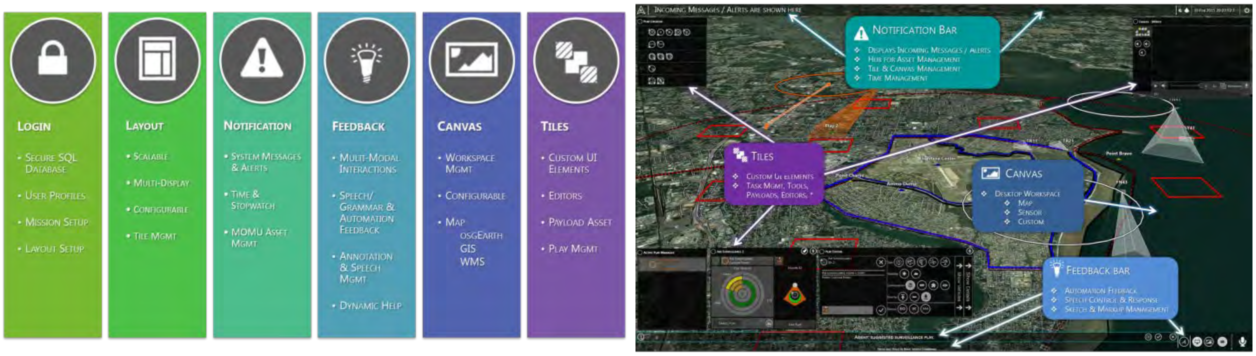


Figure 9. Fusion Visual Framework Components

6.2 Virtual Distributed Lab

The notion of a virtual distributed laboratory (VDL) connecting various DoD and contractor sites throughout the CONUS was paramount to foster a more cohesive and distributed development and research environment. Fusion has 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 Wright-Patterson AFB, and there are currently several offsite laboratory development teams (see Figure 11). Fusion is hosted on a secure web server and program access can be requested at https://www.vdl.afrl.af.mil/ (please contact the authors for further instructions). The software is developed on a standard Windows 8.1 PC platform in Microsoft Visual Studio and several third party developer libraries (See Figure 4). All distributed laboratory sites have similar hardware configurations and the software developers use a common set of software development and configuration management tools.

The Fusion software development team leverages SCRUM, an agile software development process. 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 researchers. In the coming year, the team will transition to a fully on-line software development cycle utilizing Git (a software configuration repository structure) and the secure Defense Research & Engineering Network (DREN). This process allows all offsite labs to keep up to date with the core Fusion team as well as keep their software well maintained. The concept of a virtual distributed laboratory has been successful due to Fusion providing a robust and flexible software architecture.

File Sharing
Software Repositories
Configuration Management
Software Requirements Management
ARL
SPAWAR
fusion
Software Development
Visual Studio
ReSharper
dotCover
Configuration Management
YouTrack
TeamCity

Figure 10. Fusion Virtual Laboratory Concept.

6.3 Fusion Keystone Program Application

The Fusion program formed to support two ASD R&E Autonomy Pilot Research Initiative (ARPI) projects. The purpose of this initiative was to foster research and push the envelope for autonomy-based researcg. One such project, Realizing Autonomy via Intelligent Adaptive Hybrid Control, is developing an “Intelligent Multi-UxV Planner with Adaptive Collaborative/Control Technologies” (IMPACT). This is a three year effort (FY14-FY16) with a focus on maximizing human-autonomy team agility. The first year of this effort focused on designing and implementing the user interface for higher level, goal-oriented plays (analogous to the sports ontology), which included asset management and integrating the various autonomous components. This “play” centric concept allows operators to focus on higher level goals for the vehicles, leveraging autonomous aids in accomplishing those goals, thus reducing the need for the user to direct and control vehicles manually.

The IMPACT instantiation of Fusion, as shown in Figure 12, employs a four screen layout: system tools, real time tactical situation awareness (SA) display, sandbox tactical SA display, and sensor management. The operator uses the sandbox display to perform all of their play calling and chat monitoring tasks. The other screens display tools to enhance the operator’s SA.

All of the goals of the Fusion framework were critical to the success of the first year of IMPACT. Fusion, autonomous agents, external simulations, a cooperative control planner (CCA), and machine learning algorithms were combined to form a comprehensive, richly interactive environment. This was enabled by Fusion’s flexible software architecture through a robust simulation environment, software extensibility, and interface instrumentation.

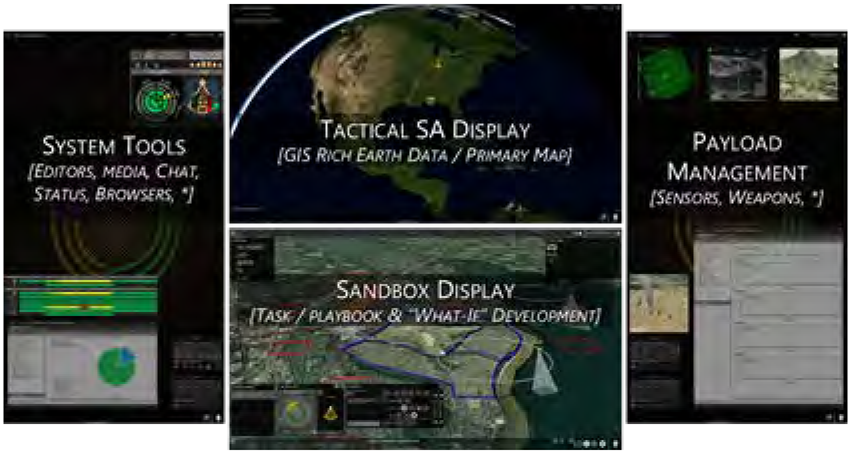

System Tools
[Editors, media, Chat, Status, Browsers, *]
Tactical SA Display
[GIS Rich Earth Data / Primary Map]
Payload Management
[Sensors, Weapons, *]
Sandbox Display
[Task / playbook & "What-If" Development]

Figure 11. IMPACT instantiation of Fusion.

6.4 Flexible Software Architecture

There are four primary research threads that Fusion is addressing to accomplish the goals of developing a framework for human interaction with flexible-adaptive automation across multiple UxSs: (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 at a rate near 30hz and makes it user-accessible to enable discrete and continuous retrospection of the system in real-time.

6.5 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., UDP, TCP/IP, ZeroMQ) and communicate using various messaging protocols (LMCP, JSON, DIS, and custom protocols). In some form, all the components are linked together in their communications modalities by use of a hub. 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 (discussed in detail later in this section) is distributed to the communications hub, and any component that wishes to consume the data can do so with a simple 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 Fusion team has worked closely with developers of other software components in order to ensure a seamless integration. The publish/subscribe architecture present on the communications hub makes for a natural assembly: all the associated data published by any software entity is available to any other entity that needs to leverage it, thus enabling great flexibility in the potential interactions between the entities, including Fusion and its operator(s). It also establishes the framework that will be needed for our near-future extension of Fusion to a MOMU (multiple operator/multiple unmanned system) interface.

With Fusion as an operator's interface, the communication with the components are more transparent and natural. For example, in the IMPACT work that Fusion has supported, a user can define high-level goals that Fusion dispatches to an intelligent agent. The agent allocates the vehicle or vehicles to be used for the achievement of the goals and submits the realized requirement to the vehicle planner. The planner reports the plan back to Fusion, which is then shown as a candidate solution. When accepted, the plan is delivered to the vehicle simulation for execution, while another component monitors the plan execution to alert the operator in case of issues. Each of these pieces communicate using a different set of message protocols. To the operator, it appears as if all these components are part of Fusion: the operator defines a goal and approves a plan, and it appears to the operator that the deliberative activity occurs exclusively within Fusion. In this way, Fusion enhanced the operator's teaming with the autonomous system components while ensuring that the communication and feedback are transparent to the operator. Since the operator is able to define high-level goals for the IMPACT project, the system enables a realization of an enhancement in the dialog between the operator and the various software entities that provide access and control of autonomous vehicles. This enhancement was exercised in the first IMPACT evaluation, which had positive feedback from all the subjects. The flexibility and extensibility of the communications framework has provided a baseline from which the development team will further extend and enhance the human-machine teaming, creating a more immersive and flexible interactive environment.

6.6 Software Extensibility

Fusion is being used in several different projects, all of which share the goal of improving operator interactions with highly autonomous systems (with future extensions to MOMU), but have vastly different human machine interface (HMI) designs and algorithms. Due to this, Fusion was built with the goal of extensibility.

Fusion's infrastructure allows developers to override aspects of the HMI easily. Fusion adopted a layered architecture in which the framework contains the building blocks for HMI tools and services. Developers 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 user can choose which modules are loaded, and therefore affect how the Fusion user interface looks and reacts.

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 many different places in the user interface. In 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 places. Most projects have their own vehicle symbol design and it require a lot of work to replace that symbol everywhere for every project. Fusion was built with this in mind, therefore, there is a default simplistic vehicle symbol included in the framework. Every project has the ability to design and implement their own vehicle symbol. With a single line of code in the project-specific vehicle symbol code, all vehicle symbols in the entire Fusion test bed can be replaced. An additional feature is that developers do not even need to consider this during implementation: the framework handles it at run time.

Extensibility saves a great amount of development time and allows designers to try out multiple solutions. A user interface can be designed and implemented in multiple ways and, depending on which modules the user loads, a specific design is realized. This facilitates experimentation to determine the best design.

6.7 Interface Instrumentation

Data collection, agents, and machine learning all require capturing of data, which must be stored or packaged up and sent across the network. User interface interactions is one of these critical data sources. This capability was built into the Fusion framework. It is fairly non-invasive to the developers and provides a host of information, both after the fact and real time. Every user interaction, such as button clicks, typing, and mouse clicks are recorded and saved to a file.

All instrumented data is also packaged up and sent through the network to any agents, machine learning algorithms, cognitive modeling services, or other automated services that are subscribing to the data source. Instrumentation of all operator interactions is critical for human-autonomy teaming. There are many uses for this feature of Fusion. Agents use it to better understand user behavior and take or recommend actions. Machine learning employs instrumentation data to learn how individual users perform, and potentially recommend an interface change either after the fact or in real time (e.g., if a user uses certain buttons more often, the buttons can be reorganized to better suit their use). The data can inform cognitive modeling services, improving researchers' understanding of how the operators are performing.

During evaluations, all instrumentation data is recorded into a comma delimited log file. The experimenter can go back after the fact to analyze performance data. They can determine reaction times and accuracies based on the times of various user actions saved in the file. This was utilized in the IMPACT year one evaluation. For example, a chat module was employed to request tasks from the user. The chat requests and responses were instrumented, as were the user reactions. The experimenter leveraged this data to analyze how quickly and effectively the task was performed. The experimenter also noted any extra steps the operator performed, the sequence of steps taken, and the modality of their actions. All of this data is being used to analyze and improve the user interface as well as any automated services.

6.8 Human-Autonomy Dialog through Retrospection

All of the instrumentation data can be used for retrospection. Since all the data is stored, it is natural to allow 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 rewind the scenario introduced the concept of a sandbox. The sandbox is an area of the user interface where the operator can invoke actions that aren't instantly carried out by the UxSs. The sandbox allows the user to evaluate autonomy-proposed actions and tweak various parameters before committing to them. Other displays within Fusion still depict current vehicle activities in real time, so the operator maintains effective SA. This can give 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.

6.9 Summary

Fusion fills a needed role in human-autonomy teaming. Since humans will remain a critical part of autonomous systems development and deployment for the foreseeable future, a clear representation and extension of an operator's intent is required. Fusion was developed to support this, and it continues to expand within the human interface role as it pertains to the Air Force Research Laboratory's goals. It is well positioned to aid in achieving many goals related to autonomy defined by AFRL, the Air Force, and the Department of Defense. Fusion directly addresses two of the four goals established by the AFRL Autonomy Science and Technology Strategy (AFRL, 2013): delivering flexible autonomy systems with highly effective human-machine teaming and creating actively coordinated teams of multiple machines to achieve mission goals. Fusion also participates in addressing two challenges from the 2010 Technology Horizons (USAF Chief Scientist, 2010) for the Air Force: highly autonomous decision-making systems and fractionated, composable, survivable, autonomous systems. Finally, the defense science board (DoD Defense Science Board, 2012) identified perception, planning, learning, human-robot interaction, natural language understanding and multi-agent coordination as key areas that would benefit from improved autonomy. Fusion has encompassed and is continuing work in five of these domains (except perception). Fusion is thus supporting and driving the emerging technologies with which the goals and challenges across all levels of the AFRL hierarchy can be satisfied.

7.0 Conclusions

Unmanned aerial systems (UAS) are becoming an increasingly critical aspect of military operations. To meet this demand, the USAF is seeking ever more capable UASs, to include the ability of a single operator to simultaneously control multiple platforms, increased connectivity to net-centric information sources, and the ability to accomplish more complex, dynamic missions. These capabilities include:

1) close collaboration with manned assets

2) find, fix, track, and target difficult targets in complex urban and difficult terrains

3) destroy or neutralize difficult targets

4) precisely deliver select effects to maintain controllable collateral damage

5) persistence in multiple areas of interest

To fulfill these missions, the USAF is exploring multi-vehicle UAS concepts to carry out tactical intelligence, surveillance, reconnaissance and combat missions. In many of these concepts there is an emphasis on managing UAS systems and conducting missions with minimal crew. UASs have evolved from being primarily remotely controlled systems to being pre-programmed or semi-autonomous, changing the role of the crew from flying to supervising. This change has opened the door to increasing the vehicle to operator ratio. While progress has been made in developing more capability with multi-vehicle systems (e.g., more simultaneous vehicle orbits managed from one control station), further research is needed to increase mission effectiveness on per vehicle and per operator basis. To increase mission effectiveness, crew performance and capability enhancements are needed. Technology development and advanced designs are required to facilitate more timely and effective operator situation assessment and decision-making.

8.0 REFERENCES

Cavada, R., Cimatti, A., Jochim, C., Keighren, G., Olivetti, E., Pistore, M., Roveri, M., & Tchaltsev, A., (2010). NuSMV 2.5 User Manual, FBK-IRST: Fondazione Bruno Kessler / Istituto Ricerca Scientifica e Tecnologica, Trento, Italy, Retrieved from: http://nusmv.fbk.eu/NuSMV/userman/v25/nusmv.pdf, 2010.

Cimatti, A., Clarke, E., Giunchiglia, E., Giunchiglia, F., Pistore, M., Roveri, M., Sebastiani, R., & Tacchella, A., (2002). "NuSMV 2: An Opensource Tool for Symbolic Model Checking," Computer Aided Verification (pp. 359-364). Springer Berlin Heidelberg.

Doherty, P., Kvarnstrom, J., & Heintz, F., (2009). A Temporal Logic-Based Planning and Execution Monitoring Framework for Unmanned Aircraft Systems. *Autonomous Agent Multi-Agent Systems*, *19*, 332–377.

Dwyer, M., Avrunin, G., & Corbett, J., (1999). Patterns in Property Specifications for Finite-State Verification. *Proceedings of the 21st International Conference on Software Engineering*, (pp. 411–420). ACM.

Feitshans, G., Rowe, A., Davis, J., Holland, M., & Berger, L., (2008). "Vigilant Spirit Control Station (VSCS) "The Face of COUNTER", Proceedings of the AIAA Guidance, Navigation, and Control (GNC) Conference.

Humphrey, L., (2012). "Model Checking UAV Mission Plans," Proceedings of the AIAA Modeling and Simulation Technologies (MST) Conference.

Humphrey, L., & Patzek, M. (2013). Model Checking Human-Automation UAV Mission Plans. Proceedings of the AIAA Guidance, Navigation, and Control (GNC) Conference.

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.

Konrad, S. & Cheng, B., (2005a). "Facilitating the Construction of Specification Pattern-Based Properties," 13th IEEE Int'l Requirements Engineering Conference (RE'05).

Konrad, S. & Cheng, B., (2005b). "Real-Time Specification Patterns," Proceedings of the 27th Conference on Software Engineering (ICSE'05).

Kress-Gazit, H., Fainekos, G., & Pappas, G., (2008). "Translating Structured English to Robot Controllers," *Advanced Robotics*, *22*, 1343–1359.

Rothwell, C., Eggert, A., Patzek, M., Bearden, G., Calhoun, G., & Humphrey, L. (2013). Human-Computer Interface Concepts for Verifiable Mission Specification, Planning, and Management. AIAA Infotech@Aerospace, AIAA 2013-4804.

Rowe, A., Liggett, K., & Davis, J., (2009). Vigilant Spirit Control Station: A Research Testbed for Multi-UAS Supervisory Control Interface Technology. *Proceedings of the 15th International Symposium of Aviation Psychology*.

Stanard, T., Bearden, G., & Rothwell, C. (2013, September). A Cognitive Task Analysis to Elicit Preliminary Requirements for an Automated UAV Verification & Planning System. In *Proceedings of the Human Factors and Ergonomics Society Annual Meeting* (Vol. 57, No. 1, pp. 210-214). SAGE Publications.

Billings, C. (1991). Human-centered aircraft automation: A concept and guidelines. NASA TR-103885.

Carretta, T., M. Patzek, L. Warfield, S. Spriggs, A. Rowe, A. Gonzalez-Garcia, and K. Liggett.: Target Acquisition Involving Multiple Unmanned Air Vehicles: Interfaces for Small Unmanned Systems (ISUS) Program. In: AFRL-RH-WP-TR-2009-0021. Wright-Patterson Air Force Base, OH: Air Force Research Laboratory. (2009)

Casbeer, D., Meier, K., and Cao, Y. (2013). Estimating the State of an Intruder with a UAV and Unattended Ground Sensors, AIAA Infotech@Aerospace (I@A) Conference, No. AIAA 2013-4581, Boston, MA.

Chandler, P., M. Patzek, M. Pachter, C. Rothwell, S. Naderer, K. Kalyanam.: Integrated Human Behavior Modeling and Stochastic Control (IHBMSC). In: AFRL-RQ-WP-TR-2014-0191. Wright-Patterson Air Force Base, OH: Air Force Research Laboratory. (2014)

Chen, H., Kalyanam, K., Zhang, W., and Casbeer, D (2014). Continuous-time intruder isolation using unattended ground sensors on a general graph. 2014 American Control Conf., Portland, OR, pp 5270-5275.

Davies, D., and Parasuraman, R. (1982). The psychology of vigilance. London: Academic Press.

Endo, Y., MacKenzie, D., & Arkin, R. (2004). Usability evaluation of high level user assistance for robot mission specification. IEEE Transactions on Systems, Man, and Cybernetics-Part C: Applications and Reviews, 34(2), 168- 180.

Endsley, M. (1988). Situation awareness global assessment technique (SAGAT). Proceedings of the National Aerospace Engineering Conference, May, 1988.

Endsley, M. (1995). Measurement of situation awareness in dynamic systems, *Human Factors, 37*, 65 - 84.

Endsley, M., and Kaber, D. (1999). Level of automation effects on performance, situation awareness and workload in a dynamic control task. *Ergonomics, 42*, 462-492.

Feitshans, G., Rowe, A., Davis, J., Holland, M., & Berger, L. (2008). Vigilant Spirit Control Station (VSCS) “The Face of COUNTER”. AIAA Guidance, Navigation, and Control Conference and Exhibit, Honolulu, HI. doi: 10.2514/6.2008-6309

Gross, D., S. J. Rasmussen, P. Chandler, and G. Feitshans. “Cooperative Operations in UrbaN TERrain (COUNTER)”. In: Proceedings of SPIE, volume 6249 of Defense Transformation and Network-Centric Systems. (2006)

Holsapple, R., P. Chandler, J. Baker, A. Girard, and M. Pachter.: Autonomous Decision Making with Uncertainty for an Urban Intelligence, Surveillance and Reconnaissance (ISR) Scenario. In: Proceedings of the AIAA Guidance, Navigation and Control Conference. (2008)

Harel, D., (1987). Statecharts: A visual formalism for complex systems. Science of Computer Programming, 8, 231-274.

Hart, S.G. and Staveland, L.E. (1988). Development of NASA TLX (Task Load Index): Result of Empirical and Theoretical Research. In P. A. Hancock & N. Meshkati (Eds.), Human Mental Workload (pp. 239-250). Amsterdam: North Holland Press.

Pachter, M., P. Chandler, and S. Darbha.: Optimal MAV Operations in an Uncertain Environment. In: International Journal of Robust and Nonlinear Control, 00:1–6, (2006)

Parasuraman, R. (1987). Human-computer monitoring. *Human Factors, 29,* 695-706.

Parasuraman, R., Molloy, R., and Singh, I. (1993). Performance consequences of automation-induced “complacency.” *The International Journal of Aviation Psychology, 3,*1-23.

Patzek, M., Rothwell, C., Bearden, G., Ausdenmoore, B., and Rowe, A (2013). Supervisory control state diagrams to depict autonomous activity. In Proceedings of the 2013 International Symposium on Aviation Psychology, Dayton, OH.

Patzek, M., Zimmer, D., Feitshans, G., Draper, M., Hughes, T., & Flach, J. (2009). Multi-UAV supervisory control interface technology. In Proceedings of the 2009 International Symposium on Aviation Psychology, Dayton, OH.

Sarter, N., and Woods, D. (1995). “How in the world did I ever get into that mode?” Mode error and awareness in supervisory control. In R. Gilson, D. Garland, & J. Koonce (eds.) *Situation Awareness in Complex Environments*, Daytona Beach, FL: Embry-Riddle Aeronautical University Press, pp. 111-124.

Sarter, N., Woods, D., and Billings, C. (1997). Automation Surprises. In G. Salvendy (ed.) *Handbook of Human Factors and Ergonomics* (2nd ed.), New York, NY: Wiley, pp. 1926-1943.

Skitka, L., Mosier, K., and Burdick, M. (1999). Does automation bias decision-making? *International Journal of Human-Computer Studies, 51*, 991-1006.

Taylor, R.M. (1990). Situational Awareness Rating Technique (SART): The development of a toll for aircrew systems design. In: AGARD Conference Proceedings No 478, Situational Awareness in Aerospace Operations. Aerospace Medical Panel Symposium, Copenhagen, 2nd - 6th October 1989.

United States Air Force Chief Scientist (2010). “Technology Horizons: A Vision for Air Force Science & Technology during 2010-2030.” Technical report AF/ST-TR-10-01-PR, AF/ST, Office of the Chief Scientist of the U.S. Air Force. Retrieved from http://www.af.mil/shared/media/document/AFD-101130-062.pdf

Woods, D., and Sarter, N. (2000). Learning from automation surprises and going sour accidents. In N. Sarter and R. Amalberti, (eds.), *Cognitive Engineering in the Aviation Doman*, Hillsdale, NJ: Lawrence Erlbaum, pp. 327-354