Man-Machine Interoperation in Training for Large Force Exercise Air Missions

Patrick L. Craven, Kevin B. Oden,
Kevin J. Landers, David J. Macannuco

Lockheed Martin Corporation
Orlando, Florida
patrick.craven@lmco.com, kevin.oden@lmco.com,
kevin.j.landers@lmco.com, david.j.macannuco@lmco.com

Ankit J. Shah and Julie A. Shah

Massachusetts Institute of Technology
Cambridge, Massachusetts
ajshah@mit.edu, julie_a_shah@csail.mit.edu

ABSTRACT

The United States Air Force strives to maximize human abilities in highly complex operational environments, and artificial intelligence (AI) affords opportunities to transform voluminous data into meaningful information to support human decision making. A Mission Analysis and Review System (MARS) was developed to explore how AI can automate current mission debrief processes and to visualize that information in a mission-specific context. The current effort explored the development of AI to assist Air Force mission commanders in evaluating the mission performance of a Large Force Exercise (LFE), which affords pilots the chance to hone their abilities to execute their individual role within a mission that may include dozens of aircraft.

In an earlier but related effort, the research team developed AI to automatically label mission phases of a two-ship strike formation using entity state data of aircraft flown by human pilots in simulation. In the current effort, an LFE with 18 friendly aircraft was simulated using the Joint Semi-Automated Forces (JSAF) simulation engine. Models were created to score both individual aircraft behavior as well as overall mission objective success. Templates were used to determine if acceptable levels of key mission objectives are being estimated and evaluated. By enumerating the propositions included in the three temporal behaviors in the classification model, the behaviors the model deems necessary for evaluating the execution as acceptable were interpreted. Results showed that mission phases and their objectives could be correctly classified with an accuracy of .92 to .96 using a technique where mission objectives were encoded in a linear temporal logic (LTL) format.

The findings suggest that AI can be used to make meaning of raw data for use by mission commanders to support LFE planning and debrief. The effort described is the first known application of AI to automatically score mission performance for an LFE.

ABOUT THE AUTHORS

Dr. Patrick L Craven is a certified human factors professional with Lockheed Martin Rotary and Mission System where he focuses on developing advanced human-centered technologies, predominately for us in training systems. He has led efforts that developed human performance augmentation strategies, increased system usability, evaluated system and operator functionality, and enhanced interface design. He has experience designing and evaluating human-technology interactions including neurophysiological-based measures of cognition, human-autonomy interaction and teaming, command and control, handheld, aircraft, and intelligence analysis applications.

Mr. Ankit Shah is an advanced PhD graduate student at MIT. His current research focuses on inferring formal logic specifications for complex tasks from demonstrations. His broader interests also include planning and learning within domains with hierarchical state descriptions. He is also interested in applying these algorithmic techniques towards enhancing the capabilities of human-robot teams. He has previously received his SM (2016) from the Department of Aeronautics and Astronautics at MIT, and his B.Tech. (2013) from the Indian Institute of Technology, Bombay (IIT-B).

Dr. Kevin Oden Dr. Oden is the Principal Investigator of the Human Systems and Training (HST) Lab for the Advanced Simulation Center. In this role, he leads far-reaching R&D efforts that aim to accelerate the rate at which individuals and teams achieve expertise. Recent efforts have focused on the use of advanced sensing technologies to create objective measures of cognitive skills and emotional intelligence that are not directly observable. He partners with universities (MIT, Yale, and Drexel), small businesses and commercial companies to create new capabilities that improve and extend human performance. Dr. Oden holds a Ph.D. in Applied Experimental and Human Factors Psychology from the University of Central Florida, where he also earned a M.S. in Modeling and Simulation. A graduate from the University of Florida, Dr. Oden was also a Graduate Fellow Researcher at the Army Research Institute for Behavioral and Social Sciences with sponsorship from the Consortium of Universities located in Washington D.C.

Dr. Julie Shah is an Associate Professor in the Department of Aeronautics and Astronautics at MIT and leads the Interactive Robotics Group of the Computer Science and Artificial Intelligence Laboratory. Shah received her SB (2004) and SM (2006) from the Department of Aeronautics and Astronautics at MIT, and her PhD (2010) in Autonomous Systems from MIT. Before joining the faculty, she worked at Boeing Research and Technology on robotics applications for aerospace manufacturing. She has developed innovative methods for enabling fluid human-robot teamwork in time-critical, safety-critical domains, ranging from manufacturing to surgery to space exploration. Her group draws on expertise in artificial intelligence, human factors, and systems engineering to develop interactive robots that emulate the qualities of effective human team members to improve the efficiency of human-robot teamwork. In 2014, Shah was recognized with an NSF CAREER award for her work on “Human-aware Autonomy for Team-oriented Environments," and by the MIT Technology Review TR35 list as one of the world’s top innovators under the age of 35. Her work on industrial human-robot collaboration was also recognized by the Technology Review as one of the 10 Breakthrough Technologies of 2013, and she has received international recognition in the form of best paper awards and nominations from the International Conference on Automated Planning and Scheduling, the American Institute of Aeronautics and Astronautics, the IEEE/ACM International Conference on Human-Robot Interaction, the International Symposium on Robotics, and the Human Factors and Ergonomics Society.

Mr. Kevin Landers is a Senior Software Engineer for the Advanced Simulation Center at Lockheed Martin and has been developing advanced technology solutions for five years. He is the lead software engineer for the LM-MIT program where he has implemented scenario review and evaluation capabilities and has deep experience in implementing mixed-reality interfaces for DoD applications. He joined the Human Systems and Training (HST) Lab in 2016 and has experience in the design and implementation of data collection and visualization systems. Mr. Landers is graduate of The Ohio State University where he earned a BS in Computer Science and Engineering.

Mr. David Macannuco is the Lead Engineering Manager for the Advanced Simulation Center at Lockheed Martin where he leads a team of 50 engineers and scientists developing advanced modeling and simulation technologies. Mr. Macannuco has over 20 years of experience developing and leading research and development efforts in modeling and simulation. His current research interests include the development of distributed architectures for simulation, the application of machine learning techniques in training and the development of augmented reality systems. Mr. Macannuco earned a BSEE from the University of Rochester and the an MSEE from Boston University.

Man-Machine Interoperation in Training for Large Force Exercise Air Missions

Patrick L. Craven, Kevin B. Oden,
Kevin J. Landers, David J. Macannuco

Lockheed Martin Corporation
Orlando, Florida
patrick.craven@lmco.com, kevin.oden@lmco.com,
kevin.j.landers@lmco.com, david.j.macannuco@lmco.com

Ankit J. Shah and Julie A. Shah

Massachusetts Institute of Technology
Cambridge, Massachusetts
ajshah@mit.edu, julie_a_shah@csail.mit.edu

INTRODUCTION

Autonomous systems and robots are heavily emphasized in technology roadmaps for the United States Air Force (Dahm, 2010). In November 2015, the Deputy Secretary of Defense spoke about how the Defense Science Board Summer Study on autonomy concluded that we are at an inflection point for artificial intelligence (AI) and autonomy (Pellerin, 2015). There is recognition that currently developed technological systems can operate more intelligently and more independently than in the past, and their role in defense will soon become more prominent. As a result, these autonomous systems will disrupt current practices that have been honed for human-dominated actions. Even when technology ultimately passes the tipping point of transitioning from automation into autonomy, these systems are unlikely to replace human decision-making (Bradshaw, Hoffman, Woods, & Johnson, 2013; Murphy & Shields, 2012) because there are fundamental differences in the relative strengths of human versus machine cognition that make pairing them more powerful than using just one. As Bradshaw et al. describe, “there’s a need for the kinds of breakthroughs in human machine teamwork that would enable autonomous systems not merely to do things for people, but also to work together with people and other systems.” Thus, it is anticipated that smart technological systems will serve in an advisory capacity in collaboration with humans who have the final authority to act.

The Observe-Orient-Decide-Act (OODA) loop was formulated by Colonel John Boyd and describes the process by which fighter pilots engage threats (Feloni & Pelisson, 2017). This framework has been extended to many more activities, including the Plan-Brief-Execute-Debrief (PBED) “Win Cycle” (see Figure 1). Pilots and practitioners of the win cycle are encouraged to find ways to get inside the OODA loop of an opponent through better information faster execution, etc. However, as the volume of available information relevant to a decision has increased, the human decision-maker has become more encumbered and is challenged to make decisions efficiently. Consequently, there is a desire to augment human decision-making though AI tools. The focus of the current work is on AI tools in the debrief process and tools that allow a commander to more rapidly process what occurred during mission execution and which lessons learned should be identified.

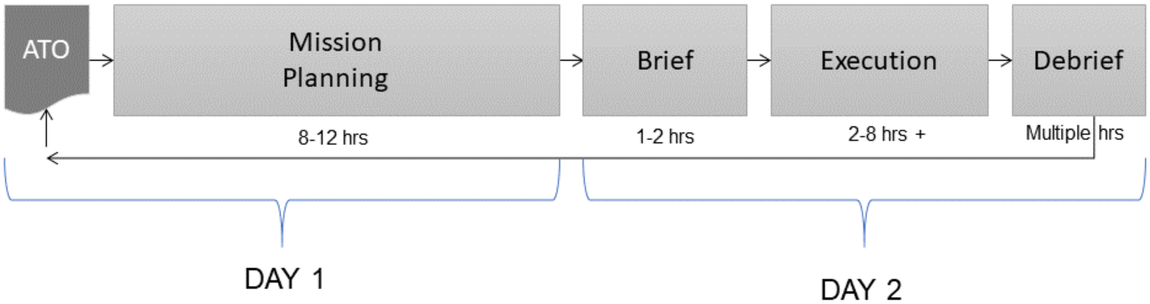

Figure 1. The Mission "Win Cycle"

In the Win Cycle the Air Tasking Order (ATO) is used to generate a mission plan, which constitutes the largest portion of time in the cycle. The plan is initially briefed, then executed, and then the flight elements participate in a debrief. For example, Nellis Air Force Base hosts RED FLAG, a realistic combat training exercise with United States and allied forces. It was established in 1975, and provides an opportunity for a large group of mixed "blue" aircraft (attack, fighter, bomber, reconnaissance, electronic warfare, etc.) to fly missions against "red" air and ground threats (Nellis Air Force Base, 2012). At the conclusion of a training mission, the mission commander guides the participants through the mission to create a ground truth from which lessons can be learned. A mission commander must accurately process the activities of numerous manned (and unmanned) aerial platforms and link their mission execution to either accomplishing or failing to meet the mission objectives. At Red Flag the debrief includes the following sub-tasks:

- **Mission Complete:** Pilot returns safely, followed by maintenance debrief & intel collection.
- **Mass Debrief Preparation (aka, Debrief Pre-mass):** Pilots recreate truth data of specific actions during mission based on recorded data.
- **Mass Debrief:** All flight elements combine truth data from each flight/formation to create a mission overview. In the Mass Debrief the mission commander also relays debrief focal points (DFPs).
- **Element Debrief:** Flight lead identifies causal factors that relate to DFPs and the contributing roles of each pilot in the formation. Also, suggestions for instructional fixes are provided to the formation members.

The goal of this research was to investigate how AI-enhanced technological tools could support the debrief process for large force exercise (LFE) training missions. Specifically, advanced classification methods can be used to automatically code the pilot's actions in order to better match execution to objectives. Machine learning techniques can be used to model the process by which mission commander's review mission execution and relate that to the mission objectives. This model can then be used to provide tailored feedback to pilots (human- or autonomous-controlled) as part of the Mass Debrief to assist in ensuring that future missions are executed in accordance with objectives.

The first phase of this research effort focused on developing the machine learning system to classify individual aircraft phase (level II) data based only on the TSPI (level I) data of those aircraft. We addressed the goal of the second phase by training a machine learning (ML) artificial intelligence (AI) system to predict a semantic label for the entire mission (level III) based on the observed level I and II data streams (see Table 1 for detailed descriptions of the levels of description). Additionally, the AI system calculated scores for achieving mission success or failure and this output was embedded in a debrief tool that could allow a mission commander to initiate an auto-scoring process that would have the AI system analyze raw mission execution data and display a properly segmented and scored mission for the mission commander's verification.

Table 1. Levels of Mission Description

Level of Data	Level Description
Level I	This first level of description uses time and space position information (TSPI) of an aircraft, also including heading, speed, roll, yaw, pitch, etc. This is the lowest level of description and includes discrete values for telemetry values of a single aircraft.
Level II	This level of description offers semantic labels that describe in which phase of the mission a single aircraft is currently. The Level II phase descriptions used were: • Ingress • Orbit • Threat Avoidance • Threat Engagement • Evasive Actions • On Target • Egress • Shot Down
Level III	This level of description provides a semantic label for the entire mission based on the Level I and Level II data of each aircraft in that mission. For the Offensive Counter-Air (OCA) mission, we defined four different sequential phases that were: • (Phase 1) **Escort Push** • (Phase 2) **Striker Push** • (Phase 3) **Time on Target (TOT)** • (Phase 4) **Egress**

MATERIALS AND METHODS

As there was no known mission execution data set that the research team had permission to use, a new data set was created for a Large Force Exercise. The method for creating the scenario design, implementing the scenario, running multiple instances of the scenario, and capturing and pre-processing the data are described in the following section.

Task Scenario Design

LFE scenarios were created with an Offensive Counter-Air (OCA) mission using 18 total friendly aircraft. Table 2 shows the aircraft and armaments for the three elements in the 18-aircraft mission. The general mission timeline progressed from Escort Push to Egress (see Level III in Table 1) from Push to Safe. During the Escort Push Phase, the F-15s enter the threat airspace and begin targeting aerial threats. Next, the F-22s and SEAD aircraft enter ahead of the Strike aircraft and continue to establish air dominance be removing threat aircraft and known and popup ground-based threats. The Escort and SEAD aircraft establish their combat air patrol (CAP) patterns to protect the Strike package which begins ingress during the Striker Push Phase. After delivering their munitions during the Time on Target (TOT) Phase, the Strike element calls the "Millertime" contract to alert the other aircraft to begin egress. During the Egress Phase the Escort and SEAD aircraft maintain a defensive pattern and usher the Strike element to safety.

Table 2. Mission Details

Element / Role	Aircraft	Armaments
Escort	BUICK (BK) F-22 x4	AMRAAM x6 Sidewinder x2 M50 480 rounds
	CHEVY (CY) F-15C x4	Sidewinder x1 AMRAAM x7 M50 940 rounds
SEAD	HARLY (HY) F-16CJ x4 NITRO (NO) F-16CJ x4	HARM x2 AMRAAM x3 Sidewinder x1 M50 540 rounds
Strike	THUD (TD) F-16 x1 HUN (HN) F-16 x1	GBU31B x2 AMRAAM x3 Sidewinder x1 M50 540 rounds

This description of the scenario represents an ideal execution, but not all missions will be executed flawlessly. For Mission Analysis and Review (MARS) tools to be able to provide mission commanders with relevant feedback on achieving mission and tactical objectives, the training data set needs to include execution that deviate from flawless performance. Thus, in consultation with a mission commander with Red Flag experience, several scenario variations were created.

The scenario variations and the expected impact to how a mission commander would score the mission are listed in Table 3. Each of the variations were given a letter ID so that our mission commander wouldn't be biased in his interpretation. Scenarios were then developed in our constructive environment (detailed below) and executed twice in that environment and appropriately marked. Both executions of the same variant were shown successively to the mission commander in the following order: J, A, K, B, N, F, H, G, C, I, L, M. During the second execution of running the A scenario in our constructive environment the B scenario was inadvertently executed instead, so our mission commander ended up scoring three executions of the B variant and one of A. All the other variants were run and scored twice. It was determined that the inclusion of one more B and one less A would likely have a negligible influence in biasing the learning model, so all the data in the training set were used to train the model.

Table 3. Scenario Variations

ID	Description and Scenario Design	Expected Violation(s)
A	SEAD and/or Escort don't hear "Millertime." Extended Escort and SEAD CAP time so they egress late	Contract violation
B	Fumble is called, but no ack from escort and it leaves the area (confuses it with Millertime). AI loops back and drops bombs round 2, but Escort and SEAD egress at normal time	Contract violation
C	Base Scenario with no variations.	None
F	Loss of > 50% of escort, abort called and executed. Improve red numbers/competence/strategic position or decrease blue competence; Pause scenario, force egress on everyone, resume	MO violation Contract fulfilled
G	Loss of > 50% of escort, abort not called. Improve red numbers/competence/strategic position or decrease blue competence; no abort	MO violation Contract violation
H	Delay in clearing air threats, spin called x 3, Strike not delivered in TOT. Add red air/ground, strikers orbit ~7min while Escort/SEAD clear the area	Contract fulfilled MO violation for TOT
I	Spin called but air threats still present in strike zone and they fire at Strike. Add red air/ground, strikers orbit and then proceed early getting caught in cross-fire	Escort TO violation
J	Spin is called (once), strike is delivered TOT. Add a few red air/ground, strikers orbit (~3min) but still deliver in TOT	None
K	Lean called and followed (could be part of one of the other scenarios, maybe the fumble). Move red CAP or SA to inside strike route, strike route leans out	None
L	Lean contract called for strike, but not followed. Move red CAP routes to inside strike route, strike doesn't avoid	Contract violation
M	Not solid ID - non-military target shot down. Add an unknown non-military jet into the strike route that gets shot down by Escort/SEAD	MO violation of 100% valid ID
N	Strike delivered to incorrect target. Change Striker(s) to target something on the ground other than the target	MO Violation Strike TOT Violation

Constructive Environment: Joint Semi-Automated Forces (JSAF)

The LFE mission described earlier was used as the outline for the scenarios that were created in JSAF. Overlays were created for starting positions, ingress and egress paths, combat air patrol (CAP) routes, targets and transition lines. For each friendly aircraft in the scenario, the correct model was selected and placed in the scenario at its starting position. Tasks for each aircraft were decided by the role of the aircraft. For Escort aircraft, a sequence of ingress followed by CAP, followed by egress was defined where transitions were decided by control measures (i.e., a line overlay indicating when to transition from ingress to CAP) or by duration of time. Similar sequences were defined for SEAD (Ingress->SEAD->CAP->Egress) and Strikers (Ingress->Attack Target->Egress). Depending on the scenario lean, fumble, spin, and others were scripted as well.

Enemy aircraft were added in a similar fashion. Most of them stay in CAP for the duration of the scenario, but some were scripted to join the active regions of the scenario later than others, depending on what the scenario prescribed. In the scenarios, outcomes would change simply by adding additional enemy aircraft at specific times to elicit different behaviors from friendly forces.

JSAF was setup in conjunction with a data reporting and replay tool called Caber. Caber provided a way to record each mission execution in JSAF without degrading performance. Caber provides the ability to play back a mission in JSAF and to report out the time and space information (TSPI) data needed for replay in the MARS Tools as well as processing by the AI. We executed each of the 12 scenarios twice which resulted in 24 missions that could be scored by our mission commander.

Mission Review and Scoring

The mission execution phase labels were manually identified and logged by an F-22 pilot who had experience serving as a mission commander multiple times at Red Flag large force training exercises. This expert reviewed each mission execution using the Mission Analysis and Review (MARS) tool in order to segment the mission into four separate time blocks of Level III labels (see Table 1).

The playback controller and bookmarking interface combine to give users a toolset to annotate each scenario. Individual aircraft phase labels and callsigns show up for each unit during playback. Yellow highlights help designate phase changes; red highlights are used for phase transitions where the end transition is "DOWN" (meaning shot down). Contract calls and time on target (TOT) timespan both have times associated with them and are displayed above the time scrubber in the timeline controller. All these elements combine to provide a good visual breakdown of the scenario. During the annotation portion of this project, pilots can review the scenario with the bookmarks already displayed. As they go through the scenario using the playback positions and phases as guides, they can add bookmarks for phase transitions and any tactical objective or contract failures. Phases show up on the timeline as spans (see the Timeline Controller widget in Figure 2), while contract or tactical objective failures associated with a time are displayed as tick marks. Double-clicking on these spans forces the current time in playback to jump to the associated time. The overall phase of the mission is displayed at the top of the web interface along with the last contract that was called.

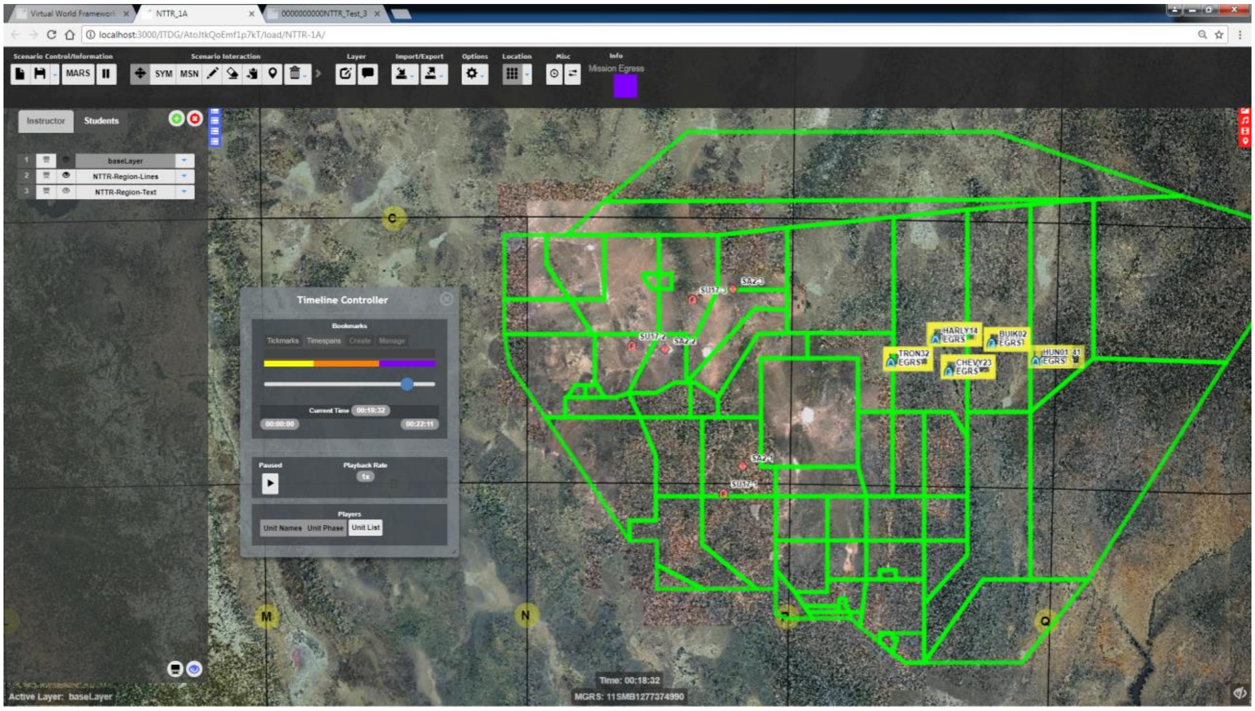

Figure 2. Mission Analysis and Reviews System (MARS) for scoring and reviewing missions

Additionally, the mission commander completed a questionnaire for each mission execution where he scored success or failure of overall mission objectives, success or failure of flight element tactical objectives, and marked whether contracts were called and followed. Questions were asked for each of the four phases of the mission and for each of the flight elements (Escort, SEAD, Strike).

RESULTS

The data from the scenario executions were organized for easier ingestion by the ML training algorithms. The survey data was converted into a text file and verified for completeness. Additionally, Level II data was automatically coded using a logical algorithm that used the aircraft's TSPI data and pre-defined mission parameters to code the aircraft behavior. This method offered similar output to what the ML model produced in Phase 1 of this research, but because we were using JSAF instead of pilot-flown missions we were unable to use the earlier ML model to encode Level II data. Nevertheless, the logical algorithms we created produced reasonable accuracy and were verified by inspection by the research team as well as by our mission commander while he coded the Level III data.

For the AI system to be able to learn from the training data, the mission objectives needed to be constructed as a set of simple Booleans and operationalized based on mission execution data. The expressions that were used to operationalize these objectives are listed in Table 4.

Table 4. How the Mission Objectives Were Operationalized

Mission Objective	Objective Operationalized
MO1 Gain and maintain air superiority	**MO1.1** Enemy attrition 50%, 75%, 100% **MO1.2** Striker shot down **MO1.3** Striker shot upon **MO1.4** Area sanitization (multiple ways of defining areas)
MO2 Suppression of enemy IADS	**MO2.1** SAM attrition rates 50%, 75%, 100% (Will compute as per detonation times)
MO3 Destroy ATO assigned target within TOT:	**MO3.1** Weapon release **MO3.2** Striker on target phase (both planes) **MO3.3** Correct target hit (detonation of the last munition)
MO4 Blue force attrition < 25%	**MO4.1** Self attrition rates 25%, 50%, 75%, 100%

AI Model Creation

As indicated above, the focus of this phase of research was on developing machine learning models capable of automatically predicting the mission phase segmentation (Phase III) and the mission objective evaluation of the entire mission trajectory. We demarcate the level III data by independently representing mission phase segmentation and evaluation of successful execution. The success evaluation is further demarcated into mission objectives, tactical objectives per flight group, and mission phase and appropriate calling of pre-defined contracts. We defined two classification research goals; a) predicting mission phase (Escort Push, Striker Push, Time on Target (TOT), and Egress), and b) predicting the mission objective (see Table 4). Treating the two prediction problems separately allows us to identify three classifier architectures that may be used to develop the prediction model. The overall architectures are defined by different combinations of recurrent neural network-based time-series classifiers and temporal logic-based classifier inferred using Bayesian specification inference (Shah, Kamath, Li, & Shah, 2018). The overall architectures are briefly defined as follows (and explained in greater detail in (Oden et al., 2019))

Problem 1: Predicting Mission Phases

We define two kinds of long short-term memory (LSTM) classifiers for Problem 1, the decoupled classifiers (LSTM (Level-II) and Bi-LSTM (Level II), and the coupled classifiers (we label these as LSTM Coupled and Bi-LSTM Coupled).

The decoupled classifiers take the Level-II phase labels for each of the friendly aircraft as inputs and return the predicted phase labels at each time step as the output. They are both recurrent neural networks that use the output of a module at a given time step as an input of the module at the next time step. The LSTM (Level-II) classifier uses the long- and short-term memory modules to construct the recurrent net. The Bi-LSTM (Level-II) classifier operates on the time-data in both forward and backward direction, thus allowing it to model correlations of the current time instant with both past and future data. By comparison, the LSTM module only models the correlations with past data.

The next set of classifiers (LSTM Coupled and Bi-LSTM Coupled) are coupled classifiers that are provided the Level-II phase labels and the Boolean propositions (see Table 4) and return both the predicted Level-III phase labels and the evaluations of mission objectives. Thus, these networks are single input multi-output.

The network's predictions for Problem 1 are obtained considering the Phase Labels output provided for these networks. The rationale for using a coupled architecture is that this would force the model to share parameters and intermediate values of the network to optimize for all objectives, potentially resulting in better performance on each of the individual objectives. These networks are trained to simultaneously optimize the accuracy of phase predictions and mission objective evaluation.

Problem 2: Predicting the Mission Objectives

We consider three types of classifiers for Problem 2. The first is a decoupled recurrent neural network (RNN)-based classifier (we label them as LSTM (propositions), and Bi-LSTM (propositions) respectively); next is a coupled RNN-based classifier (these are the same ones used in Problem 1 but using the other output channels); lastly, we use Bayesian specification inference (Shah et al., 2018), a model developed as a part of this project to generate predictions for Problem 2.

The LSTM (Propositions) and Bi-LSTM (Propositions) classifiers take the previously defined Boolean proposition values and returns the evaluations of the mission objectives as the output. The architecture is like the LSTM (Level-II) and Bi-LSTM (Level-II) networks respectively, with the difference being the shape of the input channel and the number and shape of the output channel. These networks have four output channels. There is one output each for mission objectives 1 through 4, and each output is a scalar value that represents the belief of the model that the mission objective was achieved by the model. The LSTM (Coupled) and Bi-LSTM (Coupled) are the same networks as those defined for Problem 1 and are trained simultaneously on classifying both Problem 1 & 2. The output channels corresponding to the mission objectives are used for Problem 1.

Bayesian specification inference (Shah et al., 2018) was developed as a part of the project and proposes a probabilistic model for inferring the linear temporal logic (LTL) specifications that best explain an observed valuation of a Boolean time series along with a label representing the acceptability of the execution that the time series represents. The hypothesis space of the model consists of formulas that are composed as a conjunction of three temporal behavior templates; namely global satisfaction, eventual completion and temporal ordering. The complete specification can then be described as a conjunction of these three templates.

$$\varphi = \varphi_{global} \wedge \varphi_{eventual} \wedge \varphi_{order}$$

Thus, the input to the probabilistic model is the set of time series of the Boolean propositions and the label of whether that execution is acceptable or not and a partitioning of the propositions into candidates for global satisfaction and eventual completion respectively. An execution is determined to be successful if the mean of the evaluation of all formulas in the support of the model's distribution weighted by the posterior probability of the candidate formula. As an estimate, the model evaluates an execution as successful if $P(Successful) \geq 0.5$. However, the model can be adjusted to report evaluations only if they exceed a confidence threshold.

Amongst the previously defined Boolean propositions, the four propositions in MO4.1 pertaining to friendly force attrition and MO1.2 and MO1.3 pertaining to whether the strike aircrafts have been attacked are included as candidates for global satisfaction (the set $\boldsymbol{T}$). The rest of the propositions are candidates for eventual completion (the set $\boldsymbol{\Omega}$).

RESULTS

The following experiments were set up to evaluate the accuracy of the classification architectures described above:

Problem 1: The segmentation was performed by either an RNN trained to optimize classification accuracy of Level-III phase labels with Level-II phase labels as data features, or it was performed by an RNN jointly trained to optimize the classification accuracy of both Level-III phase labels and mission objective evaluations with both Level-II labels and Boolean propositions as inputs. For both these classifiers, we performed leave-one-out cross validation (LOOCV) where one scenario was held separate as validation data, the classifier was trained on the remaining scenarios and tested on the validation scenario.

Problem 2**:** As the deep nets are data intensive, the RNN based models for mission objective classification were also evaluated with leave-one-out cross validation. For specification inference, we first performed a fourfold cross validation, where the data set was partitioned into four equal validation sets, and for each held out validation set, the classifier was trained on the remaining three sets and tested on the held-out set. We inferred LTL specifications for the complete dataset as the training set.

Problem 1 Results

The descriptive statistics for the LOOCV are shown in Table 5. LOOCV statistics for RNN models for Problem 1. The statistics reported include the mean and median values of the test accuracy for each of the 24 left out scenarios. We also report the maximum and the minimum test accuracies observed.

Table 5. LOOCV statistics for RNN models for Problem 1

Classifier	Mean	Median	Maximum	Minimum
LSTM (level II)	0.900	0.914	0.994	0.694
Coupled LSTM	0.915	0.935	0.982	0.696
Bi-LSTM (level II)	0.909	0.930	0.984	0.748
Coupled Bi-LSTM	0.923	0.95	0.982	0.733

Problem 2 Results

For Problem 2, the RNN models were evaluated with LOOCV and the specification inference models were evaluated with a four-fold cross validation. The mean classification accuracy is reported for each of the mission objectives (see Table 5). Each of the four models [LSTM (propositions), Coupled LSTM, Bi-LSTM (propositions), and Coupled Bi-LSTM] were shown to have identical accuracies. The reason for this is that the RNN learned to predict the most frequent outcome for all scenarios. Thus, for all the models always predicted 'Achieved' for MO1 and all the model always predicted 'Failed' for MO2.

Table 6. LOOCV mean accuracy for RNN models for Problem 2

Classifier	MO1	MO2	MO3	MO4
Mean accuracy (for all four models)	0.667	0.583	0.667	0.625

Using specification inference, we performed a four-fold cross validation and tested the accuracy after training on the complete data set. The results are shown in Table 7. The accuracy for MO2 is not reported as it was not possible to operationalize the propositions for surface-to-air threats from the JSAF data accurately.

Table 7. Four-fold cross validation results for Problem 2

Classifier	MO1	MO2	MO3	MO4
Specification Inference (four-fold)	0.667		0.708	0.875
Specification Inference (full data set)	0.958		0.917	0.958

Interpretability of Bayesian Specification Inference

By enumerating the propositions included in the three temporal behaviors in the classification model, we can interpret which behaviors the model deems necessary for evaluating the execution as acceptable.

MO1: Gain and maintain air superiority
The propositions included for global satisfaction included MO4.1, MO1.2 and MO1.3. This indicates that the model believes that to maintain air superiority, the self-attrition should never exceed 25% and that the strikers should never be shot upon. The propositions included in eventual satisfaction included MO3.1, MO3.2 and MO1.1. This indicates that eventually the strikers should release their weapon, and at least 75% of the enemy aircrafts must be shot down. Note that this does not require the strikers to attack the correct target, simply release their weapons. Finally, the ordering constraints were found between MO1.1 and MO3.1 indicating that the model expected at least 50% enemy aircraft attrition before the strikers released their weapons.

MO3: Destruction of ATO assigned target within TOT
The propositions included for global satisfaction included MO4.1, MO1.3. This indicates that for MO2, the model will tolerate up to 50% friendly attrition, and it does not enforce that the strikers are never shot upon, just that they are never shot down. The propositions included in eventual completion include MO3.1, MO3.2, MO1.1, and MO3.3. The

requisite enemy attrition rate was 100% (contradictory to global constraints); like MO1, this model also expects the weapons to the eventually released, but, more crucially, it expects correct targeting by the strikers represented by MO3.3. The ordering constraints also enforce complete enemy attrition before TOT close, which seems unnecessary. Thus, the model is transparent about the extraneous requirements it imposes on the execution, and these requirements can be examined and removed by simply deleting the conjunctive clauses where they are included.

MO4: Self attrition < 25%

The propositions included for global satisfaction include MO4.1, MO1.2 and MO1.3; correctly predicting <25% friendly attrition and strikers never being shot down. For eventual completion it also includes, MO3.1, MO3.2, MO1.1 requiring 75% enemy attrition. There are no orders enforced by it. We note that the model for this includes elements of successful completion of MO1 and 3 as well.

Selection of the overall classification architecture

Defining an overall architecture involves selecting a classifier for both problems; because we have four options for Problem 1 and five options for Problem 2, there are 20 possible combinations. The average accuracies on the five classification problems for each architecture choice are show in Table 8. Since some of these combinations resulted in identical accuracy scores, the four types of LSTM-based Problem 2 classifiers are listed on the same row.

Table 8. Architecture permutations and the expected accuracies. ***Note that the accuracies for specification inference are reported for training on the full dataset.***

Problem 1	Problem 2	Level-III Phase	MO1	MO2	MO3	MO4
LSTM (Level-II)	LSTM (all four)	0.900	0.667	0.583	0.667	0.625
LSTM (Level-II)	Specification Inference	0.900	0.958		0.917	0.958
Bi-LSTM (Level-II)	LSTM (all four)	0.909	0.667	0.583	0.667	0.625
Bi-LSTM (Level-II)	Specification Inference	0.909	0.958		0.917	0.958
LSTM Coupled	LSTM (all four)	0.915	0.667	0.583	0.667	0.625
LSTM Coupled	Specification Inference	0.915	0.958		0.917	0.958
Bi-LSTM Coupled	LSTM (all four)	0.923	0.667	0.583	0.667	0.625
Bi-LSTM Coupled	Specification Inference	0.923	0.958		0.917	0.958

DISCUSSION

We note that the RNN based models reported very high accuracy for labeling Level-III mission phases. The bi-directional model performed better than the simple LSTM models. This is to be expected as the bi-directional models can also model correlation between a given time instant and future outcomes, whereas simple models are restricted to modeling correlations with only past observations.

Further, we note that the combined models trained with both Level-II labels and propositions perform better than the decoupled model. This could be either due to propositions being informative for mission phase labeling or it could be due to influence of simultaneous training on multiple objectives. Verifying the cause of better performance of combined models is a possible future research direction.

Next, we note that the RNN models perform very poorly for mission objective assessment and their predictions collapse to the most frequently observed state of the system. This indicates that RNN-based models are not suitable for this task, with only a few trajectories in the training set. Whether this is due to the limited size of the training set or due to limitations of the expressive power of neural nets for logical functions is another direction for future research.

Finally, we note that Bayesian specification inference produces equivalent or higher accuracy with smaller training data sets. Additionally, with a larger training set, there is an improvement the confidence of the model as measured by the number of formulas in the support of the posterior probability distributions and the observation that these posterior probability distributions are more ‘peaky.’

Based on the accuracies, the final classification model for the MARS tools would include a Level-III mission phase predictor based on only that output for the coupled Bi-LSTM model with access to both Level-II phase data and the propositions and mission objective evaluation using the model inferred by Bayesian specification inference.

During execution of the program the research team developed an improved understanding of the role of 'contracts' among pilots. At the start of this phase it was envisioned that contracts would be part of the evaluation of mission execution. It came into sharper focus, however, that contracts are both an evaluation measure of sorts as well as contextual input (Level I data). Contracts are called verbally, and when a pilot calls the contract it should serve as contextual data that is shared with the AI model. Contracts may not be perfectly communicated, however, and other pilots may misinterpret, misremember, or otherwise fail to act properly on the calling of the contract. These deviations in expected performance need to be accounted for as part of the debrief process.

Related future work should focus on expanding the AI's ability to accurately predict achieving tactical objectives for the individual flight elements. This was not feasible in the current study because of the limited model training data set. While the training data was sufficient for achieving a high accuracy in predicting mission objectives, there was insufficient examples to correctly model the variability of tactical objectives. A larger training set that includes variability in tactical objective outcomes would likely allow the model to successfully handle this level of mission description.

Another extension of this work would be the creation of a generative AI in which, rather than just labeling missions, the encoded representation of aircraft behaviors is used to create realistic mission executions for a given mission type. Given that mission success can be encoded in a logical specification, this can allow for a more rapid creation of a constructive in JSAF or similar environment based on mission level descriptions rather than the lower level tailoring of behaviors performed in this study. This capability could be used as part of mission briefing to illustrate a successful execution of a mission based on its objectives and timeline.

Finally, although this effort was focused on an offensive counter-air (OCA) mission, this approach can also be used to automate the scoring of other mission types such as close air support (CAS), defensive counter air (DCA), air interdiction (AI), strategic attack, and combat search and rescue. Models for these different mission types would need to be generated using the same process as was used on this effort. Modeling these other mission types may also pave the way towards greater levels of abstraction of mission objectives.

ACKNOWLEDGEMENTS

We would like to thank Dr. Kevin Gluck of the Air Force Research Labs (AFRL) for his sponsorship of the LM-MIT program. Additionally, we could like to thank Mr. Zach "Zen" Wallace and Mr. Scott "Carp" Carpenter of RCG, Inc. for their invaluable expertise as our pilot subject matter experts.

REFERENCES

Bradshaw, J. M., Hoffman, R. R., Woods, D. D., & Johnson, M. (2013). The seven deadly myths of" autonomous systems". *IEEE Intelligent Systems*, *28*(3), 54-61.

Dahm, W. (2010). *US Air Force chief scientist report on technology horizons: A vision for Air Force science & technology during 2010-2030*. Retrieved from

Feloni, R., & Pelisson, A. (2017). A retired Marine and elite fighter pilot breaks down the OODA Loop, the military decision-making process that guides 'every single thing' in life. *Business Insider*.

Murphy, R., & Shields, J. (2012). The role of autonomy in DoD systems. *Defense Science Board*.

Nellis Air Force Base. (2012). 414th Combat Training Squadron "Red Flag". Retrieved from http://www.nellis.af.mil/About/Fact-Sheets/Display/Article/284176/414th-combat-training-squadron-red-flag/

Oden, K., Craven, P., Landers, K. J., Macannuco, D. J., Sands, T., Shieh, E., & Rane, S. (2019). *Learning for Man-Machine Interoperation and Training (LM-MIT): Final Report*. Lockheed Martin Corp.

Pellerin, C. (2015). Work: Human-Machine Teaming Represents Defense Technology Future [Press release]. Retrieved from https://www.defense.gov/News/Article/Article/628154/work-human-machine-teaming-represents-defense-technology-future/

Shah, A., Kamath, P., Li, S., & Shah, J. (2018). *Bayesian Inference of Temporal Task Specifications from Demonstrations.* Paper presented at the Conference on Neural Information Processing Systems.