Phase II – White Paper

WRT-1085: Trusted Artificial Intelligence (AI) Systems Engineering (SE) Challenge: Seed Funding

Team: Samuel Cornejo, Zeinab Alizadeh, Amal Yousseef, Carter J. Buss, Joe Gregory, Afrooz Jalilzadeh, Alejandro Salado, Pratik Satam (The University of Arizona)

Document Number: UofA-DL-TR-003, v1

Date: December 9, 2024

Prime Contract Number: HQ003419D0003, DO HQ003423F0495

Subcontract Number: 2103596-04

Table of Contents

Problem statement from the sponsor and assumptions

The project consists of three phases:

- *Phase 1*. Explore performance of Artificial Intelligence (AI) models over variety of operational scenarios.
- *Phase 2*. Design of the decision system; human-machine teaming, resilience.
- *Phase 3*. Operational simulation of mission scenarios.

In essence, the teams must address throughout the three phases the following questions:

- Should a human operator assess imagery? Human review of a link takes 30 minutes vs 1 minute for the AI.
- What role should the human play?
- What notions of human trust in AI are important?
- Should the system be architected or operated to influence trust?

In Phase 1, our team primarily focused on exploring the performance of the AI model and starting the foundations to understand the potential design options of the decision system. We considered this necessary to properly contextualize the performance of the AI model.

In Phase 2, our team primarily focused on developing design solutions and algorithms for the decision system, including detailed architectures for human and machine decision making, methods to assess trust, and mechanisms to increase resilience based on estimates of model performance.

Based on the documentation received from the sponsor and communications with them, our team made the following assumptions in our work:

- The Unmanned Ground Vehicle (UGV) is a mine clearing ground robot, which is a scarce resource for the mission.
- The Unmanned Aerial Vehicle (UAV) is a fast, multi-spectral video collection system.
- The time for the UGV to clear a path is:
    - 1 hour per link if a mine/IED is encountered.
    - 20 minutes per link if a mine/IED is not encountered.
- The AI performance data that were provided to the research team corresponds to the performance of the UAV.
- The human SME (operator) reviews video imagery from the UAV. During execution of the challenge, the SME will also get feedback from the UGV on whether a mine was present and will be able to correlate actual performance with predicted performance.
- Computational resources are enough to promptly compute the planning of the tasks of the UAV and the UGV. Our model allows us to relax this assumption.
- The UGV effectiveness in clearing a mine is 100%.

For Phase II, the sponsored required the following deliverables:

- Systems engineering artifacts employed to develop the solution.
- A description of our solution to the problem.

Both are provided in this white paper.

Approach

The work in Phase 1 leveraged our work from Phase 1, which has been refined in line with our findings and the feedback from the sponsor. In addition, in the collaborative spirit of the SERC/AIRC, we reviewed the deliverables of all teams in Phase 1 to identify potential synergies. We decided to use the categorization framework proposed by the George Washington University (GWU) team when assessing the outcomes of our design effort. We found it particularly useful to characterize the main aspects of the resulting architectures and design options.

In Phase 2, we used a semi-formal systems engineering approach. Specifically, we worked on establishing mission needs and requirements, formalizing the Concept of Operations (refined after the one provided by the sponsor), establishing Operational Concepts as the result of trade studies on various mission architectures, and performed a verification and validation assessment. Key aspects of our engineering strategies were the following:

- We adopted mission engineering and defined several mission threads of interest with increasing levels of sophistication, some of them incorporating various types of security concerns.
- We tried to embody a Modular Open Systems Approach (MOSA) to the maximum extent possible. We tried to define as many flexibility/reconfigurability points across integration levels as possible. We made this choice because we believe MOSA to be an effective overarching strategy to achieve high trustworthiness at the mission level from systems and/or platforms operating at a reduced trustworthiness in certain scenarios.
- We employed a multidisciplinary team to tackle and integrate the diverse aspects that influence trust in AI systems. Particularly, the student team consisted of 3x graduate students and 1x undergraduate student, who provided the following expertise to the team: One graduate student with a background in data analytics/operations research, one graduate student with a background in security of cyber-physical systems, one graduate student with depth on all those topics working on integrating the work of the different students, and one undergraduate student focused on generating the SE artifacts in a digital engineering environment.

Mission Needs and Requirements

Mission needs and requirements (and corresponding MOEs) were elicited from the documentation provided by the sponsor, as well as the understanding we developed from discussing with them. Note that numerical or quantitative targets have not been defined by the sponsor. We assumed that, given the exploratory nature of the challenge, all mission needs are defined as objectives to optimize. When an unknown quantity is used, it is given the value ‘To Be Determined (TBD)’.

The overarching mission need is defined as:

MN-1 A battalion needs to safely traverse between two points as quickly as possible.

Definitions:

- A battalion may include TBD soldiers and TBD vehicles of type TBD.
- Safety is measured as the number of casualties and damaged property.
- Quickly is defined less time to traverse a path is preferred over more time to traverse a path. The time to traverse a path is defined from the moment the battalion indicates their intention to traverse the path to the time remaining soldiers (alive) and property (undamaged) arrive to the end point of the path.

The sponsor has defined a notional mission to satisfy this mission need. We use the notional mission without any modification, as explained earlier. The mission is built of a UAV, a UGV, a human operator, and any other elements necessary for them to interoperate.

MN-1 has been derived into the mission requirements listed in Table 1.

Table 1. List of mission needs

ID	Mission Requirement	Justification
MR-1	**Time to clear a path** The mission needs to declare a path as clear for a battalion to move from point A to point B in less than TBD h. *Note 1*: Less time is preferred to more time. *Note 2*: Points A and point B are inputs to the mission.	Identified by the sponsor in the documentation provided to the team.
MR-2	**Effectiveness of path clearance** The mission needs to clear a path for a battalion to move from point A to point B with likelihood over TBD%. *Note 1*: Likelihood refers to a mine being left uncleared on the path. *Note 2*: More confidence is preferred to less confidence. *Note 3*: Points A and point B are inputs to the mission.	Not explicitly identified by the sponsor but derives from the need to safely traverse the path. Clearly, declaring a path as cleared without being so would be inadequate.
MR-3	**Soldier trustworthiness*** The mission needs to yield trustworthiness above TBD to the soldiers that are to traverse the path. *Note*: Higher trustworthiness is preferred to lower trustworthiness.	Not explicitly identified by the sponsor but implicit in the documentation. From a mission perspective, success in the soldiers traversing the path will depend on the extent to which they trust the path has been cleared.

* Note: Trustworthiness in MR-3 refers to trustworthiness experienced by the soldiers on the recommendations provided by the mission. In this project, we will define a second trustworthiness factor when referring to the extent to which information is trusted by the different actors. For example, low trust on an information set provided by the UAV or the human operator may trigger a call for additional information, which slows the clearance of a passage. This trustworthiness would be a contributor to meeting MR-1 (as an indirect measure to *Time to clear a path*) and not a mission requirement on its own.

Characterization of External Systems and Environment

The nodes were originally characterized based on sponsor's dataset, which included the following attributes:

- Type of terrain
- AI Confidence on detecting a mine
- Human Confidence on detection a mine

- Mine vs no mine

We incorporated additional attributes to the characterization to better represent the prediction of the UAV and the SME under a probabilistic approach. These attributes include:

- *AI Mean Estimate*: It represents the first moment of the probabilistic prediction of the UAV.
- *AI Variance Estimate*: It represents the second moment of the probabilistic prediction of the UAV.
- *Human Mean Estimate*: It represents the first moment of the probabilistic prediction of the SME.
- *Human Variance Estimate*: It represents the second moment of the probabilistic prediction of the SME.

In addition, for consistency, the confidence attributes for AI and human detection provided by the sponsor were reinterpreted to represent the trust that the actors have over the predictions of the UAV and the SME, respectively.

Measures of Effectiveness (MOEs)

Four MOEs were defined at the mission level:

- *MOE0*: Time to traverse a path, traced to MN-1.
- *MOE1*: Time to clear a path, traced to MR-1.
- *MOE2*: Effectiveness of path clearance, traced to MR-2.
- *MOE3*: Soldier trustworthiness, traced to MR-3.

Note that MOE1, MOE2, and MOE3 contribute to MOE0. This is shown in detail when presenting the mission model in a later section of this report.

Mission Threads

As per Sponsor's information, mission threads were constructed as variations of different terrain conditions and security threats.

Variation of Terrain Conditions

The sponsor has explicitly provided a terrain type parameter in the scenarios as means of variations across scenarios. In Phase 1, it was identified that the terrain type had a direct effect over the performance of the predictions. In this phase, the assumption is maintained. The terrains that each node can exhibit provided by the sponsor were:

- Grassy
- Wooded
- Swampy
- Rocky
- Sandy

We acknowledge the fact that under real circumstances more terrain variations can spur and that their effect over the components of the system can be different. Up to this point, it has only been

considered that the terrain type affects the quality of the predictions made by the UAV and the SME. Nonetheless, it can be the case that the terrain type affects the performances of other capacities of the components like the traveling time, communication latency, etc. These have not been considered in Phase 2.

Furthermore, we acknowledge that there can exists scenarios in which most of the field can be characterized by few of the terrain types showing very determined variations such as: only grassy, only wooded, only swampy, half grassy-half wooded, moving from grassy to wooded to grassy to X, or even terrains for which a characterization of the detection performance of the actors is not available, such as only snowy.

All these mission threads have been characterized by the performance of the actors in detecting mines. That is, the effect of the terrain variation can be fully described by the performance of the actors. In this phase, we have defined and explored the following as proxies as mission threads:

MT1. Only nodes with high performance by the human.
MT2. Only nodes with high performance by the AI.
MT3. First half good performance human, second halg good performance AI.
MT4. Only nodes with high performance by both.
MT5. Nodes with poor performance by both.
MT6. Nodes with no characterized performance of any of them.
MT7. Random location of nodes (per performance of AI and human, mixed performance).

We characterize these as different confidence metrics on the performance of the detection actors.

Variety of Security Conditions

The sponsor has stated that the mission can be carried out under adversarial attacks, however the sponsor has not explicitly defined which adversarial attacks will happen and the way in which they would impact the mission. This is why the team has investigated the most frequent and impactful kinds of adversarial attacks and deemed GPS Spoofing and Communication Denial of Service as security concerns worthy of evaluation. Below a description of these:

1) *GPS spoofing*. The GPS spoofing attack consists of damage to the performance of the location sensors, a GPS for the case of a drone. The damage is observed by wrongful measurements of the location that in turn affect the control algorithm of the UAV.
2) *Communications denial of service (DoS)*. This attack consists of the partial or total blockade of the communication service of the system. The DoS can affect some or all the communication links and the damage can include latency increase, packets loss or total communication loss.

To evaluate the impact of GPS spoofing can be studied by adding different levels of white noise to the measurement of the location. Consequently,

To evaluate the impact of the communication DoS, communications between the agents can be delayed, lost or totally prevented inside the simulation.

These have been integrated into the mission threads identified earlier. In Phase 3, we plan to more precisely assess the effect of security conditions.

Concept of Operations (ConOps)

The Concept of Operations (ConOps) outlines the process for establishing a safe passage, including mine clearance and the declaration of the path as safe for traversal. The process begins when the battalion or another actor identifies the need to traverse a passage from point A to point B. The mission involves selecting a path, clearing all mines along it, and declaring the route safe. This information is then communicated to the battalion, allowing them to proceed along the cleared path.

Within this general ConOps, a key point of variation arises concerning the timing of the battalion's movement. At one extreme, depicted in Figure 1 as Method 1, the battalion waits until the entire passage is declared mine-free before moving. At the other extreme, depicted in Figure 1 as Method 2, the battalion advances simultaneously with the UGV as it clears mines. Intermediate solutions are also possible, such as the battalion moving one or more nodes behind the UGV. The choice of the optimal solution depends on mission-specific factors, particularly the urgency of crossing the path (related to MOE1), effectiveness in clearing mines (MOE2), and trustworthiness of the battalion (MOE3). Other factors such as threats like enemy fire during traversing should be considered as well but are outside of the scope of this work.

Our solution is designed to accommodate all possible approaches, without constraining any specific option for the battalion’s movement. We treat the variation point (i.e., the timing of the battalion's advance relative to the UGV) as a critical factor in our model, allowing us to evaluate and identify the optimal solution from a mission-level perspective (i.e., driven by MOE0).

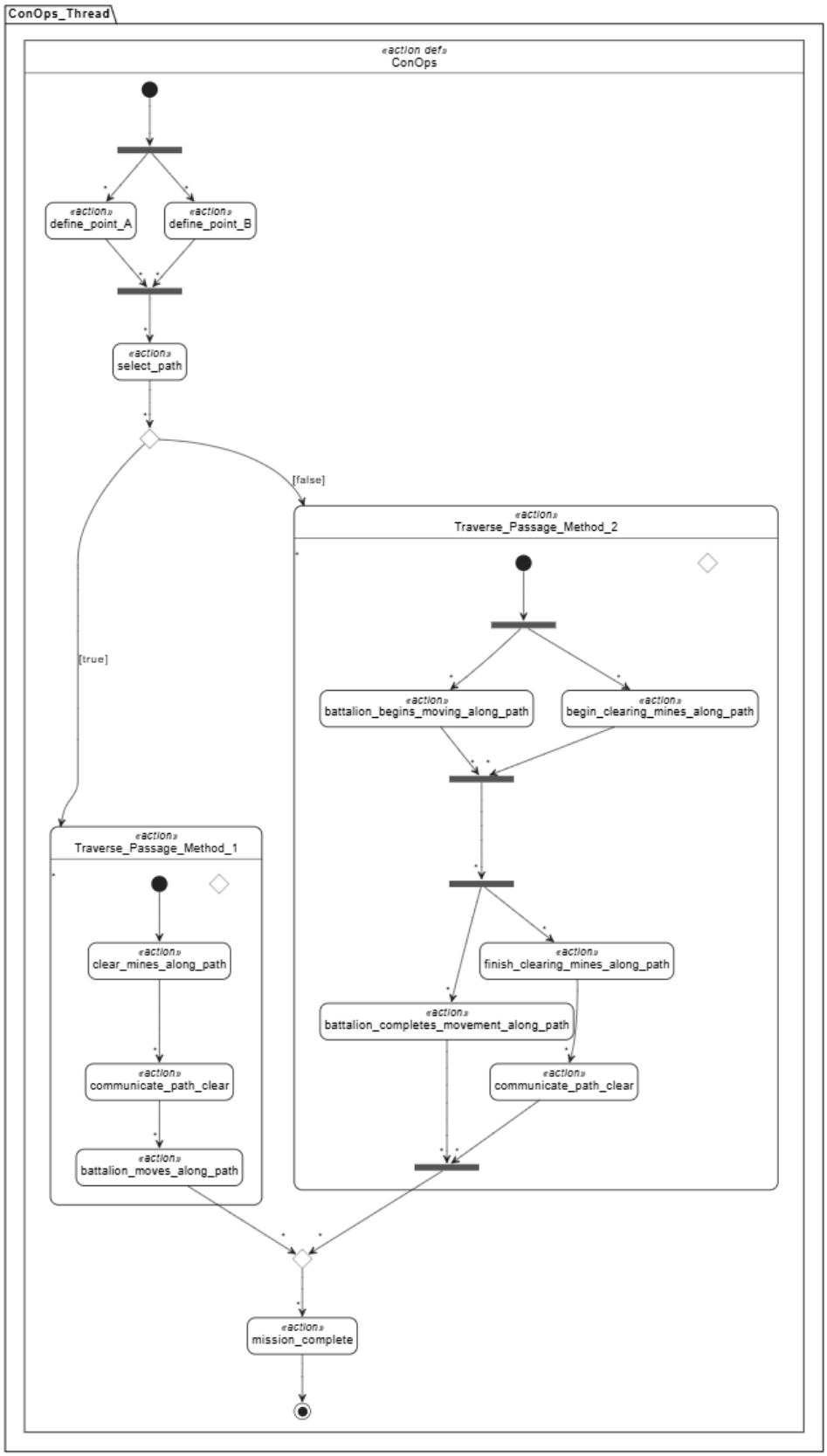

Figure 1. ConOps

Mission Architecture and Operational Architecture (or Operational Concept, OpsCon)

Main drivers and architectural approach

We have identified the *level and distribution of autonomy* as the main design factor in this project. By level and distribution of autonomy, we mean the allocation of tasks, including decision ownership, between human operators, the UAV, and the UGV. For example, does the UGV decide on the clearing of a mine based on its own detection? Does the UGV move to designated areas by the UAV without human intervention?

To enable assessing these different design alternatives, we developed in Phase I a generic functional model of the system, which can be later instantiated into different architectures. Although the sponsor provided a model with the functions or activities performed by the different elements of the system, we believe the generic model was valuable to explore in depth the questions posed by the sponsor, as identified earlier in this paper. The functions (or activities) that have identified to complete the clearance of a path are the following:

- A1. *Decide what area to survey*. This consists of selecting a large area to identify the most promising zones to be cleared, including those points here mines may have been placed.
- A2. *Survey area*. This consists of surveying the area selected in A1.
- A3. *Detect most promising zones*. This consists of identifying the most promising zones to clear in the area surveyed in A2.
- A4. *Command to survey zone*. This consists of requesting a survey of the zones identified in A3.
- A5. *Survey zone*. This consists of surveying the zone requested in A4.
- A6. *Detect mine*. This consists of detecting mines in the zone surveyed in A5.
- A7. *Command to clear mine*. This consists of requesting the clearance of the mine detected in A6.
- A8. *Clears mine*. This consists of clearing the mine requested in A7.

Note that there is no assumption about the temporal dependencies of the functions, other than they probably need to be completed in sequence from A1 to A8. Complete sequentiality may not be necessary, since some tasks might be operated in parallel. For example, A1 and A2 may be executed continuously while A4 through A8 are executed. Such decisions are explored later in this section.

Given the conditions of the problem presented by the sponsor, a minimal allocation of the functions to the different components can be performed, as shown in Figure 2. Note that the efficient survey of a large area can only be performed by the UAV and the safe survey of a zone and mine clearance can only be performed by the UGV. All other functions or activities may be performed by any combination of the Operator, the UAV, and the UGV, including partitioning the activities or a more intricate allocation of subfunctions, depending, for example, on achieving certain thresholds on confidence, conditions of the terrain, expected performance, etc.

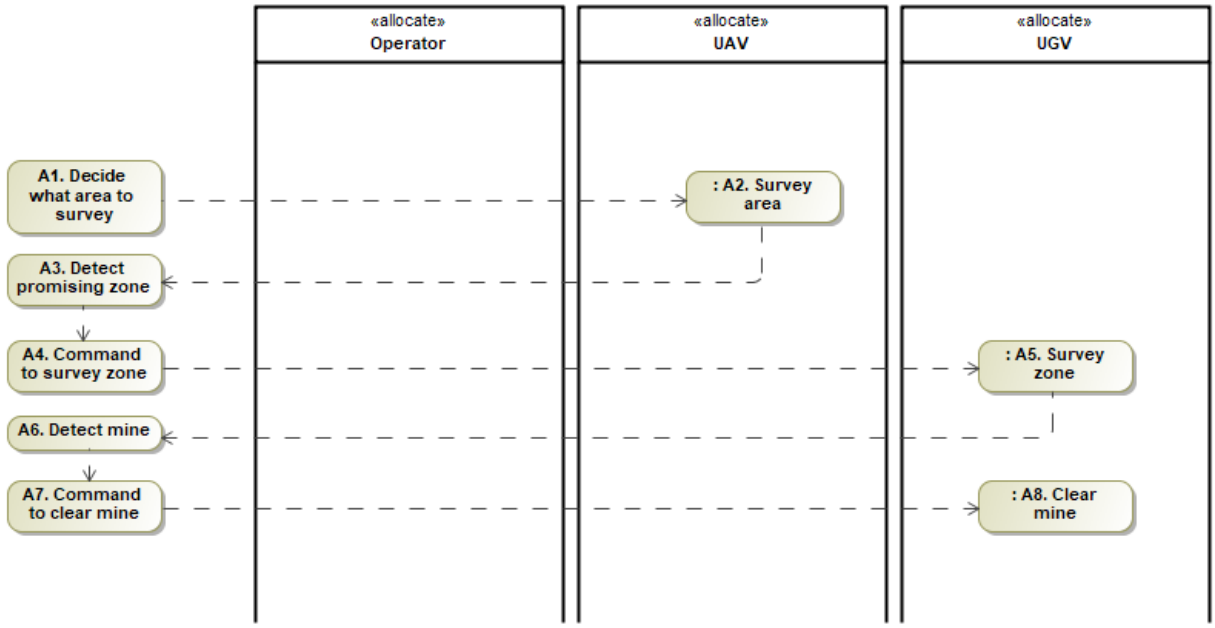

Figure 2. Minimal functional allocation

Security Considerations

As previously discussed, we suggest that the existence of vulnerabilities is a major contributor to trustworthiness. Moreover, the critical nature of the Silverfish operation makes it necessary to characterize it as a Zero Trust Cybersecurity Problem. The Zero Trust Cybersecurity Problem deviates from the traditional defense in depth framework by focusing on the philosophy- "Trust nothing, and verify everything". This section highlights a threat model, evaluated with the Zero Trust Cybersecurity in mind, analyzing the Silverfish system as a 3-Way Security Problem.

The Silverfish system has the potential for targets from different cyber-attack vectors. Here, we highlight some of the potential cyber-attack vectors that will be evaluated in future phases of the project, as part of architecting the potential solutions to the given problem:

1. *UGV/UAV*: UGVs and UAVs are complex systems that depend on various sensors, actuators, and communication networks for their normal operations, opening up opportunities for attackers to target and comprise them. These systems can be targeted with Injection attacks, trojans, side channel attacks, and reverse engineering attacks. These attacks can be used to target the UGV/UAV's control system, sensing systems, and communication systems. The mode of attack execution could be different avenues including networking-based attacks, attacks on the supply chain, or software/firmware attacks. It is critical to secure the UGV/UAV from such attacks, for example through the use of Intrusion Detection Systems and Intrusion Prevention Systems, that are able to detect and stop such attacks when they happen.
2. *AI*: AI models are notoriously vulnerable to adversarial attacks, wherein an attacker injects perturbations into the AI data to cause the AI system to misclassify their answers. Such adversarial attacks are broadly of the following types: a) Evasion Attacks: The attacker

modifies the AI Input slightly so that the model makes an incorrect prediction or classification; b) Poisoning Attacks: The attacker injects misleading or malicious data into the training dataset; and c) Model Extraction and Inversion Attacks: an attacker aims to replicate an AI model by querying it with inputs and using the outputs to reverse-engineer the model. An attacker can perform such attacks through many means, including targeting the supply chain and innovative use of deception and camouflage techniques to cause AI misclassifications. The system can be secured against such attacks through the use of adversarial training and detection systems, for example.

3. *Insiders*: The Silverfish system is vulnerable to insider attacks who can comprise different aspects of its operations, especially the decision-making process. Such insider attacks can be mitigated through strict access control, monitoring, security awareness training, behavioral analytics, and incident response planning.

Mission model

Notation

Denote the graph over which the mission is realized by $G(V, E)$ with nodes V and edges E, also denote by $\bar{V}$ the set of nodes that have a mine and by $\bar{E}$ the set of edges that lead to a node with a mine. Let $x_{i,j}^{B}$ represent the edge traversed by the soldiers from node i to node j and x_i^B represent whether the node i was visited by the battalion. Let $x_{i,j}^{UGV,(1)}$ represent the edge traversed by the UGV from node $i$to node j for the first time, with a value of 1 indicating when the edge is selected, $x_{i,j}^{UGV,(2)}$ as the edge traversed by the UGV from node i to node j for the second time.. Let $x_i^{UGV,(1)}$ represent whether the node i was visited by the UGV one time and $x_i^{UGV,(2)}$ whether the node i was visited by the UGV for a second time. Let $t_{i,j}^{B}$ the travel time of the battalion from node i to node j. Let $t_{i,j}^{UGV}$ the travel time of the UGV from node i to node j, t_{cl} the mine clearing time and t_{fm} the time it takes to the UGV to find a mine. Let $t_{i,j}^{UAV}$ the travel time of the UAV from node i to node j and let λ be a penalty parameter, t_{pred}^{UAV} the time it takes the UAV to make a prediction in node i and t_{pred}^{human} its SME equivalent. Let d_i^B the damage taken by the battalion when traversing node i in time units and let c_i^B be the confidence that the battalion have about whether there is a mine in node i and c_i^{UGV} the confidence that the UGV has about whether there is a mine in node i.

Let h_i^{UAV} represent the decision made by the UAV of making a prediction of mine existence in node i. Similarly, let h_i^{human} represent the equivalent for SME. Also, let μ_i^{UAV} be the mean of the prediction made by the UAV, ${\sigma_i^{UAV}}^2$ the variance of the prediction made by the UAV, p_i^{UAV} the estimate of the prediction made by the UAV and their respective homologues $\mu_i^{human}, {\sigma_i^{human}}^2, p_i^{human}$. Let m^{UGV} denote the efficacy of the UGV at clearing the mine (assumed at 1 when not specified), $per_i^{UAV}, per_i^{human}$ the performance of the predictions that the UAV and the SME exhibit at node i, and a_i^{UAV}, a_i^{human} the probability that the adversary has affected the information of node i provided by the UAV or the SME respectively.

Finally, let t_i^{UAV} be the arrival time of the UAV to node i, t_i^{UGV} the arrival time of the UGV to node i, t_i^B the arrival time of the battalion to node i, tp_i^{UAV} the moment in which prediction realized by the UAV over node i is ready and tp_i^{human} the SME equivalent. Let $h_i^{UGV,UAV}$ be the decision of the UGV to wait

until the prediction of the UAV of node i is ready before traversing node i, let $h_i^{UGV,human}$ be the SME equivalent. Let the index i denote the node where the team starts the mission and n denote the index of the target node. Also, for ease of notation, consider that the second node in the path is defined by $i+1$ and the node before the last is denoted by $n-1$.

Assumptions

In this model, it is assumed that the battalion can only traverse the clear path and that they cannot explore undeclared nodes. However, the battalion can decide not to follow the path if they do not have confidence that the path is clear. In this case, the battalion will remain in the last visited node and not complete the mission.

In the simulation provided by the sponsor, the UGV only has the move command that realizes different actions in the scenario given the existence of a mine in the traversed node. We acknowledge that this may not be necessarily true in a real scenario, as it may be the case that the UGV may need different actions to traverse to a node where it is expected to be a mine than to traverse to another where this is not the case. Therefore, we will differentiate the movement of the UGV given the predictions that the pair UAV-SME make. Also, it is required that the UGV always moves to where it is required to, that is: the UGV cannot refuse the move command.

There is only one prediction resource in both the UAV and the SME. This assumption is taken for ease of modeling, but it can be the case that the UAV or the SME be equipped with multiple prediction engines allowing for parallelization of the predictions.

The battalion follows the UGV exactly one node behind it if it trusts that the node where the UGV just passed by is cleared, otherwise it stalls where it last arrived.

MOE0: Time to Traverse a Path

In MOE0, our primary objective is to determine the total time required by the battalion to traverse a given path. The time to traverse a path is highly influenced by the willingness of the battalion to accept the risks of crossing the declared path given that the battalion can stall while crossing if the confidence is too low.

The sponsor has not explicitly specified a metric that defines the cost that the battalion faces when crossing through a node that has a mine that has not been cleared. If a mine explodes while the battalion is traversing the path, it is considered lethal, and, therefore, the time to traverse the path can be conceived as infinite since the path is never traversed. Therefore, there exists a deterministic component of the MOE0 and a stochastic component. The former is related to the travel time of the soldiers across the path and the latter is the damage taken by the battalion when facing a non-deactivated mine. Evidently, the efficacy of the UGV in deactivating mines has a direct effect on the total traverse time since its work can mean the difference between a finite and an infinite traverse time, but it is not the only factor, as any factor contributing to trustworthiness will affect the pace at which the soldiers will move across the path.

In the code repository provided by the sponsor, the efficacy of the UGV to clear a mine is considered as perfect. However, we acknowledge the fact that this may not necessarily be the case in a general scenario and propose a flexible framework that can consider both cases. Consider that for the definition of the

MOE0, the graph is reduced to the graph conformed by the path taken by the actors. Therefore, MOE0 is modeled as:

$$\text{MOE0} = t_n^B$$

$$s.t.: t_j^{UGV} = t_{ij}^{UGV} x_{ij}^{UGV,(1)} + t_{fm} x_{ij}^{UGV,(1)} + t_{cl} x_{ij}^{UGV,(2)} + \max\left(tp_j^{UAV}, tp_j^{human}, t_i^{UGV}\right), \forall\ (i,j) \in \bar{E}$$

$$t_j^{UGV} = t_{ij}^{UGV} x_{ij}^{UGV,(1)} + \max\left(tp_j^{UAV}, tp_j^{human}, t_i^{UGV}\right), \forall\ (i,j) \in E \backslash \bar{E}$$

$$t_j^B = \max(\max\left(t_j^{UGV}, t_i^B\right) + t_{ij}^B x_{ij}^B +, d_j^B c_j^B c_j^{UGV}), \forall\ (i,j) \in E$$

$$c_i^{UGV} = \begin{cases} \dfrac{p_i^{UAV}}{per_i^{UAV}} \dfrac{p_i^{human}}{per_i^{human}}, x_i^{UGV} = 1, p_i^{UAV} < \epsilon^{mine} \vee p_i^{human} < \epsilon^{mine} \\ 1 - m^{UGV}, x_i^{UGV} = 1, p_i^{UAV} \geq \epsilon^{mine} \vee p_i^{human} \geq \epsilon^{mine} \end{cases}, \forall i \in V$$

$$p_i^{actor} = \begin{cases} \mu_i^{actor} - {\sigma_i}^{actor^2}, \mu_i^{actor} \geq\ 0.5, h_i^{actor} = 1 \\ \mu_i^{actor} + {\sigma_i}^{actor^2}, \mu_i^{actor} < 0.5, h_i^{actor} = 1 \\ 0.5, h_i^{actor} = 0 \end{cases}, \forall\ actor \in \{human, UAV\}, \forall i \in V$$

$$d_i^B = \begin{cases} 0, i \in V \backslash \bar{V} \\ \mathcal{U}(d_{min}, \infty), i \in \bar{V} \end{cases}$$

$$tp_i^{UAV} = (t_i^{UAV} + t_{pred}^{UAV}) h_i^{UAV}, i \in V$$

$$tp_i^{human} = (t_i^{UAV} + t_{pred}^{human}) h_i^{human}, i \in V$$

$$t_{i+1}^{UAV} = t_i^{UAV} + t_{ij}^{UAV} x_{ij}^{UAV}, \forall\ (i,j) \in E$$

Note that the joint probability of the predictions under independence of the predictions was used to define the confidence in case the UGV does not traverses over that node. It can also be the case that the joint probability of both predictions be defined using the conditional probability of one prediction given the other. For the current implementation, independence of the predictions will be assumed.

The confidence that the battalion has over the declared path c_i^B is understood as a subjective measure, since it is dependent on the psychological phenomenon of the battalion with respect to the information provided for the realization of the mission, and it will not be defined here. However, this will be addressed in detail when presenting the model for MOE3.

MOE1: Time to Clear a Path

This MOE can be defined as the cumulative time taken by the UGV to travel from the source to the target, given that it can only traverse to nodes suggested by the UAV-SME while also considering any conditions that may arise along the way. The accumulated time is composed of the travel time to a node for the first time such that the travel beings after the suggestion to traverse it has been realized, the mine finding time, the travel time to a node for the second time in case the UGV decides to clear the mine existing in it, and the mine clearing time. Thus, for MOE1, we aim to compute the accumulated time that the UGV faces until arriving at the destination node. Note that the time to clear a path is defined inside the MOE0 by the following equation:

$$MOE1 = t_n^B$$

$$s.t.: t_j^{UGV} = t_{ij}^{UGV} x_{ij}^{UGV,(1)} + t_{fm} x_{ij}^{UGV,(1)} + t_{cl} x_{ij}^{UGV,(2)} + \max\left(tp_j^{UAV}, tp_j^{human}, t_i^{UGV}\right), \forall\, (i,j) \in \bar{E}$$

$$t_j^{UGV} = t_{ij}^{UGV} x_{ij}^{UGV,(1)} + \max\left(tp_j^{UAV}, tp_j^{human}, t_i^{UGV}\right), \forall\, (i,j) \in E \backslash \bar{E}$$

MOE2: Effectiveness of Path Clearance

The effectiveness of path clearance is related to the path following strategy that the battalion has over the declared path, and the declared path itself depends on the strategy used to define it. For example, if we require that the UGV traverses the declared path and if the battalion chooses to follow the UGV exactly one node behind it, the effectiveness of path clearance will only be affected by the UGV efficacy at clearing the mines, since the UGV will traverse all the nodes of the declared path. However, if the battalion chooses to follow a declared path that does not have all the nodes visited and if needed cleared by the UGV, the effectiveness of the path clearance is affected by information provided by the SME and the UAV. Similarly, as MOE1, MOE2 is partially defined in MOE0 since it is a component of it. Denote MOE2 by:

$$\text{MOE2} = \sum_{i \in V} c_i^{UGV} x_i^{UGV}$$

$$c_i^{UGV} = \begin{cases} \dfrac{p_i^{UAV}}{per_i^{UAV}} \dfrac{p_i^{human}}{per_i^{human}}, x_i^{UGV} = 1, p_i^{UAV} < \epsilon^{mine} \vee p_i^{human} < \epsilon^{mine} \\ 1 - m^{UGV}, x_i^{UGV} = 1, p_i^{UAV} \geq \epsilon^{mine} \vee p_i^{human} \geq \epsilon^{mine} \end{cases}, \forall i \in V$$

Note that these general, probabilistic models are refined in the simulation to guarantee consistency of the model with probability theory on the basis of the number of nodes visited.

MOE3: Soldier Trustworthiness

In MOE3, we focus on evaluating the trustworthiness of the soldiers in the decision-making system. As mentioned in the MOE0 section, we acknowledge that soldier trustworthiness is a subjective metric. Nonetheless, there are factors that can positively affect and negatively affect this trustworthiness. In principle, the soldiers should have more trust in a predictive system that has more True Positives (TP) and True Negatives (TN) than a system that presents more False Positives (FP) and False Negatives (FN). Since TP, FP and TN, FN are related to each other, improving one directly affects the other. Therefore, to define the MOE3 we will assume that trustworthiness can be represented by a function that is proportional to the number of occurrences of FPs and FNs stated by each actor of the declared path.

Consequently, we track the occurrence of False Negatives (FN) and False Positives (FP) of the declared paths. FN refers to the cases where there is a mine but the predictions indicated that there was not a mine, and FP refers to situations where the system incorrectly declares there is mine when there is not. Let v = {nodes in battalion path} and v' ={nodes that have mine}. The trustworthiness of the system can therefore be quantified using the following formula:

$$\text{MOE3} = \sum_{i \in V} c_i^B x_i^B$$

$$FN^{actor} = \sum_{i \in v'} \mathbb{I}_{(p_i^{actor} < e^{mine})} \quad , \quad FP^{actor} = \sum_{i \in v \setminus v'} \mathbb{I}_{(p_i^{human} > e^{mine})}, \forall\, actor \in \{UAV, human\}$$

$$c_i^B = f((FP^{UAV}, FN^{UAV}), (FP^{human}, FN^{human}))$$

Analysis of AI vs Human performance: Boundaries

We developed a predictive model using a neural network to assess AI and human performance based on environmental metadata. The features considered in the model include Surface type, Time of day (Day or Night), Temperature, Wind speed, and Visibility. By training the neural network on the available data, using 70% for training and 30% for testing, we enabled it to predict outcomes under various environmental conditions and determine the accuracy of both AI and human performance. The neural network model was evaluated using the Mean Squared Error (MSE) on the test data. The MSE was found to be 8.8532e-05 and 7.3178e-4 for AI and Human, respectively, indicating the average squared difference between the predicted and actual values. This metric demonstrates the predictive model's ability to predict accurately. The neural network consists of a single hidden layer with 5 neurons. The activation function used was ReLU (Rectified Linear Unit), and the output layer consisted of a single neuron with a linear activation function. The network was trained using the Levenberg-Marquardt algorithm.

To optimize the performance of the neural network, we experimented with different numbers of hidden neurons. We found that using a small number of hidden neurons, such as one or two, resulted in an underfitting model that could not capture the complexity of the data. On the other hand, using a large number of hidden neurons led to overfitting, where the model was too complex for the relatively small and straightforward dataset, resulting in poor generalization to new data. After careful consideration, we settled on five hidden neurons, as this provided a good balance—allowing the model to learn the necessary patterns without becoming overly complex.

Figure 3 represents the training processes of two distinct neural network models—one focused on AI and the other on Human performance. Both plots depict the Mean Squared Error (MSE) across training epochs, comparing the model's behavior on training, validation, and test datasets. In both plots, the MSE for the training, validation, and test datasets shows a consistent downward trend, with all three curves closely aligned. This indicates that the model is well-trained, with minimal overfitting, and generalizes effectively to unseen data.

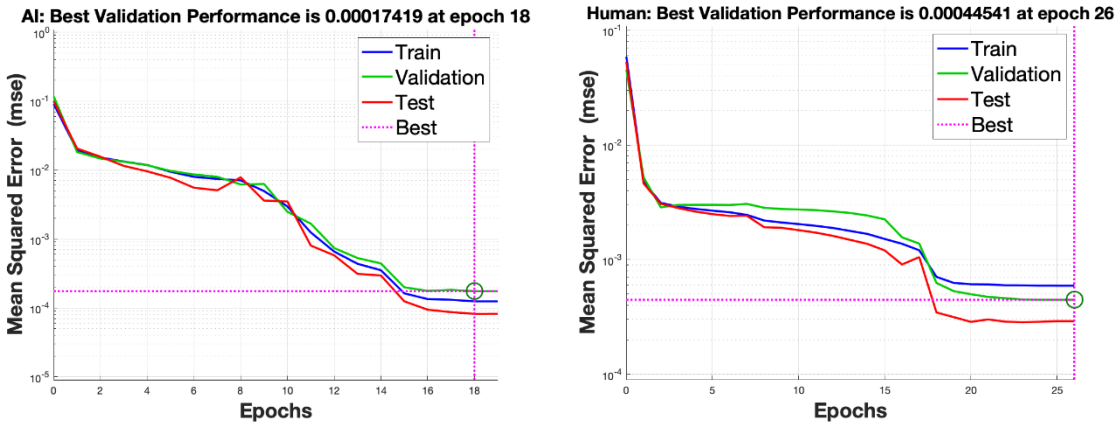

Figure 3. Training process of neural network models (left: AI performance; right: human performance)

The model now allows users to input specific environmental conditions—such as road type, temperature, visibility, and other relevant factors—and receive a predictive output on whether AI or human decision-making would be more effective under those circumstances. This interactive capability ensures that decision-makers can tailor their strategies to real-time conditions, optimizing the performance of the system in diverse environments.

It is important to note that the model is designed with flexibility in mind and can be updated or retrained when new data becomes available. This capability ensures that the predictive model remains accurate and relevant as additional information is gathered, allowing for continuous improvement in performance predictions for both AI and human assessments.

Considering the extensive data on surface conditions but limited information on other features like time of day, temperature, wind speed, and visibility, we now prioritize road condition as the primary variable to derive meaningful insights into when AI outperforms human decision-making. By analyzing the data, we calculated the mean and 95% confidence intervals for the predicted outcomes across different road types. This analysis allows us to clearly identify the scenarios where AI has an advantage over human decision-making and where it might fall short, enabling more informed decisions on which approach to rely on in specific situations. The analysis is visualized in Figure 4. The first plot shows the mean accuracy and the 95% confidence interval for AI across different road conditions, while the second plot presents the same metrics for human decision-making. Both AI and Human models demonstrate lower accuracy in wooded and rocky conditions, with the Human model exhibiting slightly higher accuracy in these challenging environments. However, the confidence interval for the AI model is tighter, indicating less variability in its predictions.

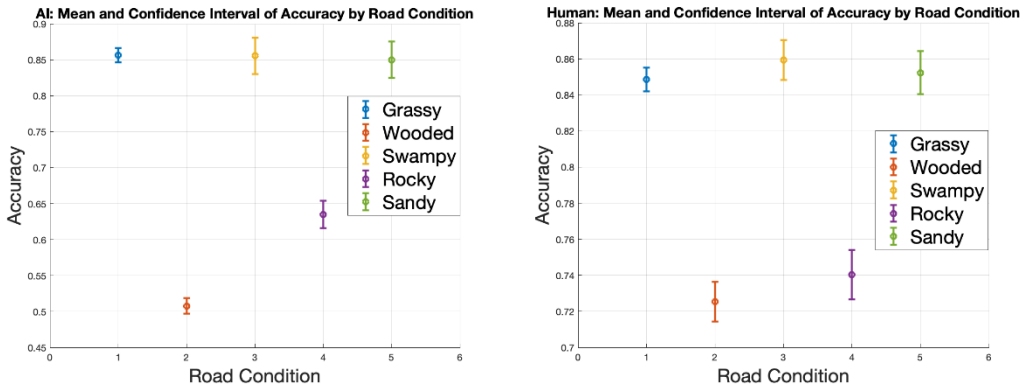

Figure 4. Comparative analysis AI vs human performance

Additionally, Table 2 summarizes the mean accuracy for both AI and human performance under each road condition. This table complements the plots by offering a quick reference to the average performance in each scenario, facilitating a more straightforward comparison between AI and human decision-making capabilities across the different terrains.

Table 2. Mean accuracy summary for AI and human performance under each road condition

	Grassy	Wooded	Swampy	Rocky	Sandy
AI	0.8561	0.5076	0.8555	0.6352	0.8500
Human	0.8485	0.7254	0.8594	0.7403	0.8524

The ultimate goal is to solve the shortest path problem within a mine-clearing mission. The approach will begin by determining whether to rely on AI or human evaluation based on the predicted accuracy for each link in the network. This decision-making process is critical and presents substantial challenges that require a carefully optimized model. The accuracy of predictions not only dictates whether AI or human evaluation is chosen but also significantly impacts the overall time required for demining operations. An incorrect prediction could lead to delays—either by directing the UGV along a path where mines are missed or by wasting time clearing paths that are already safe. Therefore, the next phase of this project will focus on developing an optimization model that carefully balances prediction accuracy with operational efficiency, minimizing the risks associated with incorrect predictions and ensuring the mission is completed as quickly and safely as possible.

Architectural options explored

Architectural options have been created by exploring differences in four variation points:

1) Allocation of prediction tasks, that is, which actor performs the mine prediction of which node.
2) Tasks sequencing and parallelization, that is, the extent to which the UAV, the human operator, and the UGV can perform different tasks in parallel.
3) Centralized / Decentralized coordination of decision making.
4) Predictions tasks prioritization, that is, the extent to which the UAV can explore the terrain without necessarily moving following the physical path between the two points of the path.

These variation points are relevant because they can directly affect the efficacy of the battalion while traversing the path and their efficiency.

The first variation point must be explored since the performance of the UAV and SME is directly affected by the terrain of the nodes at the terrain themselves can change across scenarios. Therefore, three architectural variations are proposed:

a) Predictions done only by UAV, as shown in Figure 5.
b) Predictions done only by SME, as shown in Figure 6.
c) Dynamically allocated predictions to UAV or SME, as shown in Figure 7.

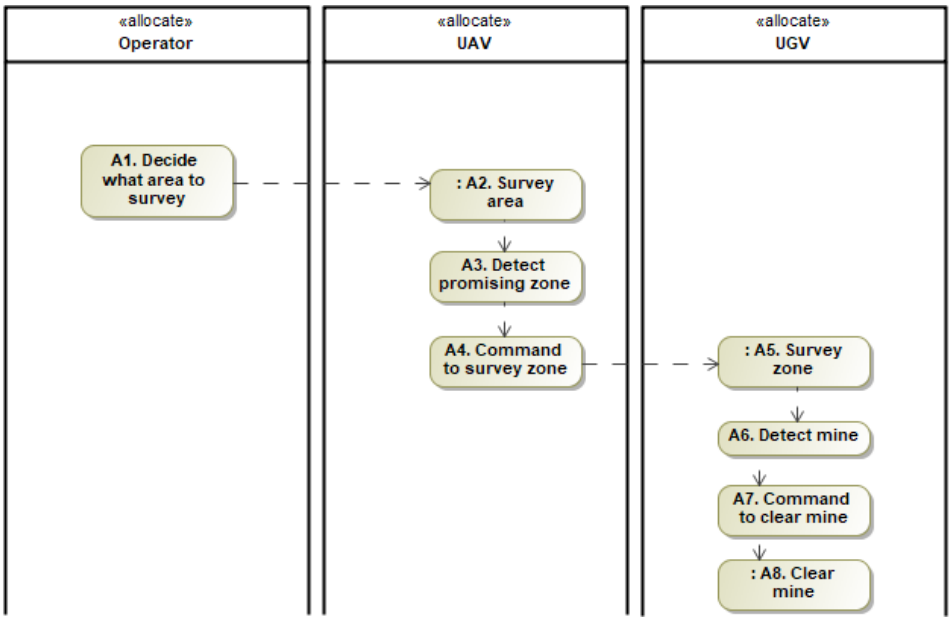

Figure 5. AI-heavy functional allocation

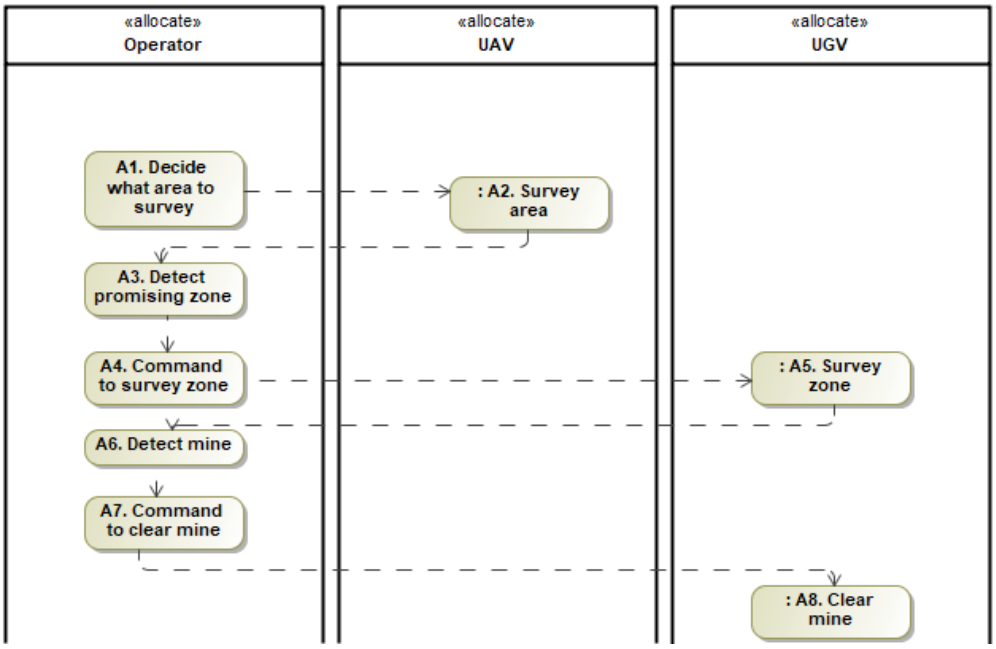

Figure 6. Human-heavy functional allocation

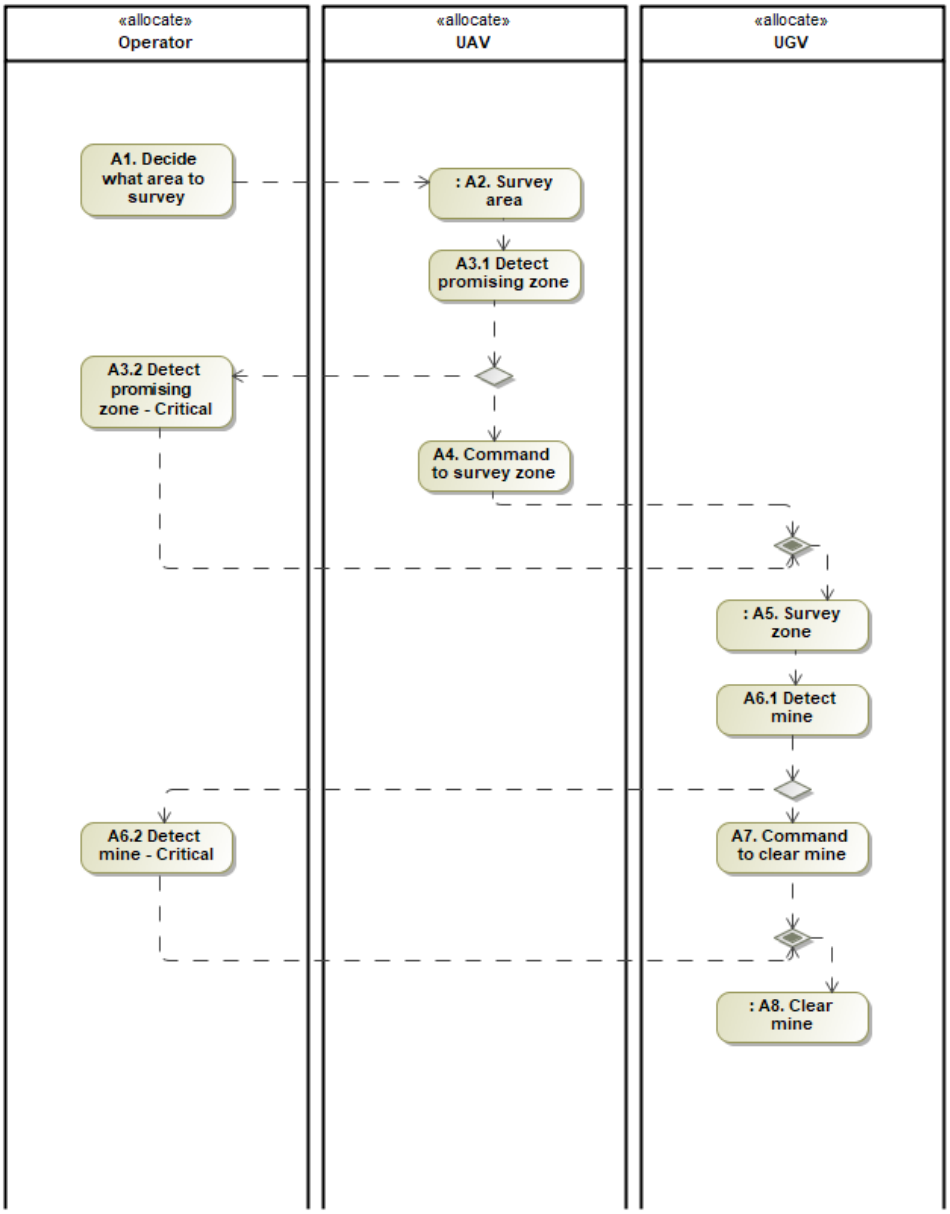

Figure 7. Intricate functional allocation, depending on AI vs human performance

The second variation point studies the effect that tasks parallelization and task precedence constraints have over the MOEs. In this variation point there are two cases. The first allows the UGV to traverse nodes before having received any information from the UAV-SME about it. The second prevents the UGV from traversing nodes about which no predictions have been made yet. These architectural variations are:

a) UGV Greedy Search.
b) UGV Not-Greedy Search.

The third variation point explores the effect that centralized/decentralized planning and coordination have over the MOEs. For this mission, centralized planning involves the sequencing and coordination

of the execution of the tasks in a central computation node. Decentralized planning considers that the sequencing and coordination of the execution is realized in diverse computational nodes. In this case, it will be assumed that the centralized computational node lies on the main base where the SME is. By counter, it will be considered that the decentralized computational nodes are in the UGV, the UAV and the SME's base. These architectural variations are:

a) Centralized
b) Decentralized

The final variation point explores the effect that prediction prioritization has over the MOEs. Prediction prioritization affects the choice of nodes traversed by the UAV, that is: the UAV can be encouraged to prefer to move to nodes where it knows the prediction performance of the UAV superior to the one of the SME, the opposite or show indifference. This variation can be useful to explore since there can be scenarios in which the known performance of one actor is superior to the other one. The variations are:

a) UAV predictions preferred.
b) SME predictions preferred.
c) Indifference.

These variation points allow us to generate 36 generic architectures for exploration.

Verification assessment

Overview

Here, we report the results of a simulation, evaluating the performance of 6 architectures on the mission threads identified earlier. A Monte Carlo approach was used, performing 100 runs of each scenario. Each MOE is reported as the average of the Monte Carlo analysis for each scenario.

Simulation of the Mission Threads

The simulator provided by the sponsor allows the simulation of the mission over a grid-like field. It contains a configuration file that describes the particularities of the scenario over which the mission is realized. In the simulation, the actions taken by the agents are realized in a sequential fashion. Each of the actions taken by the agents have a cost that the simulator accumulates to reflect the MOE1 (Time to clear a path).

One of the limitations of this simulation is that it does not discriminate between time and other resources (like computational power) that are consumed to finalize the mission. This can be observed when the UAV or the SME perform queries, and the cost of the task is accumulated without because the UAV or the SME can have multiple inferencing engines. Thus, instead of accumulating one unit of time and two or more units of computational power, the simulator simply adds all together as the cost of the mission.

Another limitation of simulation is the lack of capacity to simulate parallel tasks while accumulating the total time of the mission. So, if the UAV, UGC and SME want to perform a task at the same time, the sponsor's simulator cannot differentiate the time at which each one of these agents performs

the tasks. This is why we decided to enhance the sponsor's simulator to have these capabilities as we deem them necessary to accurately represent the mission at hand.

Since we are considering that UAVs, UGVs, and the SMEs can operate independently and in parallel, our simulator is designed to handle parallel decision-making, allowing each player to make decisions without waiting for others unless needed. To clarify, imagine a scenario where a UGV is sent to a node while a UAV is observing a neighboring node. The UGV takes 20 minutes to complete its observation, and the UAV does not need to wait for the UGV to finish. In this scenario, UAV's flight time is only 1 minute. Our simulator tracks both UAV and UGV movements over time. To do this, our simulator is formulated as a Discrete Event Simulator (DES) in which all the tasks realized by the agents are scheduled as events.

Since we may encounter unpredictable conditions, it is crucial to simulate how our planning algorithm can adapt to a variety of potential scenarios. To address this, our simulator not only considers the scenario provided by the sponsor but also generates samples of the mission threats. This allows us to evaluate how our strategy performs and assess its impact on all Measures of Effectiveness (MOEs) across different threats under stochastic sampling.

Planning Analysis

The designed planning strategy aims to achieve the minimization of MOE1 and maximization of MOE2 and MOE3. To do this, the role of each agent in the mission must be identified and the way in which they affect the MOEs discussed. In this mission there are two kinds of agents, those that increase the current information of the mission and those that realize the outcomes. The UAV and SME are part of the first kind and the UGV and the battalion are part of the latter kind.

The UAV increases the information of the mission by first providing the imaging of the nodes to the other agents and by making predictions about the existence of mines after visiting these nodes. The SME increases the information by making predictions about the mines with the images provided by the UAV.

The UGV realizes the uncertainty related to the mine existence by traversing the nodes and realizing the outcomes of the actual existence of mine. In case the UGV finds a mine, it can clear it with a degree of efficacy. The soldiers realize the damage of traversing over a node that has a not-cleared mine.

The accomplishment of the mission depends solely on the success that the battalion has at traversing a path. The battalion can choose to follow the path that the system (UGV, UAV and SME) suggests or stop in case they believe it is not safe enough. Therefore, the battalion is coupled with the other agents by the confidence it has in the information they provide and the time they must wait for that information to be released.

The coupling between the UAV and SME lies in the fact that their prediction can only be realized after the UAV has visited a node, therefore the use of the SME as a resource depends on the sequence of nodes visited by the UAV and the arrival time of the UAV to each of these nodes.

The UGV and the pair UAV, SME are loosely coupled depending on the policy that the UGV must traverse nodes. For example, in a greedy approach the UAV and the UGV can simultaneously

traverse different nodes to increase their knowledge of the mission in less amount of time. Thus, the UGV behaves greedily because it makes the best choice of where to move irrespective of the value of future information. By counter, if the UGV has the policy of only traversing nodes that the UAV or the human have previously run predictions on, it behaves in a non-greedy fashion since it takes into consideration the value of future information.

This preamble is relevant because it allows the construction of appropriate strategies to fulfill the mission and highlights the relevance of the proposed architecture in the section above. Since the UAV and the SME can only impact the state of information of the battalion, their objective should aim to generate the largest amount of information of the best quality possible. With regards to the UGV, its objective should aim to clear the shortest path between the origin and target nodes given the information provided by the UAV and the SME, its traveling time, mine finding time and mine clearing time.

For this phase it will be assumed that the battalion moves one node behind the UGV path and therefore, if the UGV is able to reach the end goal, it will be considered that the battalion reaches the goal closely after.

Below a description of the strategies defined for the UGV and the pair UAV, SME to achieve the mission goals while optimizing the MOEs.

Proposed strategy for the UAV/SME

To achieve the objective of generating the largest amount of information efficiently with the highest quality and value possible, the UAV must traverse to the nodes where its performance is the highest without deviating too much from the current path the UGV is following.

At the starting point, when the UAV has surveyed no nodes, there is no information available regarding the probability of mines. As a result, the probability of a mine occurrence on each potential path between two passage points is the same for any two paths. Therefore, the efficiency of the UGV while traversing the graph is constrained by the amount of information provided by the UAV and its trustworthiness. Consequently, the UAV is deployed to survey the environment, to continuously update the hazard probabilities over the nodes neighboring the current path of the UGV in which the prediction is reliable. These updates allow the UGV to dynamically adjust their route planning every time it arrives to a new node and predictions of the UAV, and the SME are available to update the state of information of the UGV.

To generate high quality information updates, the UAV must prefer to survey nodes where its predictions or the SMEs are more reliable, realize the prediction when its performance is better than the SME's and ask the SME to realize the prediction when their performance is better. To generate valuable information, the UAV should visit nodes that are neighboring the current path of the UGV while also being close to the UGV. The reason for this is twofold. Firstly, inspecting neighboring nodes to the UGV current path allows the pivoting or re-planning of the UGV towards other paths while minimizing its detour from its current position. Secondly, the information that closer nodes provide is more valuable for the UGV than the information of very far nodes since the UGV can use the close-nodes information to easily avoid found mines while keeping itself on the newest shortest path to the target.

To generate information efficiently, the UAV must move over the shortest paths of the neighboring nodes which in turn requires a definition of cost over the edges it is traversing. To generate big amounts of information the UAV shall investigate several neighbors. Evidently, since the objective of the mission is to clear a path, the UAV shall also focus on investigating nodes further down the path of the current path of the UGV. This generates a necessary tradeoff between investigating neighbors to the UGV current position and investigating nodes further down the current path.

To solve the tradeoff, a depth d and a breath b parameter will be required from the user. Further optimization techniques can be developed to automatically fine-tune these parameters, but this is left out for the current implementation.

The algorithm works by creating b paths from the current location of the UGV to the node d of the current path that the UGV follows while avoiding the use of all the edges of the path currently used by the UGV. To generate each neighboring path the following shortest path problem is solved given that the source node is the position of the UGV, and the target node is the d^{th}node of the current path and that some of the edges of the current path have been removed:

$$\min \sum_{(i,j)\in E} t_{ij}^{UAV} x_{ij}^{UAV} + \sum_{i\in V} \lambda(\lambda_{pref}^{UAV}(1 - per_i^{UAV}) + \lambda_{pref}^{human}(1 - per_i^{UAV}))x_i^{UAV}$$

$$s.t.: \sum_{j\,:\,(1,j)\in E} x_{1j}^{UAV} = 1,$$

$$\sum_{i\,:\,(i,N)\in E} x_{iN}^{UAV} = 1,$$

$$\sum_{i\,:\,(i,k)\in E} x_{ik}^{UAV} = \sum_{j\,:\,(k,j)\in E} x_{kj}^{UAV}, \quad \forall k \in V \setminus \{1, N\}$$

$$x_{ij}^{UAV} \in \{0,1\}, \quad \forall (i,j) \in E$$

$$\sum_{(i,j)\in E} x_{ij}^{UAV} = x_j^{UAV}, \forall j \in V$$

After the neighboring paths have been defined, they are attached to the shortest path between the current location of the UAV and the UGV, reorganized such that the even neighboring paths are reversed and generate cycles with their immediately previous cycle and finally the nodes are filtered to prevent visiting nodes that have been previously visited or that will be visited by previous neighboring paths. Evidently, some of the already visited nodes will have to be revisited to make the generated neighboring path connected. This can be further optimized but is left for the current implementation.

By creating neighboring paths by means of appending shortest paths subsequently, the efficiency of the sequencing is guaranteed. By searching for promising nodes in the neighboring areas and by allocating the predictions to whom can perform them better, the value and quality of the information is also guaranteed.

Proposed strategy for the UGV

Since the objective of the UGV is to clear the best path possible as fast as possible, the route that it takes should be the expected shortest path from its position to the target node given dynamic updates of information from the UAV and the SME. It is necessary to acknowledge that the UGV has replanning capacities once its belief of the mission is updated and that this capacity unfolds different scenarios over which the expected cost that the UGV faces gets updated. Consider Figure 8, where the term p_ϵ is a placeholder to represent indifference given that the node has not been queried. As can be seen, the UGV leverages prediction information in two out of four cases. The green nodes of the decision tree represent the leveraged cases, the blue case in which information can be generated but has not been generated yet and the orange case in which there is no possibility to generate information. The branches of this decision tree are dynamically attached as the mission progresses, at the beginning only the orange branch exists since the UAV has not had the chance to move around to query information. Later in the mission, the green and blue branches are attached as more information is available. This implies that the efficiencies of the UGV are highly dependent on the amount and quality of information that the pair UAV, SME can generate over time.

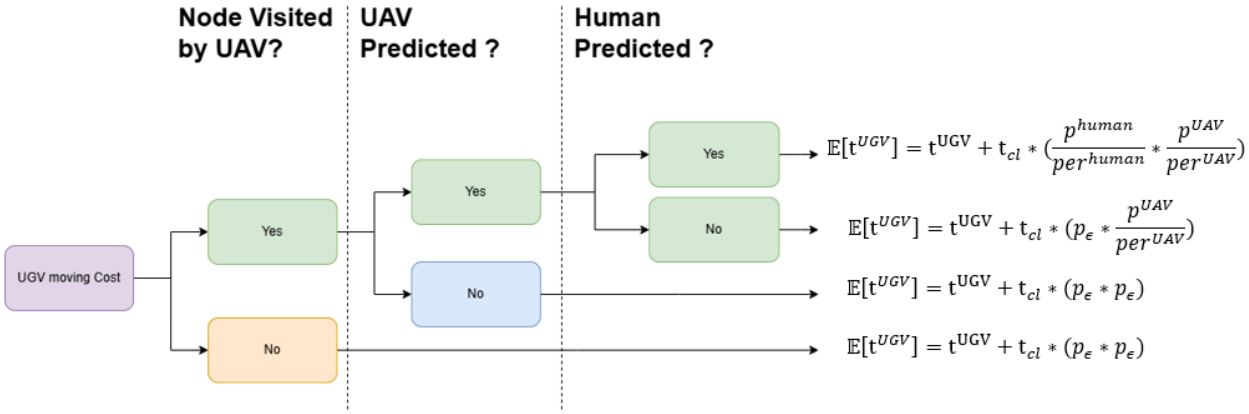

Figure 8. Planning strategy for UGV

Finally, observe the shortest path problem that the UGV must solve to generate its target path.

$$min \sum_{(i,j)\in E} \mathbb{E}[t^{UGV}]_{ij} * x_{ij}^{UGV},$$
$$subject\ to;$$

$$\sum_{j\,:\,(1,j)\in E} x_{1j}^{UGV} = 1, \quad \text{(Starting node constraint)}$$

$$\sum_{i\,:\,(i,N)\in E} x_{iN}^{UGV} = 1, \quad \text{(Ending node constraint)}$$

$$\sum_{i\,:\,(i,k)\in E} x_{ik}^{UGV} = \sum_{j\,:\,(k,j)\in E} x_{kj}^{UGV}, \quad \forall k \in V \setminus \{1, N\}\ (Flow\ conservation)$$

$$x_{ij}^{UGV} \in \{0,1\}, \quad \forall (i,j) \in E\ (Binary\ decision\ variables)$$

Solving the shortest path problem guarantees that whichever graph the UGV is traversing is traversed as fast as possible. But solving the stochastic version presented above guarantees that the best possible paths are generated under uncertainty. The reason is simple, the estimates p_i^{UAV}, p_i^{human} will only be updated when the UAV and SME make predictions about node i. However, the UAV is discouraged from visiting nodes where its predictions are not reliable. Therefore, the formulation above guarantees that the best path is taken given the most reliable information at every step and that such best path is traversed optimally.

Implementation

The simulation used to estimate the MOEs and simulate the mission was realized by wrapping and enhancing the simulation provided by the Sponsor. The scripting was performed using the Python3 programming language with the use of the packages listed in Table 3 in their default versions.

Table 3. Packages used for the implementation of the simulation

Package Name	Description	Documentation Link
NetworkX	Library used to represent graphs and perform algorithms over them.	Software for Complex Networks — NetworkX 3.4.2 documentation
Simpy	Library used to generate the Discrete Event Simulation.	Overview — SimPy 4.1.2.dev7+g9cf45e7 documentation
Numpy	Library that provides several mathematical utilities.	NumPy documentation — NumPy v2.1 Manual

The Sponsor provided the Hexagon and Mission class to run the simulations. In our implementation we added a MissionManager and a Planner class. The mission manager oversees the running of the simulation in a Discrete Event Simulator while simultaneously running the simulation inside Mission class from the Sponsor.

- To generate the sequence of tasks that the UAV and the UGV must perform, the MissionManager provides information to the Planner about the current mission and the graph representation of the mission and queries sequenced tasks from it for each of the actors.
- After the MissionManager receives the tasks, it interacts with the Mission and realizes the Discrete Event Simulation and the one that the Mission class performs.
- During the Discrete Event Simulation, the MissionManager stores the relevant values used to estimate the MOEs and after finishing the simulation it gives it back to the user.

Since we are interested in evaluating the performance of the system under stochastic sampling of a mission thread a scenario generating script was crafted to sample multiple scenarios from a mission thread and to collect the MOEs. After the MOEs are collected, the average of them over a mission thread are computed and written to a csv file.

The simulation is being implemented as a Mission Analysis Pipeline in our Digital Engineering Factory, to mimic a systems engineering effort in a digital environment. We identified the inputs to

the Python analysis and incorporated them into the SysML model of the mission, defined as he ‘Base Scenario’ (ref. Figure 9).

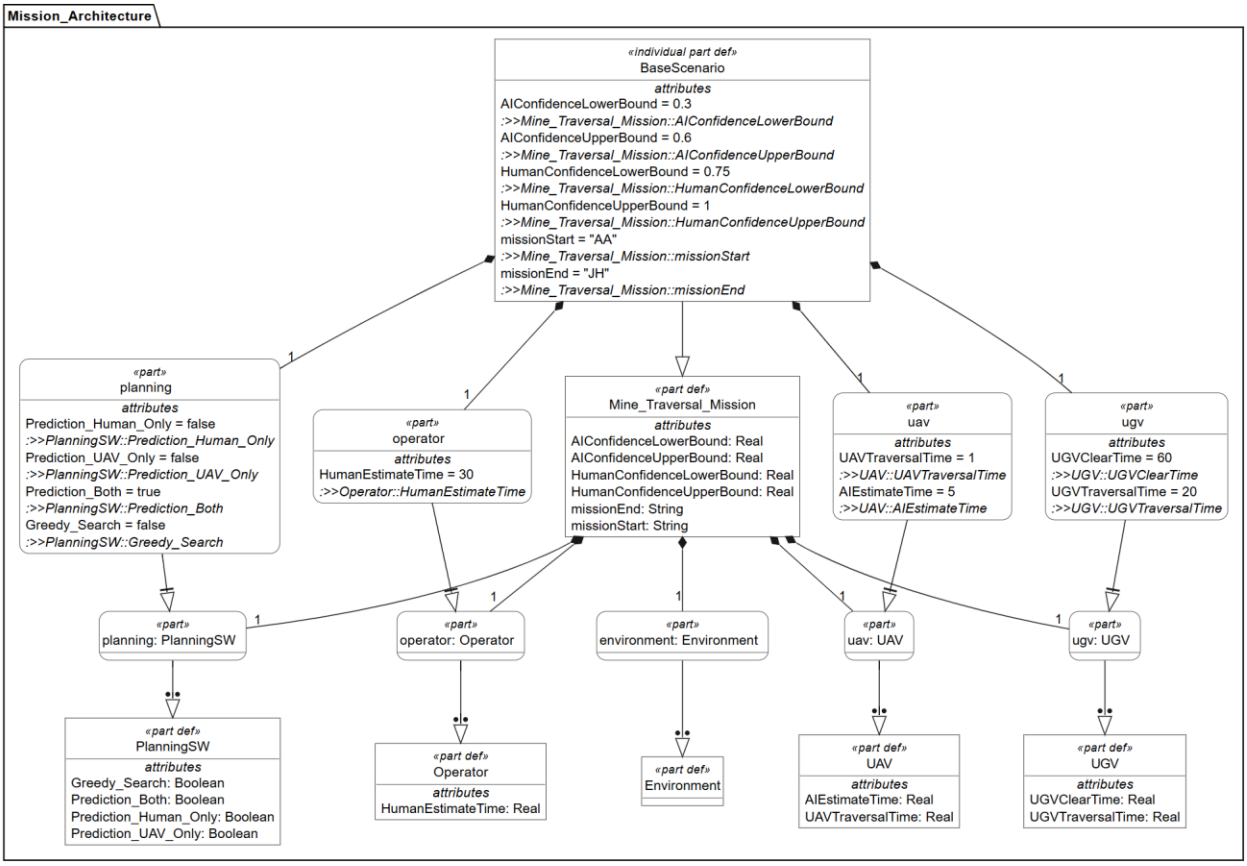

Figure 9. Mission analysis pipeline

In this model, we defined UAV, UGV, and human operator characteristics (such as human estimate time, UAV traversal time, etc.). We also added a ‘Planning’ component of the mission, in which use Boolean values to determine whether the prediction is human-only, UAV-only, or both. We also have the option to check the ‘greedy search’ option. We also define upper and lower bounds for AI and human confidence.

We imported this model into Violet, which we use to aggregate data from different models. The Python scripts identified earlier were also loaded in Violet’s IDE (which is essentially a Jupyter implementation). We updated the script to allow us to link the input parameters to the SysML model values that we had imported into Violet. The results of the simulation (MOEs in Python) are linked to requirements captured in Violet, which then regularly checks that the MOE is within the requirement limit. Some snapshots of the implementation are shown in Figure 10 and Figure 11.

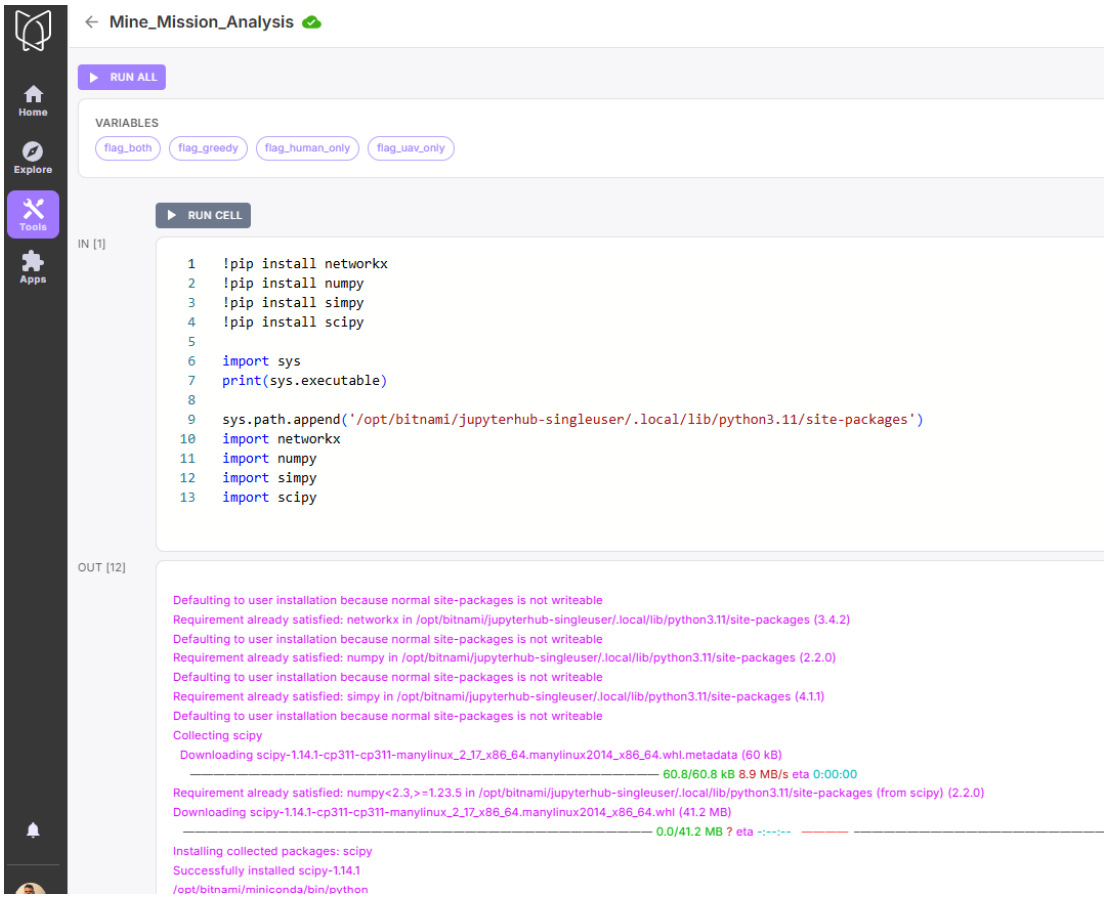

Figure 10. Example of mission analysis planning implementation in the DEF (1)

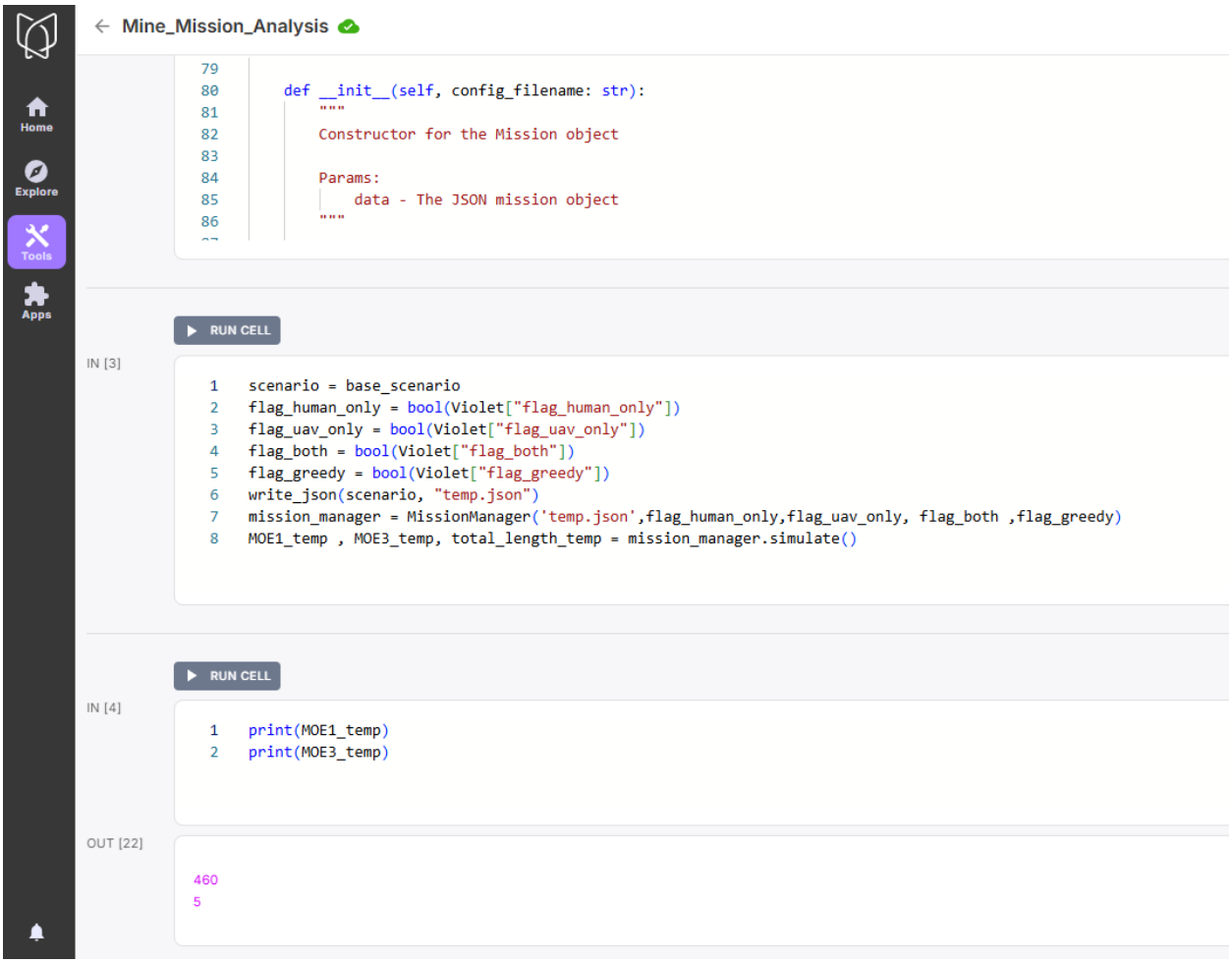

Figure 11. Example of mission analysis planning implementation in the DEF (2)

Results

The results of the simulation are summarized in Table 4. Those values listed as To Be Determined (TBD) were not explored during Phase 2 due to lack of time. We plan to complete this in Phase 3.

Table 4. Results

	Architectures					
	Full AI Not Greedy	Full Human Not Greedy	Intermediate Not Greedy	Full AI Greedy	Full Human Greedy	Intermediate Greedy
Mission Thread 1 Optimal: 282	MOE0: TBD MOE1: 487.6 MOE2: TBD MOE3: 5.1	MOE0: TBD MOE1: 399.6 MOE2: TBD MOE3: 0.48	MOE0: TBD MOE1: 449.6 MOE2: TBD MOE3: 2.33	MOE0: TBD MOE1: 459.6 MOE2: TBD MOE3: 5.01	MOE0: TBD MOE1: 441.8 MOE2: TBD MOE3: 4.63	MOE0: TBD MOE1: 451.2 MOE2: TBD MOE3: 4.15
Mission Thread 2 Optimal: 284.8	MOE0: TBD MOE1: 389 MOE2: TBD MOE3: 0.57	MOE0: TBD MOE1: 468.6 MOE2: TBD MOE3: 4.89	MOE0: TBD MOE1: 389 MOE2: TBD MOE3: 0.57	MOE0: TBD MOE1: 414.8 MOE2: TBD MOE3: 3.02	MOE0: TBD MOE1: 453.4 MOE2: TBD MOE3: 5.13	MOE0: TBD MOE1: 414.8 MOE2: TBD MOE3: 3.02
Mission Thread 3	MOE0: TBD MOE1: 419.2 MOE2: TBD	MOE0: TBD MOE1: 424.4 MOE2: TBD	MOE0: TBD MOE1: 402.4 MOE2: TBD	MOE0: TBD MOE1: 432 MOE2: TBD	MOE0: TBD MOE1: 434 MOE2: TBD	MOE0: TBD MOE1: 420.6 MOE2: TBD

Optimal: 279.8	MOE3: 2.53	MOE3: 2.66	MOE3: 1.28	MOE3: 4.33	MOE3: 4.67	MOE3: 3.77
Mission Thread 4 Optimal: 280.4	MOE0: TBD MOE1: 402.8 MOE2: TBD MOE3: 0.46	MOE0: TBD MOE1: 407.8 MOE2: TBD MOE3: 0.86	MOE0: TBD MOE1: 402.8 MOE2: TBD MOE3: 0.46	MOE0: TBD MOE1: 436.0 MOE2: TBD MOE3: 3.79	MOE0: TBD MOE1: 444.2 MOE2: TBD MOE3: 4.93	MOE0: TBD MOE1: 436 MOE2: TBD MOE3: 3.79
Mission Thread 5 Optimal: 288.4	MOE0: TBD MOE1: 481.6 MOE2: TBD MOE3: 4.85	MOE0: TBD MOE1: 491 MOE2: TBD MOE3: 5.6	MOE0: TBD MOE1: 514.8 MOE2: TBD MOE3: 5.26	MOE0: TBD MOE1: 473 MOE2: TBD MOE3: 5.34	MOE0: TBD MOE1: 468.8 MOE2: TBD MOE3: 5.47	MOE0: TBD MOE1: 482.6 MOE2: TBD MOE3: 5.3
Mission Thread 6 Optimal: 288.6	MOE0: TBD MOE1: 423.6 MOE2: TBD MOE3: 2.65	MOE0: TBD MOE1: 414.4 MOE2: TBD MOE3: 2.62	MOE0: TBD MOE1: 414.2 MOE2: TBD MOE3: 1.52	MOE0: TBD MOE1: 440.4 MOE2: TBD MOE3: 3.89	MOE0: TBD MOE1: 451.6 MOE2: TBD MOE3: 4.86	MOE0: TBD MOE1: 439 MOE2: TBD MOE3: 3.56
Mission Thread 7 Optimal: 285	MOE0: TBD MOE1: 439 MOE2: TBD MOE3: 2.31	MOE0: TBD MOE1: 430 MOE2: TBD MOE3: 2.51	MOE0: TBD MOE1: 406.4 MOE2: TBD MOE3: 0.97	MOE0: TBD MOE1: 429 MOE2: TBD MOE3: 3.6	MOE0: TBD MOE1: 442 MOE2: TBD MOE3: 4.82	MOE0: TBD MOE1: 427.2 MOE2: TBD MOE3: 3.38

Discussion

We planned to perform a detailed assessment of the results but ran out of time to do so. We plan to complete this in Phase 3. In the meantime, we can only provide some first level insights into the results.

As shown in Figure 12, dynamic task allocation generally performs better for MOE1 than relaying only on the human operator or the AI. This is sort of obvious but serves to validate the implementation of our model. The exception, interestingly, occurs in scenarios where performance is unknown for each of the actors. Using the greedy approach seems to reduce the variability of the performance, as shown in

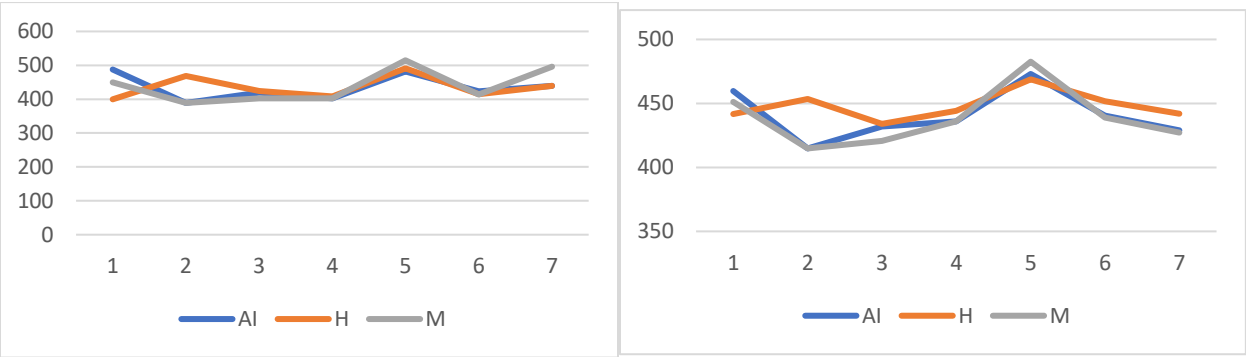

Figure 12. Comparison MOE1 Human vs AI vs Dynamic Allocation (Y axis: time units; X axis: mission thread)

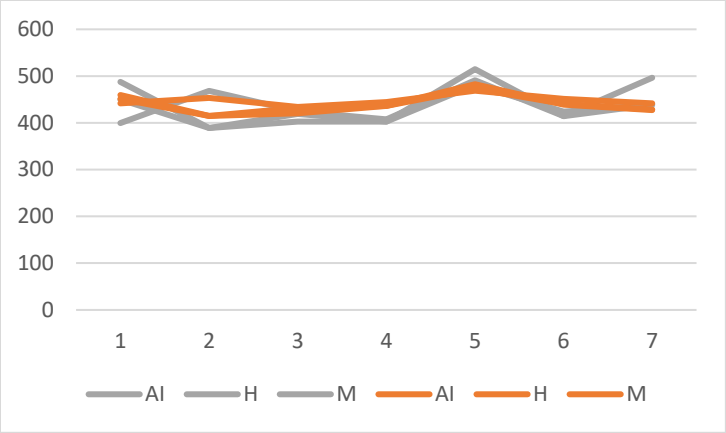

Figure 13. Comparison MOE1 non-greedy (grey lines) vs greedy (orange lines) approaches (Y axis: time units; X axis: mission thread)

Plans for Phase 3

We plan at least the following work for Phase 3:

- Complete the simulation analysis presented earlier for all mission threads, architectures, and MOEs.
- Complete the results assessment to identify insights and patterns that relate mission characteristics (e.g., terrain type, confidence of actor) to the different variation points of the architectures.
- Focus on scenarios where communications and/or performance of some actors is compromised to study mission resilience in detail.
- Document in detail the teaming approach described earlier in this document, particularly as they engage through the digital engineering environment.
- Explore any scenario or aspect of interest for the sponsor identified during the closeout of Phase 2.

Conclusions

In this phase, we have completed the design of the decision system, taking into account several factors related to human-machine teaming and resilience. We have done this by incorporating additional attributes into the source data provided by the sponsor and defining several variation points to explore multiple architectures. The variation points include the ability to allocate decision tasks, the ability to parallelize tasks, the ability to decentralize coordination, and the ability to prioritize node visits. We found this necessary to identify and address the variability necessary to adequately engineer AI-human teaming systems for contested environments. For example, parallelization of tasks and the ability to move from centralized to decentralized coordination, enables reacting to compromises in trust on any of the actors.

Furthermore, we have initiated the evaluation of some mission scenarios, primarily as a validation to the implementation of our solutions and corresponding system and mission models. While we did not have time to complete the analysis and assess the results in detail, the preliminary results indicate that the scenarios where AI-human teams are deployed in environments for which a prior characterization is

unavailable deserve deeper study. This is because the lack of prediction power must be compensated by teaming dynamics between the different actors. This is less straightforward than scenarios for which a prior characterization exists, where tasks can be preplanned to leverage the individual performance of each actor.

We have adopted mission and systems engineering to conduct this project. We have formalized the mission needs and requirements for the project, identified mission threads of interests, established the ConOps and OpsCon, and formally implemented the mission analysis as a pipeline in a digital engineering environment. The work has been performed by a multidisciplinary team, involving students from operations research (data analytics), software engineering and security, and systems engineering, as well as expert modelers to manage the digital infrastructure. We found this to be essential to comprehensively tackle the different challenges associated with the engineering of AI systems.

Our solution has identified several factors that contribute to trust in two dimensions: between the actors of the clearance mission and towards the soldiers, who must traverse the cleared path. Initial results show that trust could play a significant role in achieving the mission needs of the time that it takes to traverse the path.

Finally, the results demonstrate that our approach/solution can be used to leverage the different performance of the human and the AI to establish effective collaboration between them (through task allocation).

Appendix. List of ancillary files

The following table lists the files that were generated in completing Phase 2. They have been submitted as a link to a repository.

File	File name
Code used to assess the bounds and performance of the AI and Human models	codes-selected.zip
Code used for simulation and planning.	repository.zip
Mission Analysis Pipeline with SysML models	Access to the environment