The George Washington University

**Exploring Performance of Different Human-AI System Architectures**

Aditya Singh

Zoe Szajnfarber

December 6, 2024

**Executive Summary**

In prior work, we created a framework that decomposed human-AI systems into 12 distinct architectures. To model each architecture in the SilverFish system, we designed alternative human-AI allocations to test the tradeoffs between architectures. This involved explicitly mapping the decision loop to each actor and in some cases adding handoffs. In addition to modeling the architectures from the framework, our simulation also needed to pull specific maps to traverse and to pull simulation settings that defined variables such as the amount of time to process information for humans and AI, accuracy for both parties, and traversal time. Changing these factors enable us to study the performance vs risk tradeoff under many different operating conditions. In the simulation environment itself, we wanted to track the traversal time and number of mines hit as key performance indicators (KPIs) and to be able to simulate multiple runs.

We modeled two architectures. The results for both architectures, alongside baseline results of a human-only and AI-only system, are shown in Table 5. The first architecture we modeled is a human approver architecture, wherein the computer vision algorithm always performs the initial analysis and passes on its classifications and performance score. The human must approve these classifications before a decision on which link to command the UGV to traverse can be made. The approval heuristic we modeled was reassigning the analysis to a human if the image classifier’s accuracy for the link was lower than 51%, i.e. if it was worse than a coin flip. The human approver architecture avoids all mines in 100%, with the human approver architecture only taking 82 minutes in 76.7% of runs and 167 minutes in 23.3% of runs. This is a significant improvement over the human only version but is still quite a bit slower than the AI only in 23.3% of runs.

The next architecture we modeled was human-AI handoff where the task delineation between humans and AI is fixed. We modeled a scenario where the terrain type delineated which party would perform the classification task. We modeled a situation in which all rocky terrain links would be classified by humans and all other links would be classified by AI, with neither party having authority over each other. This lack of authority eliminates the possibility of reassignment. After classification, we modeled the analysis and decision making using the same heuristic of picking the lowest expected traversal time link. The human-AI handoff architecture avoided both mines and took 67 minutes to traverse the map in 80% of runs and hit one mine, taking 164 minutes in 20% of runs. The results show the same performance as AI only in 80% of runs but present a large decline in performance in the remaining 20% of runs. This large decline enables the architecture to avoid the worst case outcome of hitting two mines, but that worst case outcome for AI only still has higher performance.

Table 1: Results Summary

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Results | Human Only | AI Only | Human Approver | Human-AI Handoff |
| Avg Traversal Time (min) | **212** | **75.6** | **101.8** | 84.4 |
| Avg Mines Hit | 0.0 | 0.26 | 0.0 | 0.2 |

When considering time as the metric of performance and mines hit as the metric of risk, it is clear that the human only and AI only baselines exist on opposite ends of the performance/risk tradeoff. The human only system has the worst performance by far, taking even longer than the AI only system when it hits two mines, but the human only system also avoided mines in 100% of runs. The AI only system had impressive performance, traversing the map quickly even with mines hit, but only avoided mines in 80% of runs. The human-in-the-loop system architectures were able to take advantage AI performance, while retaining human decision making to avoid riskier outcomes, but these two architectures also displayed that where and how humans are placed in-the-loop matters. The human approver architecture avoided mines 100% of the time, while being markedly faster than the human only system (though still much slower than the AI only system).

The human-AI handoff architecture failed to provide as much risk avoidance as human only and human-approver architectures, only avoiding all mines in 80% of runs. In those 80% of runs, the time to traverse was the same as the AI only system avoiding all mines, meaning that in those 80% of runs, the human did not need to act since no rocky terrain was encountered.

In just two simple examples, we showed that *how* the human is integrated into the system can greatly affect the performance and risk profile of a system. Systems engineering as a practice needs to expand the system boundary to consider both the human operators and the AI, as well as the interactions between them. Without human/AI task allocation as a unit of analysis, designers may haphazardly place a human in-the-loop without sufficient attention to where and how they do so. This work also highlights the importance of a system of systems approach to assurance and trust in AI-enabled systems. Research in academia, attention from government, and emphasis from industry has often focused on improving AI to ensure that algorithms are unbiased, transparent, and not overly susceptible to attack or distortion. While these aims are important on their own, they miss the larger ecosystem needed to manage AI-enabled systems. Optimally, human operators can mitigate worst-case failures of AI and enhance its capabilities through collaboration. Creating a system of assurance that centers humans is vital as we increasingly adopt flawed AI into our systems.

The next step is to model each architecture identified in our framework. We will also continue to model architectures across different operating conditions, including varying AI and human performance. The framework provides descriptive design patterns, and the results from the simulation will help us create instructive heuristics of how the identified design patterns should be implemented under different operating conditions. Future findings will be used to develop heuristics to understand which architectures provide superior performance-risk profiles under different operating conditions.

**1.0 Introduction & Motivation**

The use of AI in safety-critical and lethal systems has prompted governments across the world including the United States [1] [2], European Union [3] [6], and China [4] to make calls for AI systems to be overseen by or partnered with human operators. Policy such as the Department of Defense’s directive on the use of autonomous weapons [5] has prescribed human control without providing substantial guidance on how oversight or partnership should be implemented. Generally, the hope is that human-AI teams take advantage of AI’s “greater computational information processing capacity” which “can extend humans’ cognition when addressing complexity, whereas humans can still offer a more holistic, intuitive approach in dealing with uncertainty and equivocality in organizational decision making” [6, p. 577]. While the intended outcomes might be clear, the path to achieving them is not.

Much of the discussion around how to achieve this ideal team has centered around human-in-the-loop system architecture. The term human-in-the-loop was used at least as early as the 1950s to discuss how humans could control increasingly automated functions in aircraft [7] [8]. Despite its growing popularity or perhaps because of it, there is a lack of consensus on what human-in-the-loop control actually means. This is especially problematic since well-meaning policy is being written at this broad level, creating a great deal of uncertainty around how humans should be placed in the loop to achieve the intended policy outcomes.

This paper first presents a systematic review of the ways that human(s) and AI(s) can be integrated in an operational decision loop. Drawing on insights from this review, we create a framework that decomposes different types of human-AI system architectures based on the roles that both humans and AI play in the system. We then show how we can model each of the architectures identified in the framework for the Silverfish Safe Passage system. This will enable us to understand the most effective way to partner AI and human operators under different operating conditions. After presenting the modeling approach, we discuss initial results and conclude with takeaways for systems engineers and plans on future work.

**2.0 Literature Review**

While humans are involved in the entire AI lifecycle from creating and preparing data to setting training objectives and verifying learning outcomes [9] [10], we are focused on human-in-the-loop architecture that defines how humans interact with autonomous systems during operations. Meng’s broad definition describes how human-in-the-loop “generally refers to the need for human interaction, intervention, and judgment to control or change the outcome of a process” in an autonomous system [11, p. 2]. This broad definition makes two key distinctions. First that ‘the-loop’ refers to a process, task, or function of a system rather than the entire system itself. Second, it stipulates that some form of human interaction and influence must be possible. While this broad definition is widely accepted, uses of the term are still inconsistent and mask critical nuances about humans are expected to team with and have control over AI systems. Other authors provide more nuance on different types of human-in-the-loop architecture.

*2.1 What is Human-in-the-Loop*

Munir et al. decomposes human-in-the-loop into three applications: direct human control, system monitoring of humans, and a hybrid of these models [12]. Direct human control is further decomposed into *supervisory control* where a human monitors an autonomous system and affects the process by changing set points and *command control* where a human provides a command that is carried out by a system that later reports the results back to the human for feedback. The second application they describe, where a system monitors a human, flips the paradigm completely. Rather than the human having oversight and control over the system, it is the system that has oversight over the human. They decompose system monitoring into open loop systems, where the autonomous system takes no action based on the information it collects about the human, and close loop systems, where the autonomous system takes action to achieve some goal or avoid an accident. A hybrid system may incorporate aspects of both of these types of human-in-the-loop systems [12].

Lacher et al. describe how human decision makers and automated elements of a system operate with respect to the control loop of a system, which they define as “a series of control operations to carry out a task and/or perform a system function, including receiving reference inputs and system state feedback, deciding the desired system state, and taking action” [13, p. 2]. Their definitions of human-in-the-loop, human-on-the-loop, human-over-the-loop, and human-out-of-the-loop are described in Table 2. Similar to Munir et. al, they also lay out an alternate view which is more focused on the role of automation in the loop, as described in Table 2 [13].

Table 2: Lacher et al.'s Human & Automation Role Analysis

|  |  |
| --- | --- |
| Architecture | Description |
| Human-in-the-loop | Human must act for the system to function; automation serves as an assistant |
| Human-on-the-loop | Automation can control the system and act without human action; human monitors and guides automation, with the ability to takeover at any time |
| Human-over-the-loop | Automation can control the system and act without human action; human monitors and can attempt to change system behavior but has no ability to control or takeover |
| Automation-over-the-loop | Automation monitors a function and alerts human to act, may provide options of action, but cannot act |
| Automation-on-the-loop | Automation monitors a function and can act if needed without human approval |
| Automation-in-the-loop | Automation must act for the system to function; human can help |

*2.2 Literature Gap*

While Munir et al.’s and Lacher et al.’s distinctions are a great improvement beyond the widely accepted human-in-the-loop definition, they do not capture the full depth of possible human-AI control architectures. Their frameworks properly describe situations in which humans and AI collaborate and when they try to provide oversight or control on each other, but they fail to capture many of the nuances of how humans and AI implement collaboration or oversight. Many of these nuances are captured in different existing scales of automation and autonomy, but these scales are often too confined to specific domains or systems to fit into system architecture definitions of human-AI collaboration and oversight, which are more useful in describing and comparing systems broadly. We study these existing levels of automation frameworks to further decompose human-AI system architecture from the current high-level definitions prevalent in the literature today. The following section provides an overview of the frameworks we analyzed.

**3.0 Framework**

Our review of existing levels of automation and autonomy revealed key mechanisms that humans and AI have to exercise control over each other and also revealed more nuanced ways of working together. Using these insights, we created a framework, shown in Figure 1 that decomposes system architecture much further than the literature does today. The framework decomposes human-AI systems by asking mutually exclusive questions about the roles of humans and AI. Our analysis of existing frameworks reveled many variables that are not shown in the framework that could be used to decompose this framework further, but we decomposed the broad array of human-AI system architectures to a level that is focused on mechanisms of control / partnership between humans and AI. The other variables identified are related to how exactly the system is implemented, such as remote monitoring vs in-person monitoring of a system, rather than providing insight as to how the human or AI is expected to exercise oversight or engage together as teammates. These implementation variables will be considered in later stages.

*3.1 Framework Development*

To create our framework, we performed a literature review of human-AI, human-computer, and human-robot collaboration literature. Levels of automation have been a common framework to define how human oversight and control is implemented. Williams describes two types of levels of autonomy scales: linear and multi-dimensional [14]. Generally, linear scales do not define the actions that make up ‘the-loop’ explicitly. Instead, they focus on the roles of humans and automation, with the automation’s role increasing as the levels go up, generally corresponding to a decrease in the human’s role in the larger system. In contrast, multi-dimensional scales define levels of autonomy along a set of tasks a system is expected to perform. This set of tasks can be considered a generalization of ‘the-loop’. In multi-dimensional scales, increasing levels of autonomy may mean increased use of AI in only one, a few, or all of the tasks explicitly defined [14]. The analysis of levels of automation and autonomy analyzes and summarizes linear scales and multi-dimensional scales. In the following subsections, we focus on how the linear and multi-dimensional scales can be aggregated and analyzed to provide further granularity to the human/AI-in/on/over-the loop definitions that exist in the literature today. We use this analysis to inform our framework in the following section.

*3.1.1 Linear Scales*

We studied two types of linear scales: foundational scales that are not context or system specific and applied scales which are context / system specific. Table 1 shows the aggregated analysis for both types of linear scales. At the lowest levels of most frameworks, the human maintains continuous control of the task while the AI system acts as an advisor or as a tool the human operator can delegate subtasks to. The following levels give the AI control over the task but require humans to either select a plan of action from a list of options or to approve the AI system’s accepted plan of action. At these levels, the AI system cannot act without human direction. The next levels see the humans in a supervisory role, meaning that their action is not explicitly required for the AI system to act. In the lower level of supervisory control, the AI system gives the human a restricted time to veto the planned action before it is implemented. The higher level of supervisory control sees the human operator take control over the function of the AI system if there is an issue.

There is also a group of levels that define human-AI teams, where both the human operator and AI system perform tasks throughout the loop and are viewed as equal partners. We found two types of human-AI teams, one in which the subtasks each party is responsible for do not overlap and allocation is fixed and one in which the subtasks may be performed by either party with dynamic tasking. The penultimate level in our table of analysis does not allow the human to act but provides them with insight into how the system is performing. The final level flips the human control paradigm, instead giving AI the ability to override human actions.

Table 3: Summarized Linear Scales

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Architecture | Description of System | Corresponding Level in Frameworks | | | | | | | |
| **Sheridan** | **Endsley** | **Yang** | | **NHSTA** | **Riley** | | **Draper** |
| AI-out-of-the-Loop | All Human | 1 | 1 | 0 | |  | 1 | | 1 |
| AI Tool | **Human maintains continuous control; AI assists** |  |  | **1** | | **0, 1, 2** | **2-7** | | **2** |
| Assists as an advisor |  |  |  | |  | 2-6 | |  |
| Assists with task |  |  |  | |  | 7 | | 2 |
| Human Selector | **AI suggests alternative plans of action à**  **Human selects à AI implements** | **2, 3, 4** | **2** |  | |  |  | |  |
| *Complete set* | *2* |  |  | |  |  | |  |
| *Narrows down alternatives* | *3* |  |  | |  |  | |  |
| *Few alternatives* | *4* |  |  | |  |  | |  |
| Human Approver | **AI executes its decision if the human approves** | **5, 6** | **3** | **2** | |  | **8, 9** | |  |
| *Human must approve* | *5* |  |  | |  | *8* | |  |
| *Human has restricted veto time* | *6* |  |  | |  | *9* | |  |
| Human-AI Team | **Both human and AI perform tasks ‘in-the-loop’** |  |  |  | |  | | **10** | **3, 4** |
| *Human & AI share responsibility for tasks* |  |  |  | |  | |  | *4* |
| *Human & AI task responsibility does not overlap* |  |  |  | |  | |  | *3* |
| *Human and AI can override each other* |  |  |  | |  | | *10* |  |
| Human-on-the-Loop | **AI performs task; human has takeover capability** |  | **4** | **3 & 4** | | **3** |  | | **5** |
| *Humans decides tasks* |  |  | *3* | |  |  | |  |
| *AI decides tasks* |  |  | *4* | |  |  | |  |
| Human-over-the-Loop | **AI executes automatically; informs human** | **7, 8, 9** |  |  | |  |  | |  |
| *Necessarily* | *7* |  |  | |  |  | |  |
| *Only if asked* | *8* |  |  | |  |  | |  |
| *Only if computer decides to* | *9* |  |  | |  |  | |  |
| AI-on-the-Loop | **AI can override human during their operation** |  |  |  |  | | **11** | |  |
| Human-out-of-the-Loop | All Autonomy | 10 | 5 | 5 | | 4, 5 | 12 | |  |
| *Within environmental constraints* |  |  |  | | *4* |  | |  |
| *No environmental constraints* |  |  |  | | *5* |  | |  |

*3.1.2 Multi-Dimensional Scales*

Multi-dimensional scales use high-level task sequences to define levels of automation. Endsley and Kaber’s framework is based around four functions: monitoring, generating, selecting, and implementing [15], although the monitoring function is always shared between the system and the human. Beer, Fisk, and Rogers’ framework for robot autonomy is centered around three functions: sense, plan, and act [16]. Proud, Hart and Mrozinski from NASA’s Johnson Space Center developed a levels of autonomy scale using the observe, orient, decide, and act (OODA) loop [17]. To analyze across these scales, we focused on the different system architectures that were common rather than focusing on the specific ways each framework chose to define ‘the-loop’.

Many levels in the multi-dimensional scales correspond to those observed in linear scales, such as assisted teleoperation being the same as AI tool. While being functionally the same, multi-dimensional scale definitions are often more detailed in describing levels since they explicitly decompose ‘the-loop’. Batch processing, shared control, decision support, and rigid system are all different implementations of human selector architecture, but the multi-dimensional frameworks distinguish systems based on who can generate possible options. Shared control with human/robot initiative is the same as supervisory control in linear scales, but Beer makes a distinction between the two levels by whose initiative, robot or human, is required for the intervention to take place. While the majority of levels in the multi-dimensional scales correspond to levels in linear scales, executive control emerged as a new level. In executive control, the human operator gives a high-level goal to a system to carry out on its own. During operation, the human operator cannot intervene with operations, but there may be opportunities for feedback or evaluation after the fact. This is similar to but distinct from human-over-the-loop architectures observed in the linear scales.

Save and Feuerberg do not define levels of automation on a system level; rather, they define separate levels of automation for four functions: information acquisition, information analysis, decision and action selection, and action implementation [34]. The key distinctions between levels for information acquisition and information analysis levels of automation depend on whose initiative is required for the system to act. Action selection differs based on *who generates options and who selects the plan of action.* Finally, action implementation differs on whose initiative is required for the system to act, whether the system is providing information on how to do a task, assisting in the task, or preforming the task itself, and if the human can monitor and intervene at any time or only during certain times.

Table 4: Summarized Multidimensional Scales

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  | **Description** | **Endsley** | **Beer** | **Proud** |
| Tool | | Human operates; system can intervene during task | --- | Assisted Teleoperation | --- |
| Human Selector | | Human selects plan of action | Batch Processing | Batch Processing |  |
|  | *Shared Control (SC)* | *Human & system both generate options; both act* | *Shared Control* |  |  |
|  | *Decision Support* | *System generates options; system acts* | *Decision Support* | *Decision Support* |  |
|  | *Rigid System* | *System generates options, human must select an option* | *Rigid System* |  |  |
| Human Approver | | System generates & selects plan of action |  |  |  |
|  | *Blended Decision Making (BDM)* | *System implements only if human approves or implements alternate selected by human* | *BDM* | *---* | *3* |
|  | Decision Making with Veto | Human has time to veto before action is implemented. | --- | --- | 4, 5 |
| Supervisory Control | | System acts on its own; human can override | Supervisory Control | Supervisory Control | 6 |
|  | *with Human Initiative* | *human may intervene with new goals / tasks if system encounters difficulty* |  | *SC with Human Initiative* |  |
|  | *with Robot Initiative* | *system can prompt human to intervene with new goals if it faces difficulty* |  | *SC with Robot Initiative* |  |
|  | Human Emergency Override | *System acts on its own, but human can override after action* | --- | --- | 6, 7 |
| Executive Control | | Human gives system a high-level goal; system acts | --- | Executive Control | --- |
|  | *Automated Decision Making (ADM)* | *Human and system generate options; system selects & acts* | *ADM* |  |  |

*3.2 Decomposing Human-AI Systems*

Human-in-the-loop vs on-the-loop is the most widely recognized distinction between human-AI systems. The exact definitions of both terms vary depending on the author, but generally, human-on-the-loop systems place the human in supervisory role where action is optional, while human-in-the-loop systems require some form of human action for the system to function. There is an often-overlooked class of system where AI serves as an optional supervisor called AI-on-the-loop. The question that differentiates these three overarching types of human-AI systems is whose participation is strictly required for the system to function. While these distinctions are helpful, the levels of automation literature provide another layer of resolution relevant to how these broad architectures function. Therefore, the framework further decomposes human-in-the-loop, human-on-the-loop, and AI-on-the-loop system architectures. We will first discuss human-on-the-loop and AI-on-the-loop system architectures since they both place humans and AI in parallel roles to each other, with humans supervising AI in the former and AI supervising humans in the latter. Human-in-the-loop systems are discussed at the end since they are a more special case.

A diagram of a company

Description automatically generated

Figure 1: Human-AI System Architecture Framework

*3.2 Human/AI-on-the-Loop*

Human-on-the-loop system architectures give humans the ability, but not the obligation, to act. AI-on-the-loop system architectures give AI the ability, but not the obligation, to act. In both, there are two distinct types of on-the-loop action. The first is supervisory control whereby the party that is on-the-loop can directly intervene directly in the operations. The second is over-the-loop control, whereby the party that is on-the-loop can only influence operations indirectly. We discuss the architectures for human supervisory control, AI supervisory control, and human/AI-over-the-loop control in the following sections.

*3.2.1 Human Supervisory Control*

There are two architectures for human supervisory control that differ based on when the human can intervene. One is command by veto in which humans may disapprove of the AI system’s planned course of action before execution, but if no negation is implemented, the system functions fully autonomously. The Patriot Missile Defense System, which defends a geographic area from missile attacks, is an excellent example of command by veto architecture. In its automatic control mode, the “system will fire unless human operator halts engagement,” meaning that the system is fully autonomous unless the human operators decide to act [18, p. 6]. The other means of supervisory control is humans taking over the system during operation, serving as a human supervisor. In contrast to command by veto where the AI is still tasked with the function after a human veto, the human takes over control of the system function for some time. Uber’s now-defunct self-driving car service provides a good example of human takeover architecture. Uber employed human operators who were “tasked with overseeing the system’s operation, monitoring the driving environment, and if necessary, taking control of the vehicle and intervening in an emergency” [19, p. 8]. This means that the main function of the system, driving, would be taken over by a human until the potential emergency was over or some other trigger point that would cause the human operator to relinquish control.

*3.2.1 AI Supervisory Control*

There are two ways for AI systems to intervene in human operations, which are differentiated by whose initiative is required for the AI system to act. One option is for an AI supervisor, which monitors the human(s) performing the task, to intervene, when necessary, by taking over control of the function. This may be in case of emergency or if human behavior deviates from some preset bounds. The Airbus Auto Pilot/Flight Director provides a classic example of an AI system that takeover for human operators. If a collision is imminent, the Auto Pilot “initiates and executes a sequence of actions to fly the avoidance maneuvers until the aircraft is clear of conflict” while the crew “cannot modify the ongoing action execution” [20, p. 52]. In this case, it is the system’s initiative that is required for it to act. The alternative is for the human performing the task to optionally delegate to an AI tool. The choice on when and what to delegate is up to the human and can be revoked to return to fully human operation. An example of an AI tool is GitHub copilot, provides suggestions as the human is working but the user must affirm a suggestion for GitHub to act and is not obligated to do so [21]. The question as to how much can be delegated to an AI system would be dependent on its capability and potential organizational constraints.

*3.2.3 Human-over-the-Loop*

Human-over-the-loop system architectures allow humans to influence the system but not to intervene. This influence can take place either before, during or after operation. Before operation, the human can set a high-level goal for the AI system to achieve, leaving the implementation to the AI system. The AutoNav system on the Mars Perseverance Rover “the rover to autonomously re-plan its route around rocks or other obstacles on its way to a pre-established destination” after getting an initial route and destination from the NASA crew on Earth [22]. Intervention during operation is not feasible given the radio delay between Earth and Mars [23]. During operation, humans can adjust set points or provide new goals if the AI system encounters issues or if there is a significant change in the operating environment. An example of this type of executive control would be the RQ-4 drone operated by the United States Air Force. The RQ-4 operators “can provide new guidance in terms of waypoints or heading but cannot directly provide inputs to the control surfaces” [13, p. 2]. After operation, humans serve as a feedback loop, providing feedback on system performance in order to change future behavior. Spam email filters like Gmail’s filter “are continuously updated with …the feedback from Gmail users about likely spammers” to continuously improve the system [24, p. 6].

*3.2.4 AI-over-the-Loop*

AI-over-the-loop architectures give AI systems the ability to influence humans but not to intervene. Just as AI supervisory control was differentiated by whose initiative is required, so too are AI-over-the-loop systems. AI systems may try to influence human behavior through warnings in case of emergency of or other behavior outside a determined set of bounds. Serving as an AI monitor, a system provides the alert or additional information on its own initiative. Traffic Alert & Collision Avoidance System (TCAS) provides an excellent example of a monitor architecture. If a collision is imminent, TCAS “triggers visual and aural indications to the flight crew to perform an avoiding vertical maneuver,” with the “execution of the maneuver itself [left] up to the flight crew” [20, p. 51]. Otherwise, humans may use AI as a decision aid whereby they call on an AI system to provide or analyze information to help them plan. We make a distinction between an AI decision aid and an AI tool by noting that a tool is performing a part of a task, while an aid only provides additional context or information for a human. In essences, an AI decision aid only augments human decision making while leaving the entirety of the task up to the human. En Route Air Traffic Organizer (ERATO) is an example of a decision aid architecture. ERATO is “activated on controller’s initiative” and “helps the controller in identifying all the potential intruders of a given flight in the medium-short term” [20, p. 52].

*3.3 Human-in-the-Loop*

Human-in-the-loop architecture requires active participation from both the human and autonomous system involved in the loop; however, within that broad definition, we found two distinct types of human participation. The first is humans serving as directors for AI systems, meaning that the human is responsible for deciding a plan of action for the AI system to carry out. The other type of participation is teaming, where both the human and AI participate throughout ‘the-loop.’ Without the action of both parties, the loop cannot be completed. We discuss the nuances of both of these types of human-in-the-loop systems in the following sections.

*3.3.1 Human-in-the-Loop: Director*

As the party responsible for deciding how an AI system should act, humans can be either implemented as approvers or selectors. The key differentiating factor is the number of potential plans of action that the human operator is assessing. If the AI system generates a single plan of action to be approved or denied, the human is serving as an approver. The system cannot act without approval. If denied, the system may propose a new plan of action or may cease function, depending on the implementation of the system. An example of a human approver system is the Patriot Missile Defense System’s semi-automatic mode, in which the “human operator must authorize engagement or system will not fire” [18, p. 6]. Under selector architecture, the AI system generates several plans of action for the human to choose among. Surgical robots in a wide range of practices including “bone milling in orthopedic surgery, prostate biopsy in urology, and hair follicle extraction in plastic surgery” employ selector architecture [25, p. 2]. Generally, these systems generate “potential strategies for [the] surgeon to select” from, after which the “system takes over control to execute the selected plan” [25, p. 3].

*3.3.2 Human-in-the-Loop: Team*

Human-AI team architectures involve humans throughout the loop, with two distinct types of teams. One is human-AI handoff where the subtasks that the human and AI system are responsible for fixed sub-tasks. For example, U.S. Customs and Border Control (CBP) uses facial recognition to match travelers to their photos, helping catch imposters, while officers perform behavioral / contextual screening [26]. Hypothetically, there should be no overlap between task responsibilities, but governance requirements can create the need for a human or AI backup. In the case of CBP, travelers have the right to request human verification, but in lieu of travelers exercising this right, the tasks are divided between humans and AI. The other type of teaming architecture is human-AI partnership where the task allocation is dynamic, meaning that many subtasks can be completed by either the human or the AI. Security robots have employed in a variety of municipalities and private businesses. These robots can actively patrol and record events, with some being able to step in and call for help if a disturbance is detected [27]. These robots act as partners to security guards since they can both do the same tasks, such as going on patrol, depending on environmental or other circumstances.

**4.0 Modeling Approach & Initial Results[[1]](#footnote-1)**

Few papers have compared the performance of different system architectures. Those that do typically focus only on ‘in-the-loop’ participatory control vs ‘on-the-loop’ supervisory control [28] [29] [30] [31]. Additionally, since ‘in-the-loop’ and ‘on-the-loop’ have typically been ill defined, it is difficult to understand how findings of these papers should be implemented. Thus, it has been unclear what the tradeoffs of different architectures are. By implementing different human-AI architectures in the same reference system, we hope to understand the tradeoffs of implementing different system architectures. The key behavior we hope to understand through this study is the tradeoffs between *performance*, using measures such as total amount of time it takes to move troops across the different networks provided and *risk*, using measures such as the percentage of safe traversals, for each architecture.

*4.1 Modeling Approach*

To be able to model each architecture in the SilverFish system, we designed alternative human-AI allocations to test the tradeoffs between architectures. This involved explicitly mapping the decision loop to each actor and in some cases adding handoffs. The baseline decision model is shown in Figure 2. We assume that the UAV must collect data for every link ahead, taking one minute per link. Data transmission is instant, and analysis of a link can begin while the UAV collects data on other links. The classification of links as clear or unclear can be done by a human or an AI. The results are parsed as the classification with the associated accuracy. If there is a lack of confidence in the party that did the initial analysis, it could be reassigned. After classifying each link, one link must be chosen. The assumed decision-making rule is always choosing the lowest expected traversal time. This decision-making rule may be overturned, and a different link may be chosen. After the final decision is made, the command is instantly sent to the UGV. This process is repeated at each node until the UGV reaches the final node. We change the assignment of responsibilities for every architecture we simulated.

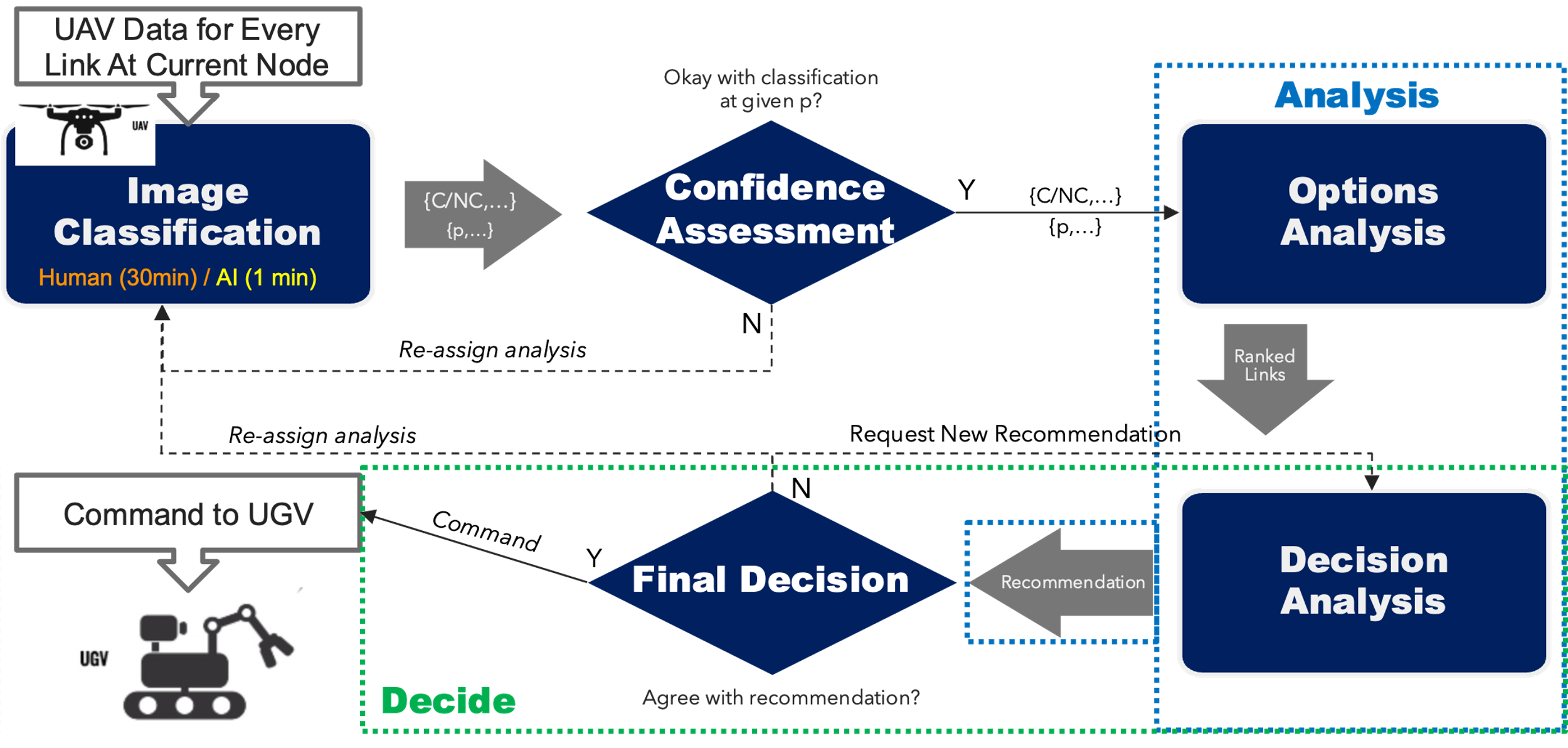


Figure 2: Baseline Decision Flow

In addition to modeling the architectures from the framework, the simulation also needed to pull specific maps to traverse and to pull simulation settings that defined variables such as the amount of time to process information for humans and AI, accuracy for both parties, and traversal time. Changing these factors enable us to study the performance vs risk tradeoff under many different operating conditions. In the simulation environment itself, we wanted to track the traversal time and number of mines hit as key performance indicators (KPIs) and to be able to simulate multiple runs. This modeling approach is summarized in Figure 3.



Figure 3: General Modeling Approach

*4.2 Baseline Simulation Results*

We first tested our simulation with two baselines: a human only system and an AI only system. The map we used for all initial results is shown in Figure 4. Under both architectures, we eliminated the ability to reassign analysis or to ask for a new link since there was only one party, human or AI, to perform the tasks. The results for the baseline simulation are presented in Figure 5. The human only system avoids the mines in 100% of runs, but due to the slow decision-making time, it takes the human only system 212 minutes to traverse the map. In contrast, the AI only system avoids all mines 80% of the time and takes 67 minutes when doing so. In 18.4% of runs, the AI only system hits one mine, adding 40 minutes to the total traversal time, and in 1.6% of runs, the AI only system hits both mines, adding 80 minutes to the total traversal time over the baseline.

A screenshot of a video game

Description automatically generated

Figure 4: Simulation Map

A screen shot of a graph

Description automatically generated

Figure 5: Baseline Results

*4.3 Results Across Different Human-in-the-Loop Architectures*

After modeling the baseline scenarios, we modified the decision flow’s assignments to fit into two architectures discussed in the literature review. The first architecture we modeled is a human approver architecture, as shown in Figure 6, wherein the computer vision algorithm always performs the initial analysis and passes on its classifications and performance score. The human must approve these classifications before a decision on which link to command the UGV to traverse can be made. The approval heuristic we modeled was reassigning the analysis to a human if the image classifier’s accuracy for the link was lower than 51%, i.e. if it was worse than a coin flip. The results for the human approver architecture are shown in Figure 7, alongside the baseline results. The human approver architecture avoids all mines in 100%, with the human approver architecture only taking 82 minutes in 76.7% of runs and 167 minutes in 23.3% of runs. This is a significant improvement over the human only version but is still quite a bit slower than the AI only in 23.3% of runs.

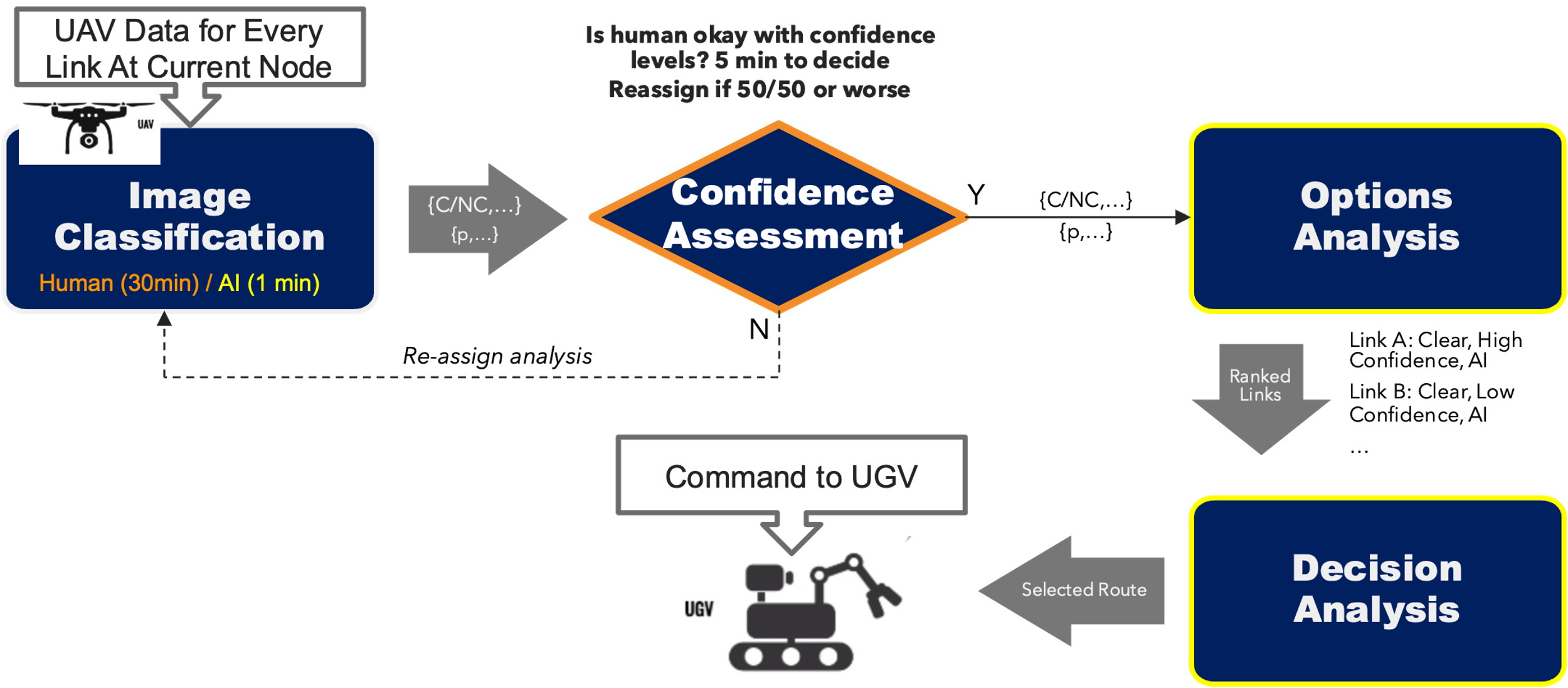


Figure 6: Human Approver Decision Flow

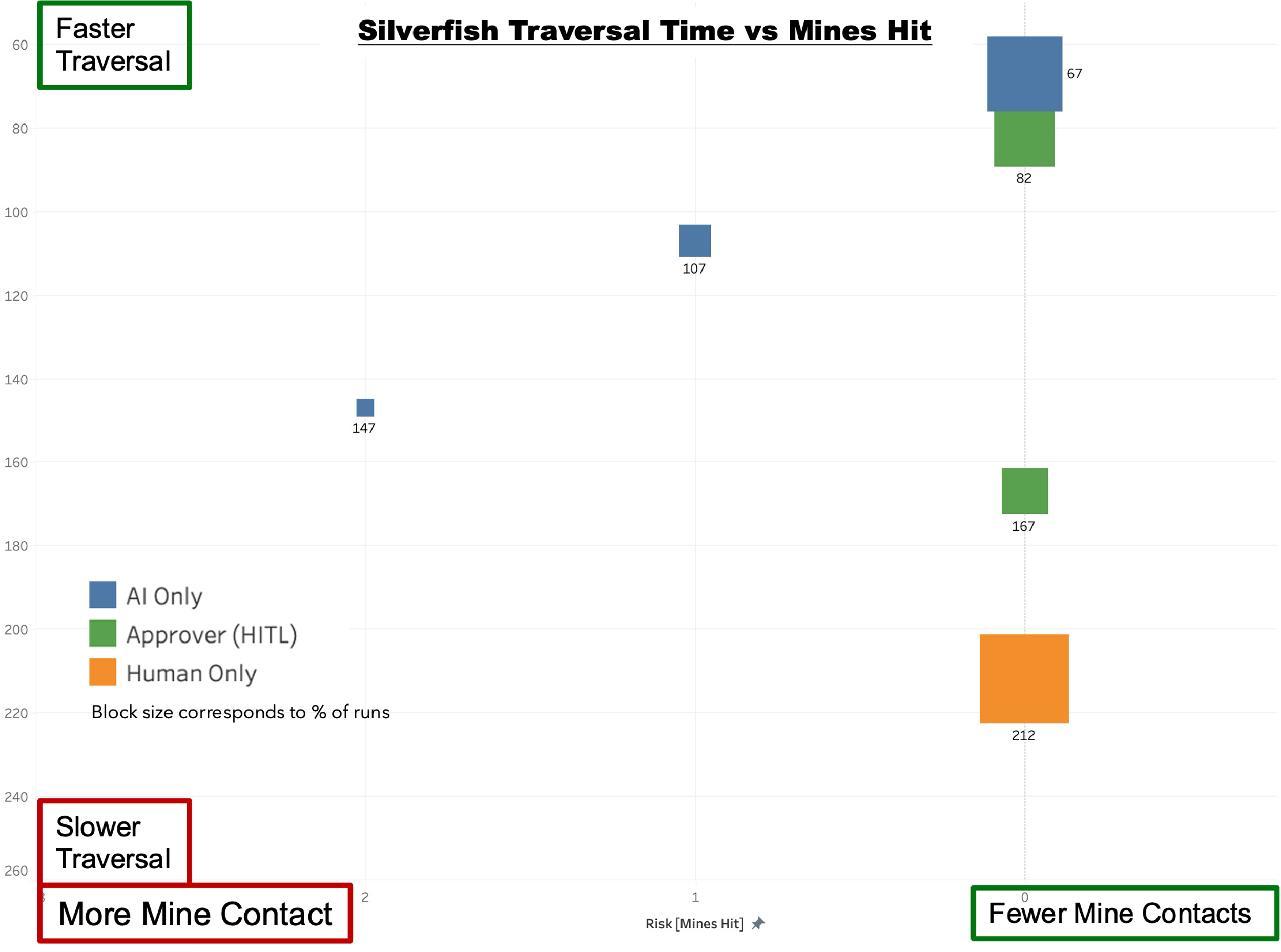


Figure 7: Human Approver Results

The next architecture we modeled was human-AI handoff, shown in Figure 8, where the task delineation between humans and AI is fixed. We modeled a scenario where the terrain type delineated which party would perform the classification task. We modeled a situation in which all rocky terrain links would be classified by humans and all other links would be classified by AI, with neither party having authority over each other. This lack of authority eliminates the possibility of reassignment. After classification, we modeled the analysis and decision making using the same heuristic of picking the lowest expected traversal time link. The results for the human-AI handoff are shown in Figure 9, with the human-AI handoff architecture avoiding both mines and taking 67 minutes to traverse the map in 80% of runs and hitting one mine, taking 164 minutes in 20% of runs. The results show the same performance as AI only in 80% of runs but present a large decline in performance in the remaining 20% of runs. This large decline enables the architecture to avoid the worst case outcome of hitting two mines, but that worst case outcome for AI only still has higher performance.

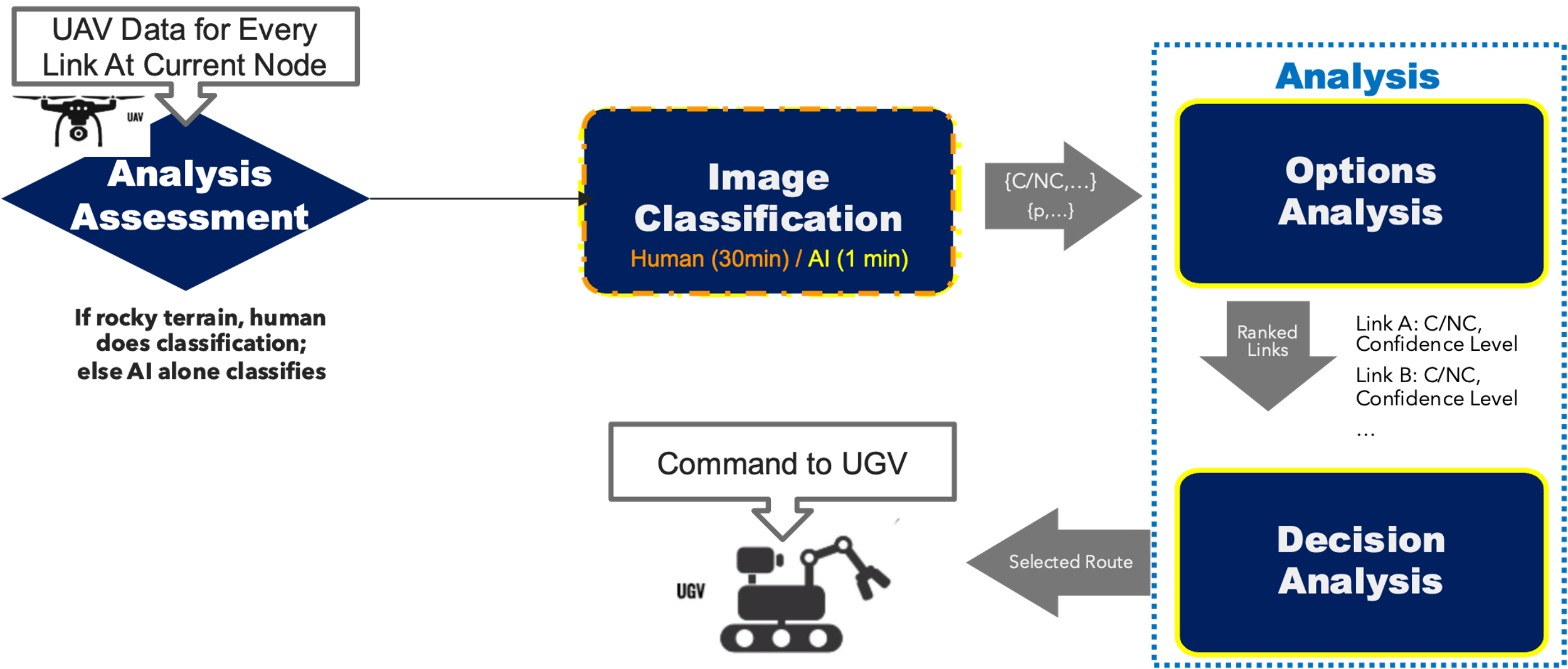


Figure 8: Human-AI Handoff Decision Flow

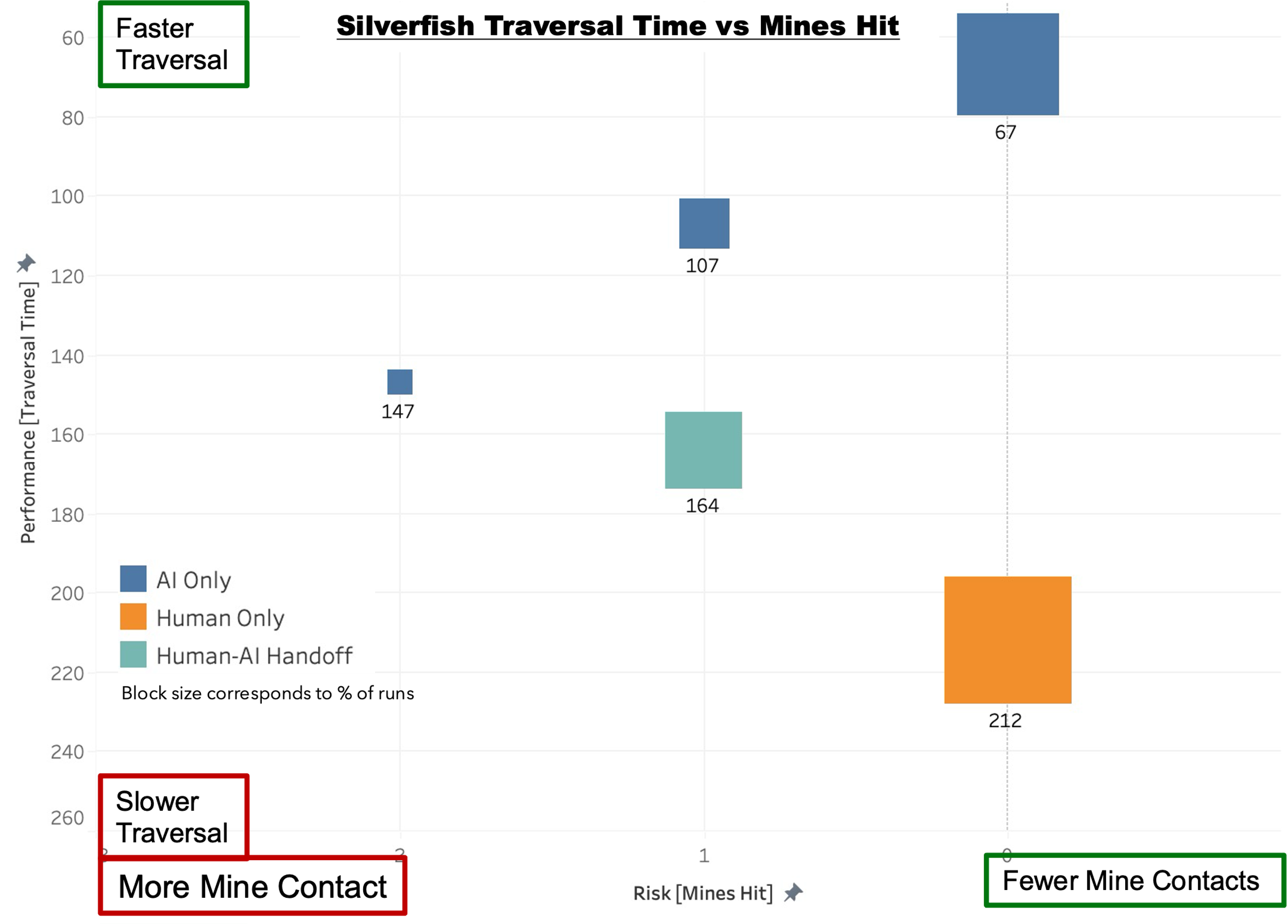


Figure 9: Human-AI Handoff Results

**5.0 Discussion**

The results for both the baseline architectures and the two human-in-the-loop architectures are shown in Table 5. When considering time as the metric of performance and mines hit as the metric of risk, it is clear that the human only and AI only baselines exist on opposite ends of the performance/risk tradeoff. The human only system has the worst performance by far, taking even longer than the AI only system when it hits two mines, but the human only system also avoided mines in 100% of runs. The AI only system had impressive performance, traversing the map quickly even with mines hit, but only avoided mines in 80% of runs.

Table 5: Results Summary

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Results | Human Only | AI Only | Human Approver | Human-AI Handoff |
| Avg Traversal Time (min) | **212** | **75.6** | **101.8** | 84.4 |
| Avg Mines Hit | 0.0 | 0.26 | 0.0 | 0.2 |

The human-in-the-loop system architectures were able to take advantage AI performance, while retaining human decision making to avoid riskier outcomes, but these two architectures also displayed that where and how humans are placed in-the-loop matters. The human approver architecture avoided mines 100% of the time, while being markedly faster than the human only system (though still much slower than the AI only system). Under the operating environment set in our simulation, human approvers were able to reallocate analysis to humans if confidence in the AI system was low. This enabled human approvers to avoid the riskiest behaviors of the AI system while still leveraging its comparative advantage in analysis speed.

The human-AI handoff architecture failed to provide as much risk avoidance as human only and human-approver architectures, only avoiding all mines in 80% of runs. In those 80% of runs, the time to traverse was the same as the AI only system avoiding all mines, meaning that in those 80% of runs, the human did not need to act since no rocky terrain was encountered. This behavior is expected since rocky terrain is only encountered at node B, which could avoided at the beginning node. The human-AI handoff architecture does however avoid the worst case outcome of hitting two mines since there is a mine present on rocky terrain after node B. In the 20% of runs where a mine is hit in the wooded terrain at the start of the map, the human-AI handoff architecture always avoids hitting the second mine, providing a clear advantage in risk avoidance over the AI only system. However, the increased risk avoidance comes at the cost of decreased performance, taking 164 minutes to traverse, even longer than the AI only system when it hits two mines.

Human-approver architecture provides a clear advantage over human only architecture with its equal risk profile and increased performance but comparing both human-in-the-loop architectures to the AI only architecture is more complex. It may be tempting to conclude that the AI only architecture is the superior architecture due to the clear performance advantage. Currently, the cost of hitting a mine is a fixed increase in traversal time. If the assumption of no lethal cost is changed, the calculus for how much risk to tradeoff for increased performance becomes far more difficult. Additionally, as human-AI handoff architecture shows, the tradeoffs are non-linear. This human-AI handoff architecture tradeoffs a large amount of performance for a relatively small amount of risk mitigation. We expect that these non-linear tradeoffs will continue to appear as more architectures are modeled. This non-linear tradeoff shows the importance and complexity of choosing the right human-AI system architecture for any system.

**6.0 Implications for Systems Engineering**

Systems engineering for AI-enabled systems poses new challenges that have not been addressed in the field and in the literature [32]. In the following subsections, we discuss the implications of our early findings have for systems engineering as a practice and describe the infrastructure and skills needed to adapt to these implications.

* 1. *What AI-Enabled Systems Mean for Systems Engineering*

The early results from this work showed that systems engineers need to consider the task allocation between humans and AI as a key function of system architecting. In just two simple examples, we showed that *how* the human is integrated into the system can greatly affect the performance and risk profile of a system. Systems engineering as a practice needs to expand the system boundary to consider both the human operators and the AI, as well as the interactions between them. While the field is well-suited to stakeholder analysis and testing of engineering artifacts, more work is needed to consider the interaction of operators and the technology they are expected to have oversight of. Without human/AI task allocation as a unit of analysis, designers may haphazardly place a human in-the-loop without sufficient attention to where and how they do so.

This work also highlights the importance of a system of systems approach to assurance and trust in AI-enabled systems. Research in academia, attention from government, and emphasis from industry has often focused on improving AI to ensure that algorithms are unbiased, transparent, and not overly susceptible to attack or distortion. While these aims are important on their own, they miss the larger ecosystem needed to manage AI-enabled systems. Optimally, human operators can mitigate worst-case failures of AI and enhance its capabilities through collaboration. Creating a system of assurance that centers humans is vital as we increasingly adopt flawed AI into our systems. While improving the algorithms themselves will improve performance and may increase trust, creating a robust system of assurance around AI-enabled systems can build trust in imperfect AI. The goal of trustworthy AI should not only be a trustworthy algorithm but also to create a system of trust and assurance around the algorithm.

*6.2 Systems Engineers Practices to Support AI-Enabled System Assurance*

The systems engineering field does not have proper testbeds to study the effects of placing a human-in-the-loop, and certainly not multiple humans in multiple loops. Current testbeds and metrics tend to focus on data, software and sometimes hardware. However, following on the above point, that the relevant unit of analysis is the HAI team/system, there is a need to adequately represent the crucial role that human operators play in implementing and managing AI risk. The simulation we are creating for this challenge is a simple version of what this shift in unit of analysis looks like and demonstrates the importance of varying the human role as part of the architecting trade space. It points to the types of models that systems engineers will need to create to test human-AI interaction. Models that explicitly consider this interaction can help designers understand the performance-risk tradeoffs of different system architectures. Designers can use these models to decide how much performance they are willing to sacrifice to mitigate risk. In addition to design, these models will be crucial to test and evaluation, helping ensure the human operators are able to provide the oversight and collaboration they are intended to. Many core systems models can be adapted for this purpose, but the additional skills required for this type of modeling are centered around human factors and human-centric design. Understanding these factors enable engineers to more accurately model how humans are expected to behave in-the-loop.

**7.0 Conclusion & Next Steps**

This early work has shown that where and how we place humans ‘in-the-loop’ affects the performance and risk of the systems they are expected to act within and have control over. With just two architectures modeled over a simple map, we have shown that there are clear and complex tradeoffs between the potential performance improvements of relying more heavily on AI-enabled technologies and the increased risk mitigation of doing so. The next step is to model each architecture identified in our framework. We will also continue to model architectures across different operating conditions, including varying AI and human performance. The framework provides descriptive design patterns, and the results from the simulation will help us create instructive heuristics of how the identified design patterns should be implemented under different operating conditions. Future findings will be used to develop heuristics to understand which architectures provide superior performance-risk profiles under different operating conditions.

**Works Cited**

[1] National Institute of Standards and Technology, “Artificial Intelligence Risk Management Framework,” U.S. Department of Commerce, Gaithersburg, MD, AI RMF NIST AI 100-1.0, Jan. 2023. [Online]. Available: https://doi.org/10.6028/NIST.AI.100-1

[2] J. Biden, *Executive Order on the Safe, Secure, and Trustworthy Development and Use of Artificial Intelligence*. 2023. Accessed: Nov. 17, 2023. [Online]. Available: https://www.whitehouse.gov/briefing-room/presidential-actions/2023/10/30/executive-order-on-the-safe-secure-and-trustworthy-development-and-use-of-artificial-intelligence/

[3] High-Level Expert Group on AI, “Independent High-Level Expert Group on Artificial Intelligence,” European Commission, Brussels, B-1049, Apr. 2019.

[4] China Academy of Information and Communications Technology and JD Explore Academy, “White Paper on Trustworthy Artificial Intelligence,” Ministry of Industry and Information Technology, Jul. 2021. [Online]. Available: https://cset.georgetown.edu/wp-content/uploads/t0390\_trustworthy\_AI\_EN.pdf

[5] Office of the Under Secretary of Defense for Policy, *Autonomy in Weapon Systems*, vol. 3009.09. 2023.

[6] M. H. Jarrahi, “Artificial intelligence and the future of work: Human-AI symbiosis in organizational decision making,” *Bus. Horiz.*, vol. 61, no. 4, pp. 577–586, Jul. 2018, doi: 10.1016/j.bushor.2018.03.007.

[7] A. B. Fontaine, “A Model for Human Tracking Behavior in a Closed Loop Control System,” Dissertation, The Ohio State University, Columbus, 1954.

[8] J. L. Decker, “The Human Pilot and the High-Speed Airplane,” *J. Aeronaut. Sci.*, vol. 23, no. 8, pp. 765–770, 1956, doi: 10.2514/8.3652.

[9] X. Wu, L. Xiao, Y. Sun, J. Zhang, T. Ma, and L. He, “A Survey of Human-in-the-loop for Machine Learning,” *Future Gener. Comput. Syst.*, vol. 135, pp. 364–381, Oct. 2022, doi: 10.1016/j.future.2022.05.014.

[10] C. Chai and G. Li, “Human-in-the-loop Techniques in Machine Learning,” *Bull. IEEE Comput. Soc. Tech. Comm. Data Eng.*, p. 16, Sep. 2020.

[11] X.-L. Meng, “Data Science and Engineering With Human in the Loop, Behind the Loop, and Above the Loop,” *Harv. Data Sci. Rev.*, vol. 5, no. 2, Apr. 2023, doi: 10.1162/99608f92.68a012eb.

[12] S. Munir, J. A. Stankovic, C.-J. M. Liang, and S. Lin, “Cyber Physical System Challenges for {Human-in-the-Loop} Control,” presented at the 8th International Workshop on Feedback Computing (Feedback Computing 13), 2013. Accessed: Aug. 03, 2023. [Online]. Available: https://www.usenix.org/conference/feedbackcomputing13/workshop-program/presentation/munir

[13] A. R. Lacher, L. Ren, D. R. Maroney, C. Schulenberg, and J. Daniels, “Dimensional Role Analysis: The Role of Humans and Automation for Increasingly Autonomous Aviation Systems,” in *2023 Integrated Communication, Navigation and Surveillance Conference (ICNS)*, Apr. 2023, pp. 1–6. doi: 10.1109/ICNS58246.2023.10124260.

[14] A. Williams, “Defining Autonomy in Systems: Challenges and Solutions,” in *Autonomous Systems: Issues for Defence Policymakers*, in Innovation in Capability Development, no. 2. , Norfolk: Headquarters Supreme Allied Commander of The North Atlantic Treaty Organization, 2015.

[15] M. R. Endsley and D. B. Kaber, “Level of automation effects on performance, situation awareness and workload in a dynamic control task,” *Ergonomics*, vol. 42, no. 3, pp. 462–492, Mar. 1999, doi: 10.1080/001401399185595.

[16] J. M. Beer, A. D. Fisk, and W. A. Rogers, “Toward a framework for levels of robot autonomy in human-robot interaction,” *J. Hum.-Robot Interact.*, vol. 3, no. 2, pp. 74–99, Jul. 2014, doi: 10.5898/JHRI.3.2.Beer.

[17] R. W. Proud, J. J. Hart, and R. B. Mrozinski, “Methods for Determining the Level of Autonomy to Design into a Human Spaceflight Vehicle: A Function Specific Approach,” National Aeronautics and Space Administration, Houston, Texas, Sep. 2003.

[18] J. Hawley, “Patriot Wars,” Center for a New American Secuirty, Washington, DC, Jan. 2017. [Online]. Available: https://www.cnas.org/publications/reports/patriot-wars

[19] National Transportation Safety Board, “Collision Between Vehicle Controlled by Developmental Automated Driving System and Pedestrian, Tempe, Arizona,” National Transportation Safety Board, Washington, D.C., NTSB/HAR-19/03 PB2019-101402, Mar. 2018.

[20] L. Save and B. Feuerberg, “Designing Human-Automation Interaction: a new level of Automation Taxonomy”.

[21] “About GitHub Copilot Individual,” GitHub Docs. Accessed: Apr. 12, 2024. [Online]. Available: https://docs.github.com/en/copilot/copilot-individual/about-github-copilot-individual

[22] “How Mars Perseverance Rover AutoNav Avoids a Boulder,” SpaceNews. Accessed: Jul. 25, 2024. [Online]. Available: http://spacenews.com/how-mars-perseverance-rover-autonav-avoids-a-boulder/

[23] NASA, “NASA’s Self-Driving Perseverance Mars Rover ‘Takes the Wheel,’” NASA Jet Propulsion Laboratory (JPL). Accessed: Jul. 25, 2024. [Online]. Available: https://www.jpl.nasa.gov/news/nasas-self-driving-perseverance-mars-rover-takes-the-wheel

[24] E. G. Dada, J. S. Bassi, H. Chiroma, S. M. Abdulhamid, A. O. Adetunmbi, and O. E. Ajibuwa, “Machine learning for email spam filtering: review, approaches and open research problems,” *Heliyon*, vol. 5, no. 6, p. e01802, Jun. 2019, doi: 10.1016/j.heliyon.2019.e01802.

[25] A. Lee, T. S. Baker, J. B. Bederson, and B. I. Rapoport, “Levels of autonomy in FDA-cleared surgical robots: a systematic review,” *Npj Digit. Med.*, vol. 7, no. 1, pp. 1–8, Apr. 2024, doi: 10.1038/s41746-024-01102-y.

[26] “Say hello to the new face of efficiency, security and safety,” U.S. Customs and Border Protection. Accessed: Apr. 12, 2024. [Online]. Available: https://www.cbp.gov/travel/biometrics

[27] C. Farivar, “Security robots expand across U.S., with few tangible results,” NBC News. Accessed: Jul. 30, 2024. [Online]. Available: https://www.nbcnews.com/business/business-news/security-robots-expand-across-u-s-few-tangible-results-n1272421

[28] M. Gil, M. Albert, J. Fons, and V. Pelechano, “Designing human-in-the-loop autonomous Cyber-Physical Systems,” *Int. J. Hum.-Comput. Stud.*, vol. 130, pp. 21–39, Oct. 2019, doi: 10.1016/j.ijhcs.2019.04.006.

[29] M. Gil, V. Pelechano, J. Fons, and M. Albert, “Designing the Human in the Loop of Self-Adaptive Systems,” in *Ubiquitous Computing and Ambient Intelligence*, C. R. García, P. Caballero-Gil, M. Burmester, and A. Quesada-Arencibia, Eds., Cham: Springer International Publishing, 2016, pp. 437–449. doi: 10.1007/978-3-319-48746-5\_45.

[30] M. Gil, M. Albert, J. Fons, and V. Pelechano, “Engineering human-in-the-loop interactions in cyber-physical systems,” *Inf. Softw. Technol.*, vol. 126, p. 106349, Oct. 2020, doi: 10.1016/j.infsof.2020.106349.

[31] J. E. Fischer, C. Greenhalgh, W. Jiang, S. D. Ramchurn, F. Wu, and T. Rodden, “In-the-loop or on-the-loop? Interactional arrangements to support team coordination with a planning agent,” *Concurr. Comput. Pract. Exp.*, vol. 33, no. 8, p. e4082, 2021, doi: 10.1002/cpe.4082.

[32] I. Ozkaya, “What Is Really Different in Engineering AI-Enabled Systems?,” *IEEE Softw.*, vol. 37, no. 4, pp. 3–6, Jul. 2020, doi: 10.1109/MS.2020.2993662.

1. Many parts of Sections 4 and 5 were adapted from material presented at SE4AI 2024 and submitted to CSER 2025 [↑](#footnote-ref-1)