Use of Simulated Mental Models and Real-time Planning for Human-Robot Interaction

Alagappan Swaminathan,*† Jiming Ren,*‡ Haris Miller,‡‡ Jack Kolb,††
Ye Zhao,§ Samuel Coogan,¶ and Karen M. Feigh‖
Georgia Institute of Technology, Atlanta, Georgia, 30332

This paper introduces a communication and planning framework to facilitate efficient state update information between an autonomous robotic system and a human operator under scenarios where continuous robotic monitoring is not available. The framework estimates the operator's mental model and uses the difference between the estimated mental model and the actual world state and robot plans to trigger selective communication updates to the operator. The framework is deployed in a simulation environment where a rapid task planning algorithm adapts robot plans to a dynamic operational environment in real time. The proposed framework aims to improve situational awareness and reduce cognitive load in contrast with baseline methods where communication is triggered by mission milestones, obstacles encountered, or periodically.

I. Introduction

Human command and control (C2) of swarms of robots is challenging for two reasons: *communication* and *interpretability*. Translating an operator's intent into machine-interpretable instructions is an open research question – when robots in a swarm behave unexpectedly, operators are uncertain if the robots correctly understood the operator's intent they were supposed to do or were incapable. This challenge is exacerbated when the operator is unable to monitor all swarm robots or has additional taskwork, causing reduced situation awareness about individual robot states and plans. Moreover, open questions exist on the explainability of swarms – how a framework can synthesize the robots' progress into communication that conveys important task information to the operator. In addition, we are interested in abstracting the micromanagement of swarms away from the human operator through the use of real-time task planning algorithms. Our objective is for the framework to route robots through high-level mission objectives, automatically replan robots as they encounter obstacles, and selectively update the operator when robot plans have meaningfully changed. Our work focuses on the system design and engineering of the framework, utilizing Bayesian mental models and temporal-logic-based planners to enable fluent operator-swarm communication and efficient self-organization.

II. Related Work

First presented by Johnson-Laird in the early 1980s [1], a mental model is an internal conceptual representation, a mental construct, or a long-lasting impression or understanding that captures the essential parts of the perceived structure of the external system. It is a cognitive framework that allows the individual to perceive, understand, and interpret the elements in the environment, allowing the individual to better predict the world [2]. Initial work on human-robot communication focused on understanding how people form mental models of a robot's factual knowledge. Lee et al. concluded that "participants estimated a robot's knowledge by extrapolating from their own knowledge and

This work is sponsored by Lockheed Martin Corporation with Steven Lincoln as Technical Monitor. The work is that of the authors and does not represent an official position of LMCO.

*Equal Contribution.
†PhD Student, Daniel Guggenheim School of Aerospace Engineering.
‡PhD Student, George Woodruff School of Mechanical Engineering.
‡‡MS Student, Daniel Guggenheim School of Aerospace Engineering.
††PhD Candidate, Daniel Guggenheim School of Aerospace Engineering.
§Assistant Professor, George Woodruff School of Mechanical Engineering.
¶Associate Professor, School of Electrical and Computer Engineering.
‖Professor, Associate Fellow, Daniel Guggenheim School of Aerospace Engineering.

the information available about the robot" [3]. According to Baker [4], humans naturally develop a "Theory of Mind" (ToM) when interacting with others, enabling them to model beliefs and intentions. The ability of humans to infer beliefs about the robot's state from low-level action information such as simple movements has been extensively studied [5]. Research by Ramaraj [6] explored human-robot interaction failures to determine the ability of mental models to resolve them. Utilizing the concept of Mental Models, Brooks and Szafron [7] focused on second-order mental models to infer human perception of robots's actions.

The significant impact of mental models in shared cognition and natural language, specifically in human-robot teams, has been critically studied [8]. Demir et al. concluded that teams that had natural language information exchange and shared mental models performed better than those without, in restricted conditions. The study also concluded that the perceived workload of the robot by the human was higher in the absence of a shared mental model. The challenge of communication from the robots and their ability to impact human perception, thereby their mental models, is an existing problem in human-robot interaction. The preliminary work presented by Rueben et al. [9] focused on enabling robots to "autonomously estimate and influence human beliefs about robot perceptual capabilities" [9] using a web-based game. Further work by Nikolaidis et al. [10, 11] extended this to manipulation capabilities, emphasizing the need for accurate human-robot shared models.

Current work on Shared Mental Models (SMM), specifically the review by Andrews et al. [12], focuses on the definition and application of SMMs in human-AI teaming, signifying its importance in enabling and improving team performance while proposing design considerations for enhancement of shared cognition. The importance of explicit communication [13] in human-robot teams is undeniable to build team cognition and trust. Following that, a computational framework demonstrating the positive impact of SMMs on coordination and team efficiency in simulated space robotics tasks is implemented by Gervits et al. [14, 15]. SMMs have also been proposed as a valid method for teaming with robots in high-stakes domains such as manufacturing and hazardous environments [16] to improve overall team performance.

Across disciplines, research converges on the idea that mental models are essential for effective teamwork by aligning objectives, subgoals, and roles. Julie A. Adams has significantly contributed to formalizing SMMs in human teams, emphasizing shared cognition as a cornerstone for improving team dynamics. Her work, alongside DeLoach and Scheutz [17], underlines how shared mental models enhance both team performance and cohesion in collaborative environments. Adams et al. [17] proposed a computational framework for integrating mental models into robotics, detailing the data structures and processes necessary to develop, update, and sustain these models. Meanwhile, Johnson [18] investigated the interplay between autonomy and interdependence in human-agent-robot teams, finding that collaboration often yields better outcomes than heightened autonomy, especially in complex tasks.

These studies collectively underscore the importance of mental models in HRI, providing insights into designing robots that can effectively understand and predict human behavior, thereby enhancing collaborative efficiency, communication, and user satisfaction.

III. Theoretical Framework

This work proposes a framework comprising two main components possessed by a swarm to address the key challenges in human-swarm awareness synchronization. The first component models the human operator's understanding of the location and progress of each robot using the information delivered by the swarm to the operator. We term this component a *Simulated Human Mental Model* (SHMM) as an approximation of the operator's situational awareness. In the second component, we use a task planning algorithm that anticipates a frequent request to update individual robot trajectories in response to unknown and dynamic environmental features. Instead of solving for a point-to-point mission plan, our planner articulates planning objectives beyond the scope of pointwise navigation, while also providing the capability to optimally revise the existing plan in real time. We couple the two components to have a more efficient human-robot interaction where the human's awareness is updated only when the robot's replanning causes sufficient deviation from the SHMM to merit communication.

Fig. 1 shows a system diagram of the overall framework, which includes a SHMM (the estimated mental model of the operator), Robot States & System Plan (the swarm's own understanding of its current state), and the communication and replanning blocks. In the communication block, both the SHMM and the System Plan are used to identify whether aspects of the operator's situational awareness should be repaired. The swarm continuously tracks these with the Robot's system plan and assesses the disparity between them using metrics such as position coordinates and total mission time. If the deviation between them surpasses a threshold, the swarm will inform the human operator on the true states of the robots to realign the operator's awareness to its own situational perception. We will further discuss the mental model

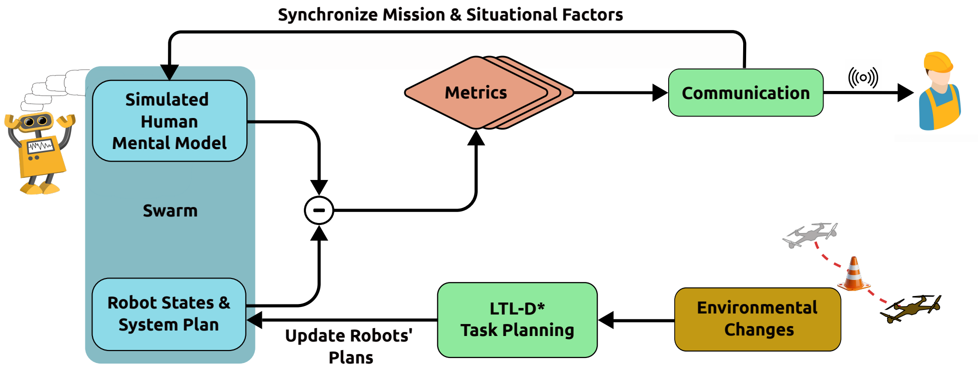


Fig. 1 The swarm tracks both the SHMM and the Robot States & System Plan and measures the disparity between them to evaluate the timing and attributes to synchronize the awareness of the operator. An optimal replanning block will respond to environmental changes and update robot states and future plans.

and communication in the following subsections.

A. Simulated Mental Model & Communication

Many different attributes can be utilized to trigger communication. In the scenario envisioned, we can identify 3 primary triggers: time, distance, and mission progress. Time indicates the amount of time that has elapsed given the mission progress and distance traveled compared to what was anticipated initially or based on previous updates. Distance indicates the amount of distance that has been traversed given the mission time and progress compared to initial estimates. Mission progress indicates the number of milestones accomplished compared to initial estimates.

The pseudocode presented in Algorithm 1 overviews the human-robot interaction framework and how it integrates the SHMM with the System Plan in order to trigger appropriate communication messages during robot swarm missions. The program begins by initializing the simulation environment and the mental model. The SHMM includes features such as robot positions, waypoints, mission objectives, and time elapsed towards the robot's mission objectives. The System Plan is initially set to be all nominal conditions when the mission starts. For each of the various missions and environments, the SHMM will be initialized as the best estimate (often the mean) of all simulation runs for the specific mission and environment for the initial plan and goal structure. It will be used to represent human perception and maintain situational awareness of the human operator.

The mission time is defined as the total time required for the robots to reach the mission objective. It is the sum of the time taken to get to the current position and the estimated time remaining to get to the goal position from the current position. For the SHMM, based on the previous simulation runs, the average mission time for the specific mission and environment is preset as the expected mission time. The 3 threshold values, low, medium, and high, are preset based on the observed mission time for the SHMM. The deviation metric of choice is selected by the human operator. Now, the communication system is connected to the simulation and replanning algorithm, and the robots begin their missions.

At each timestep, the System Plan gets populated with the feature values from the simulation, and for each of those time steps, the mission time is calculated and compared with the preset SHMM mission time value. To quantify the disparity between the mental models' mission time at each timestep, one of the following deviation metrics is utilized by the framework:

1) **Manhattan Distance (L1 norm):** The Manhattan Distance calculates the absolute sum of differences between the SHMM and System Plan mission times across all timesteps. It is useful for measuring the total deviation without emphasizing large outliers.

$$d_{\text{Manhattan}} = \sum_{i=1}^{n} |T_{\text{SHMM},i} - T_{\text{System Plan},i}|$$

where:
- $T_{\text{SHMM},i}$: Mission time from the SHMM at timestep i,
- $T_{\text{System Plan},i}$: Mission time from the System Plan at timestep i,
- n: Total number of timesteps.

Algorithm 1 Human-Robot Interaction Framework with Mental Models and Replanning

Initialize:
  Environment
  Simulated Human Mental Model (SHMM) $\leftarrow$ Initial Plan
  Robot Mental Model (System Plan) $\leftarrow$ Initial State and Initial Plan
  Communication Framework
Set Mission Goals and Parameters
Set **SHMM Mission Time** $\text{SHMM}_{\text{original}} \leftarrow$ Estimated time to execute initial plan
Set **Threshold values** based on the expected Mission Time of the current Mission
Select the deviation metric.
while Mission Not Complete **do**
  Receive System Plan data from simulation at current timestep - (Index, Pos-x, Pos-y, Time)
  Compute Mission Time for System Plan:
  $\text{System Plan}_{\text{Mission Time}} = \text{Time}_{\text{current}} + \text{Time}_{\text{remaining}}$
  Retrieve corresponding SHMM data for the current timestep
  Compute Deviation Metric to compare SHMM and System Plan:
  $\Delta = \text{deviation metric}(\text{SHMM}_{\text{Mission Time}}, \text{System Plan}_{\text{Mission Time}})$
  if $\Delta >$ Threshold **then**
    Trigger communication to Operator:
      Generate message comparing SHMM and System Plan
    Update SHMM using Bayesian logic:
    $\text{SHMM}_{\text{updated}} \leftarrow \text{deterministic\_bayesian\_update}(\text{SHMM}_{\text{original}}, \Delta, \text{Threshold}, \text{Uncertainty Factor})$
    Set $\text{SHMM}_{\text{current}} \leftarrow \text{SHMM}_{\text{updated}}$ ▷ Use updated SHMM for next comparisons
  end if
  Calculate Hybrid Distance Metric
  if Obstacle detected or Replanning needed (based on task feasibility and distance metrics) **then**
    Run LTL-D* to find the new optimal trajectories
  end if
  Update system plan and execute tasks
end while
End Mission

2) **Mean Absolute Error (MAE):** The Mean Absolute Error measures the average magnitude of deviations between the SHMM and System Plan mission times. It provides a simple measure of overall deviation.

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n} |T_{\text{SHMM},i} - T_{\text{System Plan},i}|$$

3) **Root Mean Squared Error (RMSE):** The RMSE calculates the square root of the average squared differences between the SHMM and System Plan mission times. It emphasizes larger deviations, making it sensitive to outliers.

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n} \left(T_{\text{SHMM},i} - T_{\text{System Plan},i}\right)^2}$$

4) **Correlation Coefficient (Pearson's r):** The Correlation Coefficient measures the linear relationship between SHMM and System Plan mission times. A value close to 1 indicates a strong positive correlation, while a value near -1 indicates a strong negative correlation.

$$r = \frac{\sum_{i=1}^{n} \left(T_{\text{SHMM},i} - \bar{T}_{\text{SHMM}}\right)\left(T_{\text{System Plan},i} - \bar{T}_{\text{System Plan}}\right)}{\sqrt{\sum_{i=1}^{n} \left(T_{\text{SHMM},i} - \bar{T}_{\text{SHMM}}\right)^2 \sum_{i=1}^{n} \left(T_{\text{System Plan},i} - \bar{T}_{\text{System Plan}}\right)^2}}$$

where:

- $\bar{T}_{\text{SHMM}}$: Mean mission time of SHMM,
- $\bar{T}_{\text{System Plan}}$: Mean mission time of System Plan.

5) **Relative Mean Absolute Error (RMAE):** The Relative Mean Absolute Error expresses the MAE as a fraction of the average mission time of the SHMM. It provides a normalized measure of deviation, allowing comparisons across different missions.

$$\text{RMAE} = \frac{\text{MAE}}{\text{mean}\,(T_{\text{SHMM}})}$$

When the deviation exceeds a threshold, a communication message is sent to the commander addressing the deviation. Subsequently, the SHMM is updated to reflect the commander's expected current state of perception. The update follows a deterministic Bayesian logic defined as:

$$\hat{T}_{\text{SHMM}} = \begin{cases} T_{\text{SHMM, original}} + \delta, & \text{if } |\delta| > \theta \\ T_{\text{SHMM, original}}, & \text{otherwise} \end{cases}$$

Where:

- $\hat{T}_{\text{SHMM}}$: Updated SHMM mission time,
- $T_{\text{SHMM, original}}$: Original SHMM mission time before update,
- $\delta = T_{\text{System Plan}} - T_{\text{SHMM}}$: Deviation between the System Plan and SHMM mission times,
- θ: Threshold value beyond which the update occurs.

The deterministic update logic is based on the idea that the system's prior belief or state (the SHMM mission time) is updated based on the observed deviation between the expected SHMM mission time value and the observed System Plan mission time value. This update only happens when the observed deviation exceeds the predefined threshold. Once the update happens, the current SHMM mission time value is replaced with the updated value and the robot continues on the path with no further communication to the human operator until the next update point.

Some other communication triggers could include those triggered by environmental changes and the robot's internal state. Some of the environmental changes can include variations in the terrain or expected weather conditions, and obstacles. It can be handled by integrating cameras and sensors like LiDAR to trigger communication. The robot's internal state could include metrics like the battery level and this can be handled by implementing a mechanism to have the robot consistently send data at every timestep to the framework, allowing it to use it as a metric to trigger communication when it falls below a preset threshold value.

B. Temporal-logic-based System Plan & Replanning

The task planning block consistently rewires existing trajectories to update the System Plan whenever dynamic environments trigger the discovery of new optimal solutions. In this work, replanning is achieved through a state-of-the-art revision algorithm named LTL-D* [19]. LTL-D* is designed for tasks based on linear temporal logic (LTL) [20], which is used as a description of a sequence of tasks assigned and to be performed. LTL provides rich semantics that extend beyond simple point-to-point navigation among regions of interest and are also capable of expressing sophisticated logics involving temporal ordering of tasks such as sequencing and coverage [21]. LTL specifications are ideal for assigning long-duration tasks to robots in a swarm and offer complexity that exceeds human tracking capacity, particularly when the swarm contains a significant population of robots. However, a major limitation of LTL-based planning lies in its inefficiency in real-time plan revisions when environments are subject to frequent changes or robots face unexpected disturbances. As robots' workspace and state space scale, the complexity of rewiring a feasible solution grows exponentially. Instead of building a new optimal plan from scratch which is the least efficient approach, we leverage LTL-D* to enhance the computational speed of replanning in our framework.

A Linear Temporal Logic (LTL) formula is constructed upon the atomic propositions $ap \in AP$ and the Boolean and temporal connectors in the grammar of $\varphi := \top |\ ap \mid \neg\varphi \mid \varphi \wedge \psi \mid \bigcirc \varphi \mid \varphi\ \mathcal{U}\ \psi$, where the Boolean operators denote "negation" ($\neg$), "conjunction" ($\wedge$), and the temporal modalities denote "next" ($\bigcirc$) and "until" ($\mathcal{U}$). Other useful temporal connecters including "eventually" ($\Diamond\varphi = \top\ \mathcal{U}\ \varphi$) and "always" ($\Box\varphi = \neg\Diamond\neg\varphi$) are also widely used in our task specifications given instructions and rules of the missions. Consider a simple example where a robot is assigned to start from location A and instructed to move toward location F after visiting any of the locations B, C, D, or E. The robot is also commanded to go to each location at least once. We can express the corresponding LTL specification for this mission as follows:

$$\begin{aligned}
\varphi_1 &= (A \rightarrow \Diamond B) \wedge \Box(B \rightarrow \bigcirc((\neg C \wedge \neg D \wedge \neg E)\mathcal{U}F) \\
\varphi_2 &= (A \rightarrow \Diamond C) \wedge \Box(C \rightarrow \bigcirc((\neg B \wedge \neg D \wedge \neg E)\mathcal{U}F) \\
\varphi_3 &= (A \rightarrow \Diamond D) \wedge \Box(D \rightarrow \bigcirc((\neg B \wedge \neg C \wedge \neg E)\mathcal{U}F) \\
\varphi_4 &= (A \rightarrow \Diamond E) \wedge \Box(E \rightarrow \bigcirc((\neg B \wedge \neg C \wedge \neg D)\mathcal{U}F)
\end{aligned}$$

where each specification describes the robot going to one of locations and not passing by any other spot before reaching location F. The overall specification of the task for each robot is a conjunction of all the tasks above:

$$\varphi = \varphi_1 \wedge \varphi_2 \wedge \varphi_3 \wedge \varphi_4$$

In our framework, the System Plans are executed on discretized abstractions of the collision-free workspace and the robot's operating states, both of which are modeled as weighted transition systems (WTSs). LTL specification can be encoded into a non-deterministic Büchi automaton (NBA) which provides a set of feasible task sequences that are acceptable to the NBA. A feasible solution that satisfies the LTL specification and is executable within the workspace and operating states can be explored from a product automaton, which is obtained by combining the WTSs and the NBA. A product automaton resembling a graph structure allows us to leverage LTL-D* algorithm, a modified version of D* Lite [22], to rewire a new optimal solution. For brevity, we omit the mathematical formulation and proof of optimality guarantee of LTL-D* and refer the interested readers to [19] for details.

During the robot's execution of the tasks, changes in a dynamic environment can be systematically reflected as modifications to the edges of the WTSs representing the environment. Replanning scenarios addressed in this paper assume that the task assignments to each robot remain consistent, only unexpected environmental changes would cause the original plan to fail. We classify these failures into two scenarios: (i) when the desired LTL specification can be fulfilled through replanning, and (ii) when the desired LTL specification is unachievable and can only be met in a "relaxed" manner. Relaxation is achieved through considering plant that would violate the initial LTL specifications. To resolve these failures, LTL-D* seeks an optimal replanning solution that minimizes violations of the desired task specifications. Our replanning performs at least 10 times faster than the replanning from scratch for the feasible scenario and around two orders of magnitude faster than replanning from scratch for infeasible scenario and guarantee the optimal solution [19].

C. Simulation Pipeline

To demonstrate the proposed framework, we developed a simulation pipeline integrating the mental models and replanning algorithm. Our simulator uses NVIDIA's Isaac Sim 4.2.0 [23] [24], which is a high-fidelity robotics simulation designed for validating robotics systems in virtual environments for human-in-the-loop study. The simulator has two objectives: to provide a realistic looking environment, and to enable testing the full framework without large-scale hardware experiments in varied environments. The simulation is configured to receive task plans from the replanning algorithm for each agent we want to simulate. We then simulate the agents in the environment sending the state information for each agent (e.g. agent position) to any external tools that might need the information. This also includes the status of the agent and in the case that an infeasible scenario is encountered, such information is shared with the replanning algorithm so that new instructions can be provided. Communication between the simulation pipeline and the framework uses ROS2. While the framework is agnostic to agent type, for our demonstrations we use unmanned aerial vehicles (UAVs). Fig. 2 provides an example where the task planning algorithm and the Isaac Sim simulation send results to an external WebServer and User Interface for testing the framework.

Simulation is the natural first step in evaluating this framework as it allows us to work through any technical issues with larger numbers of robots and environment complexities before adding in the difficulty of real-hardware. Figures 3, 4, and 5 are examples of potential environments created in Isaac Sim for such evaluation. When creating environments to test the overall framework, there are additional considerations related to the agents themselves. Firstly, we need to be able to generate an array of tasks that reflect the capabilities of the agents and which can also exploit the flaws of the agents to force replanning scenarios. For example, the warehouse environment in Fig. 4 has many loose objects in pathways which make it difficult for larger ground agents to traverse, but would be trivial for agents like UAVs to navigate over. Conversely, the maze-like environment in Fig. 3 provides fewer, more obvious pathways. In this case, the complexity of navigation tasks now comes from the ability to either open doors, which bipedal agents might excel at,

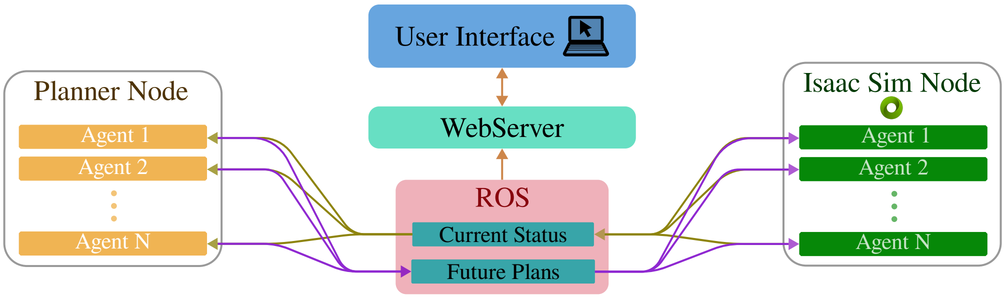


Fig. 2 Diagram overview of the framework message transmission pipeline. It consists of the Planner Node (task planning algorithm), the Isaac Sim Node (the simulation environment), the WebServer and the User Interface aiming to interact with human subjects to collect and analyze data. In the diagram, both the Current Status and Future Plans are data structure used under ROS communication framework.

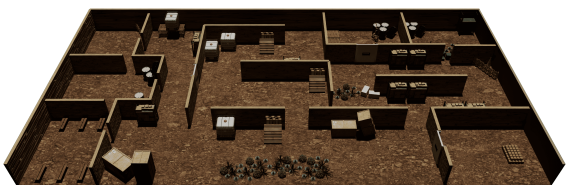


Fig. 3 Maze-like outdoors environment showing well-defined gridded pathways with sparse tasks in specified locations.

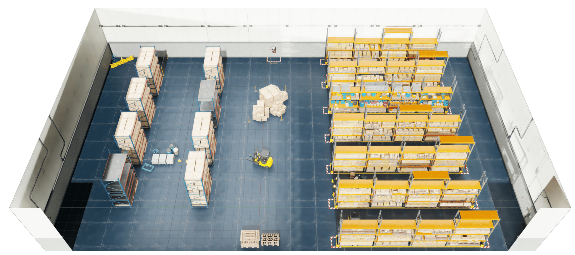


Fig. 4 Warehouse-like indoors environment showing semi-defined gridded pathways and loose tasks in random locations.

Fig. 5 Unstructured outdoors environment showing non-gridded pathways with loose objects and tasks.

or to find another path which may include more difficult terrain, which a quadcopter would be better suited for. We can also incorporate more dynamic tasks which are unexpected by the agents. In the example environments, falling objects like boxes have been incorporated which both forces replanning and adds potential complexity which tests the robustness of the SHMM. These dynamic tasks and changes are easily controlled either with the simulation directly or through communication from other tools connected via the simulation pipeline.

IV. Discussion

The proposed framework in human-swarm interaction combines real-time path replanning and cognitive mental model-based communication and leads to improved human situational awareness and selective communication of the essential details, thereby reducing the cognitive load on the operator. In this section we present preliminary results of the framework for two specific mission goals, addressing the theoretical advantages and challenges, followed by a description of the future direction for this project.

A. Framework Performance across Mission Goals

Figures 6 (a) and 6 (b) show the framework in action for a mission goal position or an end destination of (19,19) and (29,29), respectively. In the figures, the X-axis represents the simulation time step or index, and the Y-axis represents the mission time. The dotted blue lines indicate the SHMM values for mission time and orange lines indicate the System Plan mission time values. Each time, the System Plan mission time deviation exceeds the preset threshold, an update is sent to the commander with a communication message detailing specific mental model features comparing SHMM and System Plan array values for that specific timestep. The deviation is marked by the purple stars, which act as the update points on the plot. At each update point, the SHMM updates the mission time value to represent the updated human perception due to the communication message.

B. Theoretical Advantages and Challenges

The proposed framework addresses specific challenges in human-robot interaction associated with cognitive load, situational awareness, and information overload on the operator.

- **Reduce Operator Cognitive Load:** Communication to the operator only happens when major deviations occur and they are with regards to the overall mission time. This communication strategy significantly reduces cognitive fatigue and helps avoid information overload on the operator.
  Operators are only notified of significant deviations, avoiding information overload.
- **Improve Situational Awareness:** This communication message details specific aspects of the environment and the path corresponding to the robot's mission and sends it when the mission time deviation exceeds threshold. This communication message enables the human operator's perception to synchronize with the reality in the simulation, thereby improving situational awareness.

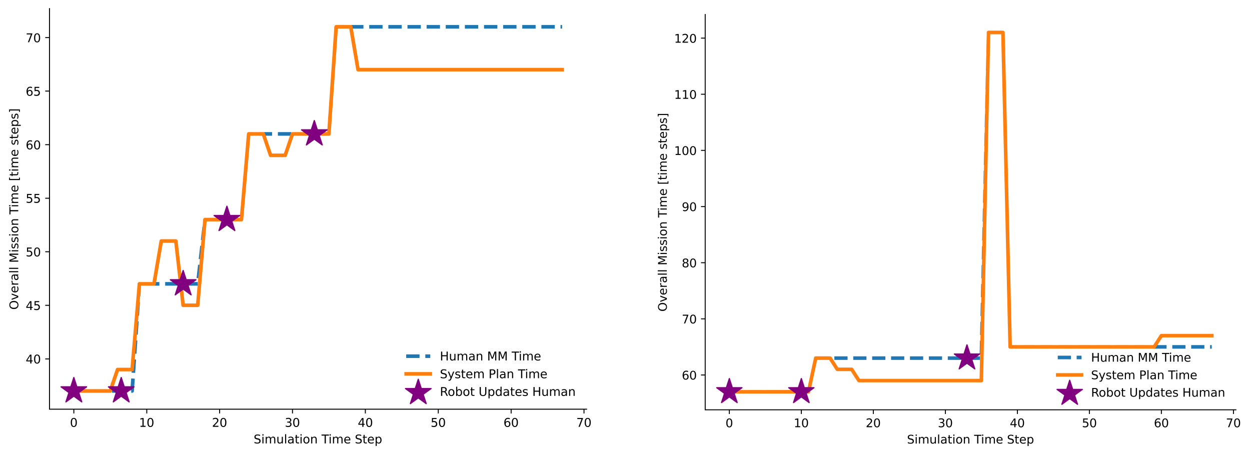
Some of the challenges associated with the framework include:

- **Threshold Calibration:** Utilization of preset threshold for specific environments and missions leads to excessive fine-tuning of the algorithm and makes it very mission-specific, thereby affecting scalability and affecting communication and human perception in unknown environments and missions.
- **Integration and Compute:** Integrating the simulation, replanning algorithm, and communication framework efficiently in real-time will likely introduce complexities associated with software compatibility and computational overhead. Ensuring optimal performance with ideal integration in dynamic environments will require a careful selection of software packages, high-performance computers, and optimization techniques. Managing a swarm of robots is also computationally expensive.
- **Deterministic Bayesian update:** The utilization of a simple deterministic Bayesian update logic could result in an oversimplification of complex decision-making processes where more advanced stochastic and non-stochastic probabilistic methods might function better.

C. Future Work

Future work aims to refine the framework and evaluate the proposed framework via user studies. A software pipeline integrating the replanning algorithm, simulation, and communication framework will be created to publish the results onto a User Interface (UI). The UI will then be used to design and conduct a user study comparing two scenarios: the baseline scenario, where the system communicates every slight deviation and updates the SHMM with the exact System Plan feature values, and the proposed system, which employs Bayesian update logic to communicate deviations only when the deviation exceeds the threshold value.

With the initial mental model provided, the situational awareness, cognitive load, decision-making accuracy, response time, and satisfaction level of the participants will be measured using standardized tools like the Situational Awareness Rating Technique (SART), the Situation Awareness Global Assessment Technique (SAGAT), and the NASA Task Load Index (NASA-TLX). Using a combination of quantitative statistical tests and qualitative analysis, we will determine the ideal communication strategy and effectiveness of the proposed communication framework to maintain effective situational awareness while minimizing the cognitive load on the commander.



(a) Performance for the 19x19 Map

(b) Performance for the 29x29 Map

Fig. 6 Framework Performance Across Mission Goals: (a) shows the performance for a mission goal at (19,19), and (b) illustrates the performance for a goal at (29,29). The X-axis represents simulation time steps, and the Y-axis shows mission time. The dotted blue lines denote SHMM mission time values, the orange lines show System Plan mission time values, and the purple stars mark deviations where the commander receives updates.

Additionally, this UI will allow for more intuitive interactions for the operator. It will provide features to control the mental model disparity threshold, and it will present communication messages along with the actual simulation video. It will also potentially have features to take input from the operator and serve as an interface for bidirectional communication.

Refinements and Improvements

Refining and improving the existing framework will be a primary area of focus for our future works on the framework. Some of the potential improvements include:

- **Increased features:** Enhance the System Plan and SHMM to accommodate more features associated with the mission and in multiple modes, allowing for more dimensions of improvement for the framework.
- **Dynamic threshold determination:** Utilization of a dynamic threshold determined by the observed disparity will allow the framework to be adaptable and scalable to varying missions and environments.
- **Composite harmonic score:** Instead of one specific deviation metric, a composite harmonic score of a combination of all the metrics will allow a more comprehensive measure of the disparity between the mental models. It will also allow the human operator to set weights to specific individual metrics based on mission-specific parameters and operator preferences, thereby improving usability and flexibility.
- **Complex Bayesian models:** Instead of the deterministic Bayesian update logic used here, a variety of stochastic and non-stochastic Bayesian logic models could result in better performance of the framework and reduce potential confusion in human perception.
- **Utilization of LLMs for communication:** Instead of hard-coded communication systems, Large Language Models (LLMs) could be utilized to generate more realistic, natural, and human-readable context-aware messages personalized for the operator's requirements and preferences in communication, thereby enhancing usability and interoperability. It can also account for unforeseen situations and environments and account for the experience level of the current human operator or commander.
- **Task Coordination between robots:** Our task planning algorithms is primarily designed to handle task assignments for individual robots. However, many robotic swarms are capable of performing coordinated tasks that require collaboration among multiple robots. These coordinated tasks often involve synchronized actions or shared goals that cannot be achieved by a single robot operating in isolation. For instance, two or more robots may need to jointly transport a heavy object, execute a complex assembly process, or perform area coverage tasks that require dynamic communication and role reassignment.
  Such coordinated behaviors demand a higher level of task planning complexity, as the system must account for inter-robot dependencies, timing constraints, and the spatial relationships between robots. Incorporating these capabilities into task planning algorithms would significantly enhance the effectiveness and versatility of swarm robotics in dynamic and unpredictable environments. This necessitates the development of algorithms that can

seamlessly integrate individual task assignments with multi-robot coordination strategies while maintaining computational efficiency and scalability.

- **Simulation Improvements**
There are three main areas where we would like to improve the simulation pipeline in the future. The first area is to improve the capabilities and types of agents we can test in the simulation environment. At the moment, agents can only perform path base tasks regardless of agent type. This means that the range of issues that can cause replanning is restricted to issues that limit the pathways available to certain agents and the difficulty of traversing such pathways. We would like to expand these tasks to include things such as picking up objects or interacting more with the environment to give a wider array of potential issues for testing. The second area we would like to improve is the range of information we can get from the system. At the moment, we mainly get agent states (e.g. position, velocity, etc.) and when the agent either completes a task or needs replanning, also known as the agent status. We would like to expand this to also include other information which might be useful in testing the SHMM such as sensor data, agent visual data (e.g. cameras), and simulated status information (e.g. remaining battery). Finally, we would like to greatly increase the number of testing environments available to us, which includes changing the types of environments represented and the features in each environment. Type of environments describes both how they look visually, such as outdoor versus indoor environments, and the degree to which pathways in the environment follow a rectangular grid. Additionally, features can include potential obstacles and the available tasks encountered in the environment. We intend to expand on both types of environments and features to give us a larger array of scenarios to further test the robustness of the proposed framework.

V. Conclusion

The proposed human-robot interaction framework marks a step towards the integration of cognitive mental models and real-time trajectory replanning in swarm robot missions. Despite challenges with integration, the framework's potential to maintain effective situational awareness and reduce cognitive load is promising. Further work on the framework will explore system refinements, testing its capability at scale in dynamic environments, validating it's capabilities whilst addressing existing limitations to create a more robust and real-world human-robot interaction. The framework's modular design and metrics-based deviation assessment allow seamless integration into autonomy software engineering pipelines whilst ensuring flexibility across varying missions and uncertain and dynamic environments. This framework sets a precedent for future developments in adaptive decision-making and autonomy software engineering for swarm robots, and presents a theoretical foundation for cognitive-mental model-based communication systems.

References

[1] Johnson-Laird, P., "Mental models in cognitive science," *Cognitive Science*, Vol. 4, No. 1, 1981, pp. 71–115.

[2] Johnson-Laird, P. N., *Mental models: towards a cognitive science of language, inference, and consciousness*, 1983.

[3] Lee, S.-l., Lau, I. Y.-m., Kiesler, S., and Chiu, C.-Y., "Human Mental Models of Humanoid Robots," *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*, IEEE, 2005, pp. 2767–2772. https://doi.org/10.1109/ROBOT.2005.1570532.

[4] Baker, C. L., Jara-Ettinger, J., Saxe, R., and Tenenbaum, J. B., "Rational Quantitative Attribution of Beliefs, Desires and Percepts in Human Mentalizing," *Nature Human Behaviour*, Vol. 1, No. 4, 2017, p. 0064. https://doi.org/10.1038/s41562-017-0064.

[5] Bartlett, M., Edmunds, C., Belpaeme, T., Thill, S., and Lemaignan, S., "What Can You See? Identifying Cues on Internal States from the Movements of Natural Social Interactions," *Frontiers in Robotics and AI*, Vol. 6, 2019, p. 49. https://doi.org/10.3389/frobt.2019.00049.

[6] Ramaraj, P., "Robots that Help Humans Build Better Mental Models of Robots," *Companion of the 2021 ACM/IEEE International Conference on Human-Robot Interaction*, Association for Computing Machinery, 2021, pp. 595–597.

[7] Brooks, C., and Szafir, D., "Building Second-Order Mental Models for Human-Robot Interaction," *arXiv*, 2019. https://doi.org/10.48550/arxiv.1909.06508, URL https://arxiv.org/abs/1909.06508.

[8] Demir, M., McNeese, N. J., and Cooke, N. J., "Understanding human-robot teams in light of all-human teams: Aspects of team interaction and shared cognition," *International Journal of Human-Computer Studies*, Vol. 140, 2020, p. 102436.

[9] Rueben, M., Matarić, M. J., Rothberg, E., and Tang, M., "Estimating and Influencing User Mental Models of a Robot's Perceptual Capabilities: Initial Development and Pilot Study," *Companion of the 2020 ACM/IEEE International Conference on Human-Robot Interaction*, Association for Computing Machinery, 2020, pp. 418–420.

[10] Nikolaidis, S., Forlizzi, J., Hsu, D., Shah, J., and Srinivasa, S., "Mathematical Models of Adaptation in Human-Robot Collaboration," *arXiv*, 2017. URL https://arxiv.org/abs/1707.02586.

[11] Nikolaidis, S., Nath, S., Procaccia, A. D., and Srinivasa, S., "Game-Theoretic Modeling of Human Adaptation in Human-Robot Collaboration," *12th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, IEEE, 2017, pp. 323–331. https://doi.org/10.1145/2909824.3020252.

[12] Andrews, R. W., Lilly, J. M., Srivastava, D., and Feigh, K. M., "The role of shared mental models in human-AI teams: a theoretical review," *Theoretical Issues in Ergonomics Science*, Vol. 24, No. 2, 2022, pp. 129–175.

[13] Schelble, B. G., Flathmann, C., McNeese, N. J., Freeman, G., and Mallick, R., "Let's Think Together! Assessing Shared Mental Models, Performance, and Trust in Human-Agent Teams," *Proceedings of the ACM on Human-Computer Interaction*, Vol. 6, No. GROUP, 2022, pp. 13:1–13:29.

[14] Gervits, F., Fong, T. W., and Scheutz, M., "Shared mental models to support distributed Human-Robot teaming in space," *2018 AIAA SPACE and Astronautics Forum and Exposition*, 2018. URL https://doi.org/10.2514/6.2018-5340.

[15] Gervits, F., Thurston, D., Thielstrom, R., Fong, T., Pham, Q., and Scheutz, M., "Toward Genuine Robot Teammates: Improving Human-Robot Team Performance Using Robot Shared Mental Models," *Proceedings of the 19th International Conference on Autonomous Agents and MultiAgent Systems*, International Foundation for Autonomous Agents and Multiagent Systems, 2020, pp. 429–437.

[16] Nikolaidis, S., and Shah, J., "Human-Robot Teaming using Shared Mental Models," *Proceedings of the 7th Annual ACM/IEEE International Conference on Human-Robot Interaction Workshop on Human-Agent-Robot Teamwork (HART)*, Massachusetts Institute of Technology, 2012. URL http://stefanosnikolaidis.net/papers/HART2012_nikol_shah.pdf.

[17] Scheutz, M., DeLoach, S. A., and Adams, J. A., "A Framework for Developing and Using Shared Mental Models in Human-Agent Teams," *Journal of Cognitive Engineering and Decision Making*, Vol. 11, No. 3, 2017, pp. 203–224. https://doi.org/10.1177/1555343416682891.

[18] Johnson, M., Bradshaw, J. M., Feltovich, P., Jonker, C., Van Riemsdijk, B., and Sierhuis, M., "Autonomy and Interdependence in Human-Agent-Robot Teams," *IEEE Intelligent Systems*, Vol. 27, No. 2, 2012, pp. 43–51. https://doi.org/10.1109/MIS.2012.43.

[19] Ren, J., Miller, H., Feigh, K. M., Coogan, S., and Zhao, Y., "LTL-D*: Incrementally Optimal Replanning for Feasible and Infeasible Tasks in Linear Temporal Logic Specifications," *arXiv preprint arXiv:2404.01219*, 2024.

[20] Baier, C., and Katoen, J.-P., *Principles of model checking*, MIT press, 2008.

[21] Fainekos, G., Kress-Gazit, H., and Pappas, G., "Temporal Logic Motion Planning for Mobile Robots," *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*, 2005, pp. 2020–2025. https://doi.org/10.1109/ROBOT.2005.1570410.

[22] Koenig, S., and Likhachev, M., "Fast replanning for navigation in unknown terrain," *IEEE transactions on robotics*, Vol. 21, No. 3, 2005, pp. 354–363.

[23] NVIDIA, "Isaac Sim," , 2024. URL https://developer.nvidia.com/isaac/sim.

[24] Makoviychuk, V., Wawrzyniak, L., Guo, Y., Lu, M., Storey, K., Macklin, M., Hoeller, D., Rudin, N., Allshire, A., Handa, A., and State, G., "Isaac Gym: High Performance GPU-Based Physics Simulation For Robot Learning," *CoRR*, Vol. abs/2108.10470, 2021. URL https://arxiv.org/abs/2108.10470.