Shared Mental Models to Support Distributed Human-Robot Teaming in Space

Felix Gervits
Tufts University, Medford, MA, 02155, USA

Terrence Fong
NASA Ames Research Center, Mountain View, CA, 94035, USA

Matthias Scheutz
Tufts University, Medford, MA, 02155, USA

Effective human-robot interaction (HRI) is a critical requirement for current and future space operations. However, given the limitations of autonomous technologies, robots are not yet capable of coordinating with human crew as peers under real-world mission constraints. Due to the complexity inherent in space robotics operations, it is crucial that robots are able to coordinate the various aspects of human-robot teaming and the task at hand. In this paper, we extend our Shared Mental Model (SMM) interaction framework to show how it can be used to overcome some of the challenges inherent in distributed HRI and facilitate coordination in space robotics teams. Since this framework has not yet been implemented in real systems, we conducted an exploratory study to identify potential benefits that the SMM mechanisms can afford in a task domain involving simulated free-flying robot assistants on a spacecraft. We found that the SMM framework offers advantages to task performance and team efficiency due to its support of shared knowledge representations, however these benefits do not seem to reduce workload or improve other subjective measures. The significance of these findings for future space operations is discussed, as are directions for future research.

I. Introduction

Robots have a critical role to fill in supporting humans in current and future space operations. They can perform dangerous, tedious, and long-duration tasks that would otherwise be prohibitive for human crews. Robots are also much cheaper to deploy than astronauts, and so are more practical for a variety of missions such as exploration of small bodies (e.g., asteroids) and planetary surfaces. For these reasons, robots are frequently utilized in space exploration to carry out a variety of tasks, including scientific field work, reconnaissance, site preparation, structure assembly, and many others[1].

A. Robots to support human teams

Interaction between humans and robots is also critical for many space operations. This interaction can range from co-located teams performing joint work to spatially separated teams that operate at different time scales and that have no physical contact. One key distinction here is the degree to which agents depend on the actions of their teammates in order to perform their own actions - a type of interdependent relationship[2]. Tasks with a high degree of interdependence are those in which a human and robot collaborate in real-time on some problem, e.g., a joint repair task. In contrast, a task can be considered more “independent" [*] if agents can perform their part of the task without waiting for their teammates.

Both interdependent and independent interactions are important moving forward, but the former involve significantly more technical challenges to implement in robotic systems. The biggest challenge is operating at human time scales. Human time is perhaps the most valuable resource in space, so it is highly important that it not be wasted waiting for a robot to act. Because few if any systems exist that can operate at human time scales under real-world mission constraints,

[*]Note that true independence may not even be considered *interaction* by some definitions. We are using the term to refer to tasks in which the interdependent human and robot actions are separated in time.

the current research focus[†] has been on utilizing robots in distributed interactions in which the robot performs their task before or after human activity[4, 5].

Previous work has addressed this challenge, seeking to evaluate the utility of distributed HRI for supplementing human exploration in the context of robot reconnaissance and follow-up activities[6]. In one scenario, advanced robot scouting was found to improve mission objectives in a simulated lunar EVA mission. Data gathered from the preliminary robot mission was integrated with orbital sensing data to establish new traverses for future human crew missions, ultimately enhancing scientific exploration. A related experiment demonstrated the utility of robot follow-up after a simulated human mission on the moon. It was found that robot follow-up was useful for exploring additional locations that the humans did not have time for, as well as for collecting additional data. In sum, these experiments show that HRI can be useful to support human exploration when the actions of each agent are separated in time, and thus require less interdependence.

B. Robots to support robot teams

While the above studies demonstrate the successful use of robots to supplement human exploration objectives, it is important to extend these principles to other domains and address the new challenges that arise. A particularly relevant and under-explored domain is one in which robots support other robots aboard a spacecraft, such as the International Space Station (ISS) or Deep Space Gateway (DSG)[7]. The DSG is envisioned to only be manned by astronauts for 30-60 days of the year, so it is important that on-board robots can maintain the systems and repair any faults that arise in the absence of human activity. Since different robots have different capabilities, they must coordinate their actions to support the overall team. For example, Astrobee[8] is effective for navigating the spacecraft and performing maintenance duties, while Robonaut 2[9] is better suited for actions that involve manual dexterity such as repair tasks. Computational approaches for managing the interaction challenges of such teams are underexplored (but see [10, 11], yet they are crucial for future space robotics missions.

The ability of robots to perform interdependent tasks in sequence opens up new interaction possibilities, but also many unique challenges. One such challenge is a much shorter time between interaction. While the interaction is still separated in time, the period between initial and follow-up activity is significantly reduced from the above human-robot lunar scenarios (from weeks/months to minutes/hours). This requires that any tasks completed or information gained by the initial robot team needs to be immediately transmitted to the follow-up team. It also means that the initial team will need to represent what aspects of the task are relevant for the follow-up team, and ensure that this information is transmitted effectively. For example, if a robot's task is to locate tools for a future repair crew to use in making repairs, the robot will need to know which tools need to be found, and which other pre-conditions need to be met for the future team to be able to perform their task. Overcoming all of these challenges requires a way to support coordination among multiple agents (human and robot) that operate across multiple spatial ranges, time scales, and interaction modalities and that share a common goal.

C. Shared Mental Models

The concept of shared mental models (SMMs) has been widely explored in the Human Factors and Organizational Psychology literature as a means for managing complex teams. An SMM is a distributed construct that contains the knowledge and functions necessary for successful teamwork, including: monitoring goal and task states, evaluating performance of teammates, inferring beliefs and intentions, tracking task focus, and adapting behavior as a result of this knowledge[12]. Previous studies in humans have shown SMMs to be critical for coordinating team activities in varied domains such as software development[13], flight crew planning[14–16], and product design[17].

As SMMs have been implicated in successful human teamwork, it is theorized that they can facilitate interaction and coordination in mixed human-agent teams as well. Several studies have shown promising results in software systems[12, 18] and more recently in robots[19–21]. However, these studies often focused on a single aspect of the SMM (e.g., predicting workload[22]) and were not grounded in a comprehensive formal or computational framework.

D. Present work: A framework to support distributed team interaction

Recently, we have developed the first integrated formal and computational framework for SMMs[23]. This framework supports the structures and functions by which SMMs can be realized in robotic architectures. In this paper,

[†]Note that operating the Mars rovers (Opportunity and Curiosity) involves human-robot activity occurring in parallel, albeit over lengthy temporal delays. See [3]

we extend our SMM framework (see section section II below) to a space robotics domain, and show how it can address many of the coordination challenges that arise. We also extend our prior work [24] on evaluation paradigms for space robotics teams to include autonomous agents that are supervised by a remote human operator.

Perhaps the most important question for applying our framework to human-robot teams is whether the framework mechanisms actually provide a benefit to the teams that use them. For example, does it serve to improve task performance, team efficiency, or operator workload? To examine these questions, we conducted an empirical study in which we simulated the SMM capabilities in virtual agents. For this initial study, we developed a simulated environment which allowed us to focus on specific elements of the SMM, namely the ability of the agents to share information and adapt behavior in a team-oriented manner. Of central importance was how these mechanisms affect the operator and team performance. As our interest was in evaluating how the user responds to the SMM, the agents were not integrated in a robotic architecture, but instead were programmed to behave autonomously as if they had the mechanisms of interest. By simulating the agents in this way, it allowed us to examine performance in a more complex domain, without the limitations of using actual robots (e.g., limited autonomy), or the overhead associated with a large-scale distributed robotic architecture. Thus, the preliminary study serves as an exploration to identify if the SMM approach is worth pursuing, and does not constitute an evaluation of the framework. Future work will need to implement the symbolic framework in a robotic architecture in order to truly evaluate its efficacy.

In the following sections we describe the SMM framework in more detail and show how it was applied to our task domain. Next, we introduce our exploratory study, including the design and goals. We then discuss the results of our user study, and implications of this approach on current and future space robotics operations.

II. Shared Mental Model Framework

To address the numerous interaction challenges posed by space environments requires an approach that supports coordination in heterogeneous teams in a dynamic and flexible way. It also needs to handle teams that are distributed in space, communicate over variable time ranges, and that operate with varying control modes. Such an approach can be realized through a set of interaction mechanisms integrated within an SMM framework.

A. Overview of framework

Task Model		**Team Model**	
Equipment	**Task**	**Team Interaction**	**Team (Teammates')**
Equipment Functioning	Task Procedures	Roles/Responsibilities	Knowledge
Operating Procedures	Likely Contingencies	Information Sources	Skills
Equipment/System Limitations	Likely Scenarios	Role Interdependencies	Attitudes
Likely Failures	Task Component Relationships	Communication Channels	Preferences
	Environmental Constraints	Interaction Patterns	Performance History
	Task Strategies	Information Flow	Tendencies

Fig. 1 Components of a Shared Mental Model.

In our framework[23], SMMs are represented as knowledge structures with two main components: the Task model - comprised of the Equipment and Task components, and the Team model - comprised of the Team Interaction and Team Member components (see Fig. 1). The computational framework incorporates these components, and stores the information in data structures that are operated on by various computational processes. These processes serve to update the SMM state and also adapt agent behavior and make predictions about changing Team and Task states. For example, if an SMM agent detects a weak communication signal, it might predict more errors or delays with its speech recognition, and could adapt to exhibit more clarification requests in its dialogue behavior. Importantly, while each agent has their own internal SMM, the information contained therein is synchronized between all robotic agents on the team such that they all have access to an updated knowledge base of team functioning. This feature directly addresses the problem of multiple robots distributed across space and time, because each robot will now have updated information about the knowledge and status of every other teammate. In sum, the SMM framework is a step in the direction of enabling robots to serve as tools to aid human exploration and eventually genuine partners to human crew.

B. Supervised emergency maintenance scenario

Here we describe the basic scenario that we are using. In subsection II.C below we describe how it was formalized in our framework, and how we created a correspondence between this formalization and the autonomous policies of the robots in our exploratory study.

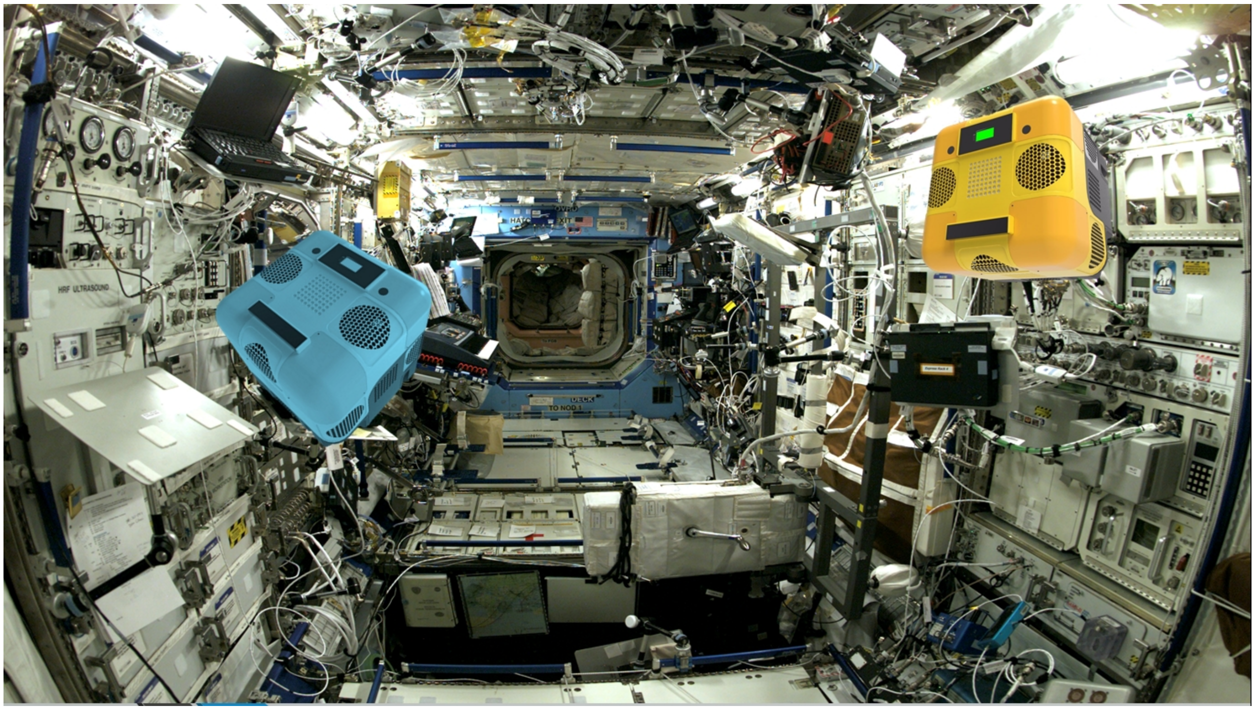

Fig. 2 Concept image of the proposed scenario involving two Astrobee free-flyers aboard a spacecraft.

Our scenario is modeled after a potential use case for robots on the ISS/DSG[8, 25] and involves a ground-based human operator and two free-flying robot assistants (*R1*, *R2*) on a spacecraft orbiting the Moon (see Fig. 2). In this way the team is distributed across spatial ranges and (to some extent) time scales. The spacecraft is currently unmanned, and the on-board robots maintain the systems in the absence of human crew by performing routine fault inspection and repair duties.

In the scenario, there is an emergency and several wall panels on the spacecraft have short-circuited due to an electrical failure. The human-robot team must perform two tasks to handle the situation: 1) they must locate and shut off the damaged panels, and 2) they must identify the location of several tools throughout the spacecraft that are necessary for repairs. These two tasks are needed for a follow-up robot repair crew which will make the actual repairs, and which relies on having the panels shut off and knowing the tool locations. The decision to separate the repair task into a preliminary inspection and follow-up repair phase is due to the capabilities of the robots involved. Certain robots such as Astrobee are designed to navigate a spacecraft environment and inspect faults, whereas robots like Robonaut 2 are designed for more dextrous behavior such as repairing panels. While we do not include this follow-up task in our study, our central concern is how the initial team can best prepare for this follow-up work. Ultimately, the goal is for the robots and human operator to work together to navigate the environment, identify the location of all the tools, and shut off all damaged panels. More details of how this scenario was implemented in our study are provided in section III below.

C. Framework Implementation

We represent the task domain using the representational primitives from the formal SMM framework introduced in [23]. These primitives are represented as logical predicates that capture relevant features of the task and of the agents. Since the present domain is deterministic, we do not include probability distributions for each predicate, but the framework allows for these to be added as arguments for stochastic domains. While the primitives themselves are domain-independent, several of the arguments are tailored to the specific task domain. Information from Fig. 1 was populated, including relevant equipment and operating procedures (e.g., map of the spacecraft, equipment functionality, etc.), task procedures (such as how to search a toolbox), team structure (including various roles, responsibilities, and interdependencies), communication channels (including limitations due to LOS), and knowledge of the team, including goals, task progress, and task focus for each agent. We also introduce a set of general update and control principles to allow the agents to update their representations and to leverage the SMM knowledge to adapt behavior. Finally, we created a mapping between the SMM representations and our simulated task domain. In general, this involved converting the logical expressions and rules described in this section into the equivalent classes, variables, and algorithms of the simulation.

1. Domain knowledge

First, we define the basic elements of the task domain, including the agents and environment. These include:

Agents: {*R1, R2, O*}

Object Types: {*GUI*, *camera* (R1/R2), *sensor* (R1/R2), *gripper* (R1/R2), *toolbox* (#'s 1-12), *panel* (#'s 1-20), *room* (1a, 1b, 2, 3, 4, 5), *tool* (wrench, screwdriver, hammer, nails, drill, screws), *signal*}

Agent/Object Properties: {*isBusy(A), isOccupied(room), movingTo(A,room), location(A,room) isScanned(A,room), isSearched(A,toolbox), isFound(A,tool) isRevealed(A,panel), isShutoff(A,panel), isUpdated(GUI), isWorking(A,camera), isWorking(A,gripper), isWorking(A,sensor), isLost(signal)*}

Activities: {*navigate(room), scan-room, shutoff-panel, search-toolbox, reboot(<Equipment>), update-GUI, change-roles, supervise*}

There are three agents in the domain, including two robots (*R1* and *R2*) and the human operator (*O*). The object types represent elements of the environment such as the *rooms*, *wall panels*, *toolboxes*, etc. as well as equipment belonging to each robot, such as the *camera*, *gripper*, and *sensor*. Properties of the agents and objects are listed, and include the different states that objects (e.g., *isOccupied(room))* and agents (e.g., *isBusy(A))* can be in. The set of *Activities* represents the different actions available to the agents in the task. These include navigating from room to room, rebooting equipment after failure, changing roles, etc. Finally, the *isLost(signal)* property represents the communication link between the operator and the two robots. During LOS this flag is set to false, and the agents cannot communicate for the duration of the LOS period.

Update processes are used to update the SMM state based on new information, and involve rules to track various aspects of the task as well as to resolve inconsistencies in synchronizing multiple agents' internal SMMs. Control processes are used to adapt the agents' behavior accordingly. An example of an update process in this domain is the following rule, which holds that if a robot is in a room, then that room is occupied:

$$located(A,room) \implies isOccupied(room)$$

Another rule involves updating the status of the environmental representations as a result of agent actions. For example, if an agent shuts off a panel and that panel has not previously been shut off, then that panel is now shut off:

$$do(A,shutoff-panel) \land \neg isShutoff(panel) \implies isShutoff(panel)$$

An example of a control process is a rule which requires that agents update the GUI (i.e., inform the operator) after some action has been completed. If this action occurred during a LOS period, then the update has to be delayed until the signal returns. This is represented in the following way, using *GOAL* and *ITK* predicates (defined in subsubsection II.C.3 and subsubsection II.C.5). The *ITK* in this case triggers a process which stores ϕ in a queue and updates the task log with this message once the signal returns:

$$ACHIEVED(A,\phi) \land isLost(signal) \implies ITK(A,\phi)$$

Below, we extend the task domain knowledge to allow the agents to represent various aspects of the Team and Task state. These are split up into the following sets, including *agent capabilities*, *agent and task states*, *plans and autonomy*, *obligations and norms*, and *functional roles of agents in teams*. Though this domain knowledge will be initially provided, the agents will be able to adapt to new information and update their Task and Team models accordingly, as well as adjust behavior. This is done using additional update and control processes defined in each section below.

2. Agent capabilities

Here we capture the relevant properties related to agents' capabilities. These are represented in the form *CAPABLE(A,X,σ)*, where A is the agent, X is an activity, skill, or plan, and σ is a situation. These capabilities are represented as *CAPABLE(R1,do(R1,navigate(room)))*, *CAPABLE(R2,do(R2,scan-room)),{isWorking(R2,sensor)}*, etc. The operator in the task is capable of all of the *Activities* defined above, except for *update-GUI*, which is exclusive to the robots. This activity has several functions that involve marking some aspect of the GUI to indicate task progress. This includes incrementing the panel counter, checking off a tool that was found, marking a toolbox with an *X* after searching, highlighting revealed panels in red, graying out shut off panels, and printing a message to the task log. The robots are capable of all *Activities* above except for *supervise*, which is exclusive to the operator. This activity involves directly tele-operating the robots to perform any of the actions of which they are capable.

To represent what agents can perceive, we similarly use *PERCEIVABLE(A,X,σ)*, which states that agent A can perceive X in situation σ. In general, robots can only perceive objects in the room that they currently occupy, including toolboxes and wall panels. For example, *PERCEIVABLE(R1,toolbox2,{located(R1,room2)})*. Robots can also perceive that object properties have changed as a result of their own actions: *PERCEIVABLE(R1,isSearched(R1,toolbox2),{located(R1,room2)})*. In the SMM Condition (see subsection II.D), agents can share this information with one another.

3. Agent and task states

Here we capture the relevant properties related to agents' mental states and goals. For representing what agents know about the task, we use *KNOWS-OF(A,X)*, where X is any entity in the domain, including other agents, objects, plans, etc. For example, all agents know about the other agents and the object types, e.g., *KNOWS-OF(O,R1)*, *KNOWS-OF(R2,GUI)*, etc. Regarding object properties, robots each maintain their own knowledge base depending on their experience in the task up to that point. For example, if R1 scanned *room5* then *KNOWS-OF(R1,isScanned(R1,room5))* is true. However, it does not follow that R2 knows this, unless it is the SMM Condition wherein the robots share information.

In addition to *KNOWS-OF*, we also include *KNOWS-HOW(A,X)* to represent actions that the agents can perform. For example, robots know how to perform all the actions that they are capable of: *KNOWS-HOW(R1,do(R1,search-toolbox)*, *KNOWS-HOW(R2,do(R2,scan-room))*, etc. Similarly, the operator knows how to perform all of their actions, e.g., *KNOWS-HOW(O,do(O,supervise))*.

Agent goals are represented using *GOAL(A, γ)*, where γ includes the following:

Goals (γ): {*shutOffAll(panel), foundAll(tool), visited(A,toolbox), visited(A,panel), visited(A,room), searched(A,toolbox), found(A,tool), scanned(A,room), shutoff(A,panel), updatedGUI(A), changedRoles(A), rebooted(A,<Equipment>)*}

Some of these goals are comprised of subgoals. For example, *GOAL(A,shutOffAll(panel))* involves the completion of *SUBGOAL(A,scanned(A,room))* and *SUBGOAL(A,shutoff(A,panel))* for every room and panel in the spacecraft. These in turn require the goals *visited(A,room)* and *visited(A,panel)* to be achieved as pre-conditions. Similarly, *GOAL(A,foundAll(tool))* involves *SUBGOAL(A,searched(A,toolbox))* to be achieved for every toolbox on the spacecraft until *found(A,tool)* is true for every tool. These in turn require *visited(A,room)* and *visited(A,toolbox)* as pre-conditions. In this way, agents represent a hierarchy of goals and subgoals that are needed to accomplish the task.

4. Plans and autonomy

The robots also represent plans to allow them to perform the task autonomously. Plans (π) involve sequences of actions to achieve a goal in a given situation. Plans are represented using the primitives *ACHIEVES(π,ϕ,σ)* and *ADOPTED(A,π,σ)*, where π is a plan, ϕ is the goal that the plan achieves, and σ is the situation in which the plan is carried out. These are used to give us the notion of "Seeing To It" (*STI*), wherein an agent A is seeing to it that some goal is accomplished. *STI* is formally defined as follows: *STI(A,ϕ,σ):* $\iff \exists\pi[ADOPTED(A,\pi,\sigma) \wedge ACHIEVES(\pi,\phi,\sigma)]$. This is used to represent ongoing actions in progress. For example, *STI(R1,scanned(room3),{located(R1,room3)})* represents the fact that R1 has adopted a plan which will achieve the goal of scanning room3 in the current situation.

The main plan called *FINISH-ROOM* involves the robot performing all possible actions of their current role in the current room (e.g., searching all toolboxes), switching roles, and then performing all possible actions with the other role (e.g., shutting off all damaged panels). Effectively, this allows the robots to search all toolboxes and scan/shut off all panels in a room. Once FINISH-ROOM has been executed, the robot moves to the next room in clockwise order and repeats the process. This loop represents the autonomous policy that both robots follow.

In order to enable the aforementioned autonomous policies, we include a set of rules to guide agent behavior. One such rule sets as a precondition for an action that the goal of that action has not already been accomplished. Using the following rule, an agent can only do an action (e.g., scan a room) if they have not already done so:

$$located(A,room) \wedge \neg ACHIEVED(scanned(A,room)) \implies do(A,scan-room)$$

To allow agents to reboot broken equipment immediately, we include the following rule:

$$\neg isWorking(A,camera) \implies do(A,reboot(camera))$$

We also need a way for agents to be able to change roles if a critical piece of equipment has broken. The following rule accomplishes this:

$$HasRole(A,toolSearchBot) \wedge \neg isWorking(A,camera) \implies do(A,changeRoles)$$

Finally, for navigation, we need a way to prevent both agents from being in the same room. The following rule achieves that by holding that if an agent has a goal to be in a certain room, and either that room is occupied or the other robot is moving there, then the navigation goal is retracted and an error message is printed to the task log.

$GOAL(visited(R1, room)) \wedge (movingTo(R2, room) \vee isOccupied(room)) \implies \neg GOAL(visited(R1, room)) \wedge do(R1, update - GUI)$

5. Obligations and norms

Here we capture norms related to agent behavior. The predicate *Superior(A1,A2)* represents the command hierarchy. *Superior(O,R1)* and *Superior(O,R2)* are used to represent that the human operator is superior to both robots, and (implicitly) that the robots are on the same command level as peers. To capture normative behavior, we include the predicates *PROPOSES(A1,A2,X)*, along with *ACCEPTS(A2,A1,X)* and *REJECTS(A2,A1,X)*, where X is a plan or goal. This allows us to use the following rule to represent the fact that subordinates always accept proposals:

$PROPOSES(O, R1, X) \wedge SUPERIOR(O, R1) \implies ACCEPTS(R1, O, X)$

For example, if the operator instructs R1 to move to *room4*, this would be represented as follows:

PROPOSES(O,R1,visited(room4)) $\wedge$ *SUPERIOR(O,R1)* $\implies$ *ACCEPTS(R1,O,do(R1,navigate(room4)).*

By accepting the proposal, the following rule triggers, thus updating R1's goal to *GOAL(R1,visited(room4))*:

ACCEPTS(A2,A1,X) $\implies$ *GOAL(A2,X)*

Since the robots are subordinates to the operator, they will always accept proposed goals, except for when they are busy performing another action. The following rules allows us to handle this case:

PROPOSES(A1,A2,X) $\wedge$ *isBusy(A2)* $\implies$ *REJECTS(A2,A1,X)*

The additional rule below causes a rejection to update the operator's task log with an error message explaining the reason for the rejection:

REJECTS(A2,A1,X) $\implies$ *do(A2,update-GUI)*

Other situations in which the robots are obligated to update the GUI include after they perform an action successfully and after an action fails (e.g., due to equipment failure). We include an additional primitive called "Intends to Know", *ITK(A,X)*, which is used as a basis to inform the operator after a robot has completed some action. An example of this is *ITK(O,isSearched(R1,toolbox6))*. This expression is used as a pre-condition for the following rule that triggers an update-GUI action by the robot:

ACHIEVED(R1,ϕ) $\wedge$ *ITK(O,ϕ)* $\implies$ *do(R1,update-GUI)*

6. Functional roles of agents in teams

Here we capture aspects related to team structure and team role. The three roles in the primary task are *toolSearchBot*, *panelScanBot*, and *operator*. Each role is represented by the goals, actions, and obligations associated with that role. In addition to these, we also include a set of *Requirements*, which is represented as *REQUIRED(e,ϕ)*, where e is the equipment required to achieve goal ϕ, e.g., REQUIRED(gripper,shutoff(panel)). We also include *REQUIRES(A,e)*, which represents that agent A requires equipment e, e.g., *REQUIRES(A,gripper)*. These role-based requirements describe the interdependencies involved in the team activity. For example, they show that each role is only capable of achieving certain goals, and the operator can control the robots to accomplish these goals.

Communication channels are represented here as actions tied to a specific role. Note that the *update-GUI* action is the main way that the robots communicate with the operator. The robots do not have a way to communicate with each other, except for in the SMM Condition where they share a knowledge base.

Given these roles, the overall structure of the team can be defined in such a way as to represent the functional role of the various agents, the command hierarchy, and the available equipment types.[‡] Although not used in the current study, we can also represent the follow-up repair crew, which includes a robot in the role of *panelRepairBot*. The robot relies on the tasks performed by the Astrobees, and the information that they obtained about the tool locations. Since a role is defined by its requirements, this allows us to set the goals of the initial team as the requirements of the follow-up repair crew - namely *GOAL(shutOffAll(panel))* and *foundAll(tool)*.

D. Experimental conditions

Given that we want to test the benefit of SMMs, we include two conditions in our study - one in which the robots perform the task with an SMM (*SMM Condition*) and the other in which they perform the task without an SMM

[‡]Note that the roles in the initial team definition are the starting roles of the agents in the task, but these can change as the task progresses.

(*Baseline Condition*). Both conditions are identical with regards to the above framework and the knowledge that the agents represent. However, we include several additional rules in the SMM Condition which allow those agents to share team and task information with one another. This condition will be used to establish whether the core SMM capacity of sharing information provides additional benefits over robots using the same task and team representations, but without sharing a knowledge base.

The capabilities of the robots in the Baseline condition include performing all available actions corresponding to their role, operating autonomously (including during LOS) using the *FINISH-ROOM* plan defined above, tracking equipment failure and rebooting, changing roles as a result of equipment failure or to complete objectives, and updating the GUI as a result of goal completion. In general, the Baseline agents are capable of completing the task entirely on their own. However, the only information they use to update their knowledge and adapt their behavior is information that they themselves obtained during the course of the task.

In the SMM condition, the agents have the exact same capabilities, along with one key difference - agents in this condition can share information with each other. The agents are continually synchronized and know of each others' task focus, progress, and status. This largely affects their *FINISH-ROOM* plan. Since the plan involves checking whether certain goals have been met in the current room, the SMM agents can check if the other agent already accomplished some of those goals in order to save time. For example, if upon entering a room in the *panelScanBot* role, an agent knows that the room has already been scanned by the other agent, it can proceed to shutting off the revealed panels without needing to do a *scan-room* action first.

This kind of synchronization between agents is achieved through a few additional predicates and rules. First, we introduce the notion of common goal *CG(γ)*. These goals contrast with the agent specific goals such as *GOAL(A,γ)* in that it represents a joint goal, and it no longer matters which agent achieves the goal. The following common goals are included for the SMM agents:

CG(scanned(room))
CG(shutoff(panel))
CG(searched(toolbox))

When an agent completes a personal goal, the corresponding common goal will be marked as complete using the following rule (e.g., for scanning rooms):

$ACHIEVED(GOAL(scanned(A,room))) \implies ACHIEVED(CG(scanned(room)))$

With this in place, SMM agents only need to check if a common goal was achieved before performing an action. The following rule can be used for this purpose (using scanning rooms as an example again):

$located(A,room) \wedge \neg ACHIEVED(scanned(room)) \implies do(A,scan-room)$

Next, we introduce the predicate *COMMON-KNOWLEDGE(ϕ)*, which is used to represent the shared knowledge base to which both SMM agents have access. All aspects of the task which were previously represented using *KNOWS-OF(A,X)* and *KNOWS-HOW(A,X)* are now represented as common knowledge. For example, if R1 finds the *wrench* after searching *toolbox4* then the expressions *isFound(wrench)* and *searched(toolbox4)* are added to *COMMON-KNOWLEDGE*. This allows the agents to share task-relevant information and use that information to automate their behavior in a team-oriented manner.

III. Exploratory study

To identify the potential benefits to teaming of the proposed SMM framework, we implemented the maintenance task described in subsection II.B above, and ran an online study using Amazon Mechanical Turk (AMT). For this preliminary study, the simulated robots behaved as if they had SMMs, but they were not actually running a robotic architecture.

The study had the following structure. First, participants accepted the Human Intelligence Task (HIT) on the AMT web page and consented to participate in the study. Next, they were taken to the task instructions page and then the tutorial. Every participant completed a 3-minute tutorial to ensure that they understood the task procedures. The tutorial was the same as the main task, except for differences in robot starting position, reduced duration of LOS (to allow more practice time), and a different order of equipment failure. After 3 minutes of practice, they were taken back to the study page where they were able to start the main task. The main task lasted anywhere from 5-15 minutes, depending on the participant and the experimental condition. Following this was the post-experiment survey, where participants answered questions about their workload, situational awareness, and attitudes about the team (team workload).

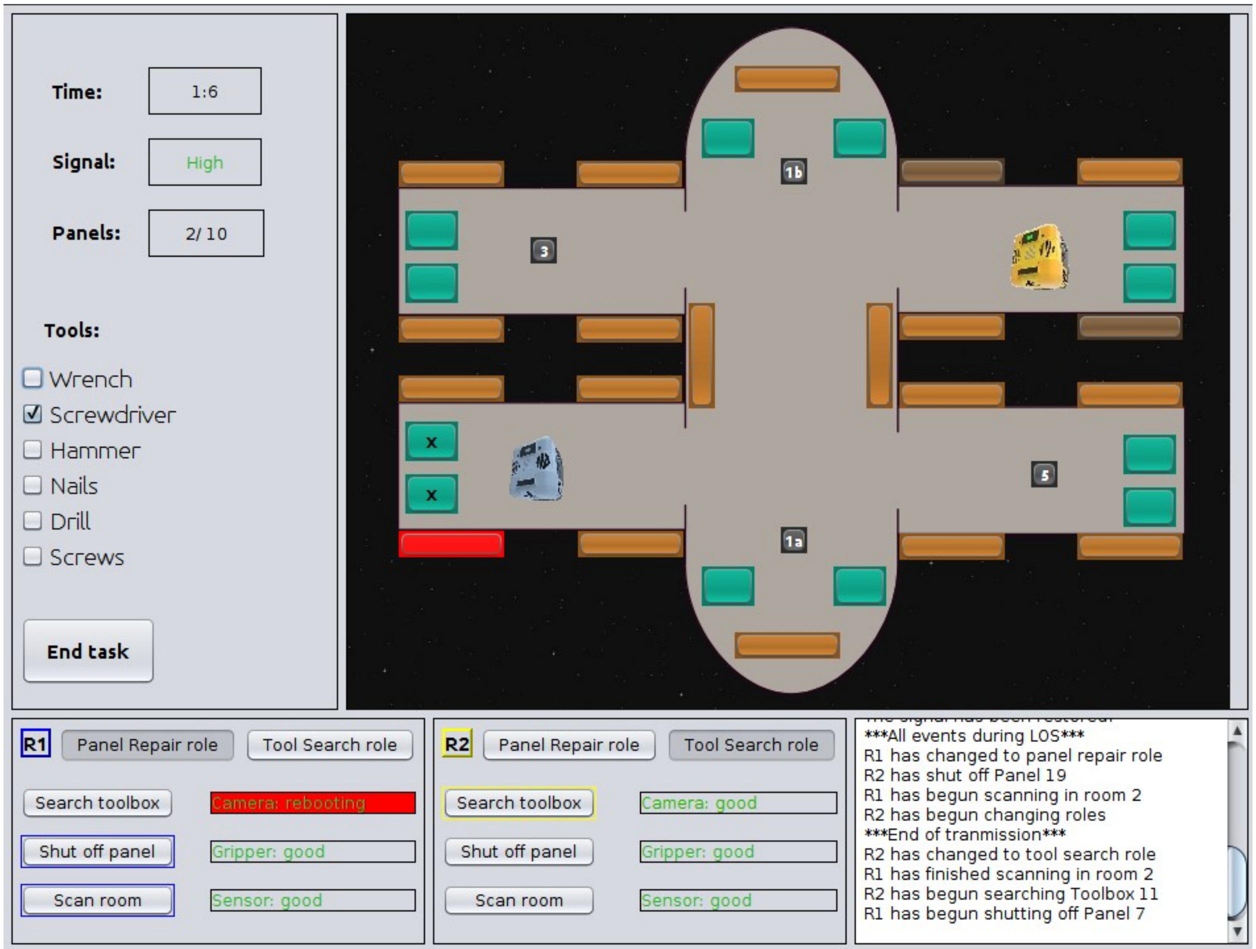

Fig. 3 Operator's graphical user interface (GUI) used in the exploratory study. R1 is the blue robot and R2 is the yellow robot. The rooms (1a, 1b, 2, 3, 4, 5) are denoted by gray squares, each with a number of wall panels (brown rectangles) and toolboxes (turquoise squares). Damaged panels reveled by a Scan action are highlighted in red, and shut off panels appear as a darker brown. Toolboxes are marked with an X after they are searched, and any tools found are automatically updated on the status panel. The status panel on the left displays the time, signal strength, as well as task progress. The robot action panel (bottom) shows the available actions for each robot corresponding to their current role - the available actions are highlighted in the robot's color. Equipment status corresponding to each action is also displayed. Finally, the task log (bottom right) displays all events that occurred during the task. After LOS, the events that the operator missed are summarized.

A. Task design

As described in subsection II.B, the task involves a human operator supervising two free-flying robots aboard a spacecraft to perform several tasks in preparation for a robotic repair crew. The participant is presented with a graphical user interface (GUI) that serves as the "operator's interface" and displays various pieces of task-relevant information (see Fig. 3). They are able to click on various parts of the GUI to control the robots to perform a number of actions. There are several actions available to the robots, including *Move*, *Search Toolbox*, *Shut Off Panel*, *Scan Room*, and *Change Roles* (see descriptions below). Each action takes some time to perform, during which time the robots wiggle back and forth to indicate that they are performing an action. The task log (bottom right in Fig. 3) also updates when an action begins, ends, and fails. Actions can fail if the user instructs a robot to: perform an action while they are in the middle of another action (e.g., scanning while moving), perform an action in a different room (e.g., searching a toolbox in a different room), perform an action for which the required equipment is broken (including the case where the equipment fails mid-action), perform an action that requires a different role, or instruct a robot to move to a room that another robot occupies or to which it is moving. In all these cases, an error message appears in the task log, and the robot does not produce that action; in all other cases, the action succeeds.

The goal of the task is to locate and take stock of six tools (wrench, screwdriver, hammer, nails, drill, screws) scattered throughout the spacecraft environment, and shut off ten panels that are damaged. The tools can be found by performing a *Search Toolbox* action at one of the multiple toolboxes (turquoise squares) on the spacecraft. Some of these toolboxes are empty, and others have one or more tools that are needed. When a tool is located, it is automatically indicated on the corresponding check box on the operator's map. A toolbox is marked with an X on the GUI after it has been searched.

There are six rooms in the spacecraft - a central one split into two parts (1a and 1b), and four wings (2, 3, 4, and 5). Robots can be directed to move between these rooms, however both robots cannot stop in the same room. To locate damaged panels, a robot must perform a *Scan Room* action inside one of these rooms, which reveals the location of all damaged panels in that room. This information is also marked on the operator's map by highlighting the damaged panels in red. Once the damaged panels are revealed, they will need to be turned off to prevent further damage and to prepare them for repair. To do this, a robot must be in the same room as the damaged panel and perform a *Shut Off Panel* action. The corresponding panel on the GUI will turn a darker shade of brown to indicate that it has been shut off.

During the task, occasional loss-of-signal (LOS) events will occur at set intervals, and will last for 10 seconds and 15 seconds alternatively. The LOS duration of 10/15s was chosen to allow for some robot autonomy while also keeping the operator engaged. When LOS occurs, the screen goes black and a message appears that says: "Loss of signal. Please wait...". During this time, robots will still perform actions, but the operator cannot see the GUI at all. Once the signal returns, the GUI re-appears and a message in the task log updates the operator on all the actions that occurred during LOS.

To add complexity to the task and require operator adaptation, several of the robot actions (*Search, Shut Off,* and *Scan*) require a corresponding piece of equipment to be functional (Camera, Gripper and Sensor, respectively). The robot equipment is prone to failure, and will malfunction during the course of the task on a set schedule. Over time, a new piece of equipment will fail in the following order: R1 - Camera, R2 - Gripper, R2 - Sensor, R1 - Gripper, R2 - Camera, R1 - Sensor. The order is arbitrary, but consistent for every participant, and will repeat in that order until the task is completed. When a malfunction occurs, the equipment status box will turn red, and the text will change to <Equipment>: rebooting. During this time, the robot cannot perform that action. After a lengthy period of time has elapsed, the equipment will be functional again and the text will change back to <Equipment>: good. This indicates that the action is once again available to perform.

B. Autonomy policies

The robots are capable of performing the entire task autonomously. They follow a simple policy in which they check all remaining actions to perform in their current room (scan, shut off all damaged panels, and search all toolboxes), perform all these actions one at a time, and then move to the next room clockwise. This policy involves changing roles in every room to accomplish both subtasks. If a piece of equipment necessary for some action is broken, then the robot will do all other actions and move to the next room.

Participants are instructed that, due to safety protocol, the robots must be under supervisory control for the duration of the task. Thus, while the robots can act autonomously, the operator can still override their actions. To accommodate this, the robots wait for a command after every action they perform. If a command is not issued within a small time window then they will proceed to the next action in their script. So while technically the operator can simply allow the robots to perform the task themselves, this will be very slow and inefficient since the robots wait in between every action. The only exception to this is that during LOS, this idle period is eliminated, and the robots string actions together immediately. This is done so that the LOS period could be reduced and robots can perform at least several actions during that period.

There are some slight differences in the way that these autonomous policies play out, and this factor distinguishes our two experimental conditions. We have two main conditions in the task - Baseline and SMM. These conditions are identical, except for the fact that robots in the SMM Condition can share task-relevant information. Specifically, they follow the rules described in subsection II.D, which prevent the robots from performing an action that has already been done by the other robot. This represents the fact that they share a knowledge base, and have up-to-date knowledge of the team progress towards the joint goal. During autonomy, SMM robots will skip over actions that the other robot performed, but if instructed to do so by the operator, they will display a message in the task log indicating that the action has already been done. In the Baseline Condition, these rules do not apply, and each robot only knows which actions it performed. Thus, it will not repeat actions that it has already done, but it may repeat actions that the other robot did, both during autonomy and when instructed by the operator.

C. Metrics and predictions

The metrics used in our study are based on those from [26] and other sources, summarized below:

- Time to task completion
- Neglect tolerance
- Movement efficiency
- Action repetitions
- Loss-of-signal productivity
- Workload
- Team workload
- Situational Awareness

The task ends when all tools are located, and when all damaged panels are revealed and shut off. Since every participant will finish the task completely, our main performance measure is the time it takes to complete the task. Another objective measure we use is *neglect tolerance*, which tracks the frequency of operator intervention[27]. Neglect tolerance is approximated by counting the number of times the operator clicked on the robots. *Movement efficiency* is the number of times that the robots moved, both autonomously and with operator instruction. *Action repetitions* is the number of repeat actions that the robots performed, counting both autonomous and instructed actions. A repeat action occurs when a robot performs a *Search, Scan,* or *Shut Off* action that has already been performed by the other robot. In the SMM Condition, robots will not autonomously perform repeat actions, but if instructed to by the operator this counter will go up even though the robot will not actually perform the action. Finally, *LOS Productivity* is a measure of how many actions were performed during all the LOS periods. The counter is incremented whenever a robot performs any of the actions available to them during LOS (moving, scanning, changing roles, etc.), except for repeat actions.

We also include several subjective metrics. Workload is measured using the NASA Task-Load Index (NASA-TLX)[28], team workload is measured using the Team Workload Questionnaire (TWLQ)[29], and situation awareness is measured using the Situational Awareness Rating Technique (SART)[30].

In terms of hypotheses, we predict that people in the SMM Condition will perform the task more quickly and efficiently. We expect to see a reduction in overall completion time, the number of movements, the number of repeat actions, and the number of operator interventions, as well as an increase in LOS productivity. This is because the SMM robots afford more efficient behavior by minimizing repeat actions, and allowing the operator to trust them to act more independently. We also expect to see a reduction in workload and an increase in situational awareness for this same reason. The mental demand and pacing will be reduced, thus attenuating the level of effort and stress experienced by the operator. Moreover, we expect operators in the SMM Condition to show less divided attention and more spare mental capacity to focus on other elements of the task since the robots will not need constant supervision.

D. Results

In total, we collected data from 70 participants: 35 in the baseline Condition and 35 in the SMM Condition. Of these, all 70 completed the task, but only 56 (28 per condition) completed all the surveys. We consider the results for the objective task measures separately from the subjective survey measures.

We conducted one-way between-subjects ANOVAs (N=70) to compare the effect of Condition on our main objective measures. We observed significant effects for *completion time* [$F(1,68) = 11.107$, $p < .005$], *movement efficiency* [$F(1,68) = 13.587$, $p < .001$] and *action repetitions* [$F(1,68) = 215.796$, $p < .001$], and a trending effect for *neglect tolerance* [$F(1,68) = 3.275$, $p = .075$]. See Table 1 for an overview of the results. Overall, these results support our hypotheses of increased performance and efficiency in the SMM Condition.

One-way ANOVAs (N=56) were also conducted to examine effects of our subjective measures. Since each of these metrics was comprised of several sub-scales, we report the average of all the sub-scales for each metric. There were no effects of *workload* [$F(1,54) = 2.502$, $p = .12$], *team workload* [$F(1,54) = 0.438$, $p = .51$], or *situational awareness* [$F(1,54) = 1.303$, $p < .256$]. See Table 2 for an overview of the results. Counter to our hypotheses, we did not observe any differences in our survey measures between the conditions.

Table 1 Table of results for objective task measures

	Baseline		SMM			
	M	SD	M	SD	F-Value	P-Value
Completion Time	9.96	2.42	8.18	2.03	11.11*	.001
Neglect Tolerance	16.86	18.15	10.54	9.83	3.28	.075
Movement Efficiency	19.11	7.32	13.97	3.81	3.98*	<.001
Action Repetitions	49.23	18.84	1.91	2.83	215.80*	<.001
LOS Productivity	12.11	2.26	11.69	2.25	0.63	.43

Table 2 Table of results for subjective survey measures

	Baseline		SMM			
	M	SD	M	SD	F-Value	P-Value
Workload (NASA-TLX)	4.88	1.00	4.40	1.26	2.50	.12
Team Workload (TWLQ)	6.10	2.28	5.71	2.02	0.44	.51
Situational Awareness (SART)	4.94	0.88	4.66	0.91	1.30	.26

IV. Discussion

In general, we found that teams in the SMM Condition performed the task more quickly and efficiently, as measured by a reduction in task completion time (see Fig. 4), as well as a reduction in the number of robot moves and repeat actions (see Fig. 5). This is in line with our predictions, and demonstrates that robots with SMMs can improve team performance in this domain. However, we did not find any effect from our survey measures, so the objective improvements did not appear to be associated with a reduction in workload or situational awareness.

A. Summary and interpretation of results

Our main performance measure was the time to task completion, and we found that the SMM group was 22% quicker, which amounts to almost two minutes on average over the course of the task. Given that the scenario was framed as an emergency, this is a significant improvement. Moreover, because the follow-up repair task cannot be completed until the wall panels are shut off and the tools located, the time reduction observed in the SMM group allows the follow-up repair task to start sooner. Because the conditions in our study were identical except for the sharing of information in the SMM Condition, the effects we found were not due to differences in GUI design between conditions but rather the benefit that SMMs provide, namely sharing task-relevant information and using it to adapt behavior. These benefits could have important implications for the success of real missions for which these robots are needed.

Apart from the improvement in completion time, we also observed an improvement in efficiency. Specifically, SMM robots moved a fewer number of times, and repeated fewer actions. This likely contributed to the overall time improvement, but is also important independently because it reduces the likelihood of adverse events in real robots. For example, a reduction in movement would also mean a reduction in the possibility of bumping into objects, running out of battery, or other faults that may occur with movement. Factors such as this were not considered in our task, but they may be important in the context of real missions.

Two objective measures for which we did not see improvement were *neglect tolerance* and *LOS productivity*. However, we still observed a 60% improvement in neglect tolerance in the SMM Condition. Though this did not quite reach statistical significance (p = .075), we expect it to with a larger sample. Ultimately, this reduced intervention can potentially allow the operator to focus on other aspects of the task, such as monitoring the task log, or tracking task progress. It also offers the potential for the mission to use a smaller team, as the enhanced autonomy allows the operator to supervise multiple robots. Finally, there was no difference in *LOS productivity* between conditions. Though we predicted that the SMM robots would show an improvement here due to the fact that they do not repeat actions,

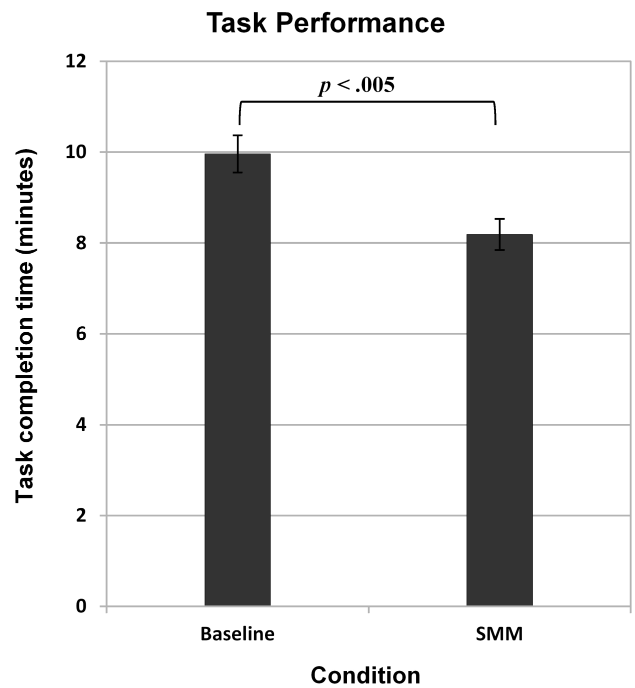

Fig. 4 **Average task completion time between the Baseline and SMM Conditions. Error bars represent SEM.**

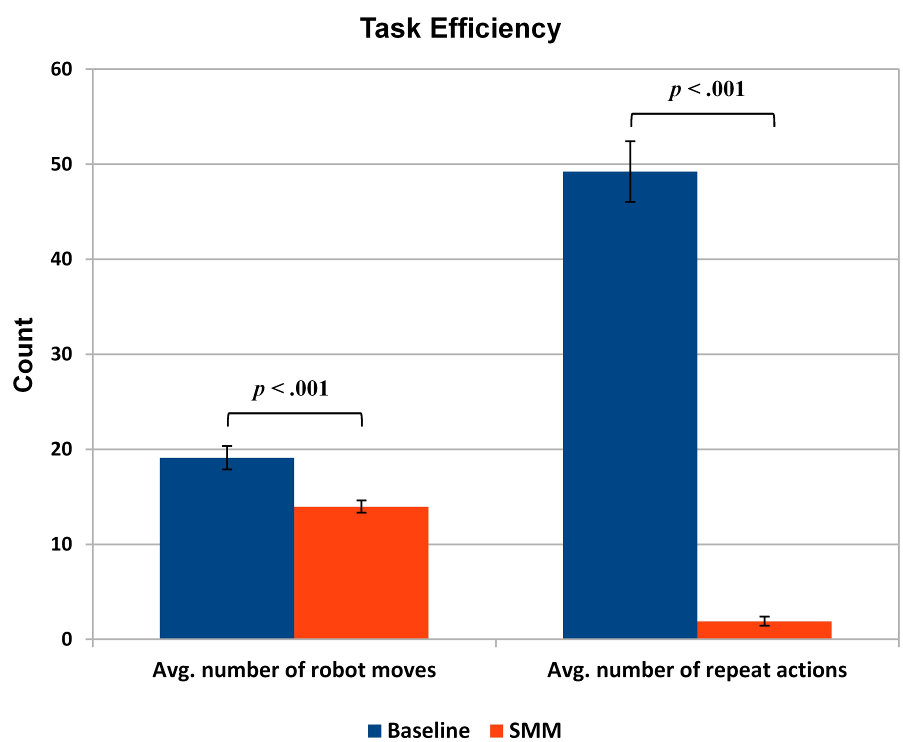

Fig. 5 **Average count of robot moves and repeat actions between the Baseline and SMM Conditions. Error bars represent SEM.**

the LOS period was perhaps not long enough for this advantage to show. Because the task involved 10- and 15s LOS periods, and the robots took 10s to perform an action, they only averaged a few actions over the course of LOS.

In terms of our subjective survey measures, we did not find any effect of workload, team workload, or situational awareness. Regarding workload, there was a numerical reduction in the TLX score in the SMM Condition for each of the six sub-scales (mental demand, physical demand, temporal demand, performance, effort, and frustration) as well as a reduction in the average TLX score, but this was not significant (p = .12). It seems that overall the task did not impose a great deal of workload on participants in either condition; they were all right around the midway point on the 7-point scale. The same appears to be the case for team workload, as measured by TWLQ. The TWLQ scale asks people to rate coordination, communication, and team monitoring on a 10-point scale. Again, participants in both conditions seemed to be right around the midway point. Finally, SART is a subjective measure of situational awareness, which asks people to rate aspects related to attentional supply, attentional demand, and understanding on a 7-point scale. We found no significant effects of this measure, indicating that people in both conditions generally had the same level of understanding and attention regarding the situation. Since many of the questions in this survey relate to the task itself, it does not appear that the effects of the SMM influenced peoples' responses.

Though we did not find significant improvements in workload in our exploratory study, it is reasonable to expect a reduction in at least some kinds of workload due to the significant increase in performance and efficiency that SMMs afford. The relationship between workload and task performance is a complex one, and large individual differences can make it difficult to study. For example, some people perform better under pressure, while others show a reduction in performance[32]. Moreover, measuring workload is not trivial, and subjective metrics alone may not capture the kinds of workload of interest in these kinds of teams[31]. More work is needed to understand the extent to which workload influences performance in both human teams and human-robot teams. Overall, though we did not find an effect of our subjective survey measures, the results of the objective performance measures are very promising.

B. Future work

Overall, this preliminary study suggests that SMMs can provide a great benefit to the operator and to the team in a space robotics domain. The next step is to integrate the formalizations described in this paper with a robotic architecture (DIARC[33]) and validate the results of the current study. By testing the framework on real robots, we will be able to more thoroughly evaluate the proposed mechanisms and the extent to which they influence teaming. The key questions we will be addressing in these evaluations include how to best facilitate coordination, what is needed for synchronization, and how to use robots to minimize team effort.

Future work will involve varying other parameters of the task domain, such as the team structure, time scale, and spatial range. Interaction modality is also an important factor to manipulate, as robots will need to seamlessly switch between various control modes within a single interaction. We also seek to explore different kinds of interactions, such as robots working after humans or concurrently with humans. Environmental factors of real space environments (microgravity, noise, variable lighting, etc.)[34] are important to consider, and will be first explored in VR simulations and, in the future, potentially in real space environments (e.g., HRI studies on the ISS). Finally, we also intend to explore stochastic domains in which robots will need to handle uncertainty regarding, e.g., sensor errors, poor communication, latency, etc.

V. Conclusion

Our study is the first to explore the potential benefit of SMMs as a means to support human-robot teaming in the complex domain of space robotics. As such, it demonstrates the potential for SMMs to coordinate the interaction of distributed teams in complex tasks and to improve task performance and efficiency. Based on the results of our exploratory study, our computational SMM framework is useful for enabling robots to keep track of various teammates' knowledge, goals, and plans. The core benefit of the framework appears to be the synchronizing of this SMM information to allow all robots on a team to utilize this knowledge and to coordinate their actions in a team-oriented manner. More work is clearly needed, but this work represents a step in the direction to bridge the gap between human-robot teaming separated in time[35, 36] and the kind of real-time interdependence envisioned for future space robotics applications.

Acknowledgments

This work was funded by a NASA Space Technology Research Fellowship under award 80NSSC17K0184.

References

[1] Fong, T., Rochlis Zumbado, J., Currie, N., Mishkin, A., and Akin, D. L., "Space telerobotics: unique challenges to human–robot collaboration in space," *Reviews of Human Factors and Ergonomics*, Vol. 9, No. 1, 2013, pp. 6–56.

[2] Johnson, M., Bradshaw, J. M., Feltovich, P. J., Jonker, C. M., Van Riemsdijk, M. B., and Sierhuis, M., "Coactive design: Designing support for interdependence in joint activity," *Journal of Human-Robot Interaction*, Vol. 3, No. 1, 2014, pp. 43–69.

[3] Mishkin, A. H., Limonadi, D., Laubach, S. L., and Bass, D. S., "Working the Martian night shift-the MER surface operations process," *IEEE robotics & automation magazine*, Vol. 13, No. 2, 2006, pp. 46–53.

[4] Fong, T., Abercromby, A., Bualat, M. G., Deans, M. C., Hodges, K. V., Hurtado Jr, J. M., Landis, R., Lee, P., and Schreckenghost, D., "Assessment of robotic recon for human exploration of the Moon," *Acta Astronautica*, Vol. 67, No. 9-10, 2010, pp. 1176–1188.

[5] Fong, T., Bualat, M., Deans, M., Adams, B., Allan, M., Altobelli, M., Bouyssounouse, X., Cohen, T., Fluckiger, L., Garber, J., et al., "Robotic follow-up for human exploration," *AIAA SPACE 2010 Conference & Exposition*, 2010, p. 8605.

[6] Deans, M., Fong, T., Bualat, M., Heggy, E., Helper, M., Hodges, K., and Zacny, K., "Using robots before and after humans to improve space exploration," *GLEX-2012.04*, Vol. 1, 2012, p. x12344.

[7] Crusan, J. C., Smith, R. M., Craig, D. A., Caram, J. M., Guidi, J., Gates, M., Krezel, J. M., and Herrmann, N. B., "Deep space gateway concept: Extending human presence into cislunar space," *2018 IEEE Aerospace Conference*, IEEE, 2018.

[8] Bualat, M., Barlow, J., Fong, T., Provencher, C., and Smith, T., "Astrobee: Developing a free-flying robot for the international space station," *AIAA SPACE 2015 Conference and Exposition*, 2015, p. 4643.

[9] Diftler, M. A., Mehling, J., Abdallah, M. E., Radford, N. A., Bridgwater, L. B., Sanders, A. M., Askew, R. S., Linn, D. M., Yamokoski, J. D., Permenter, F., et al., "Robonaut 2-the first humanoid robot in space," *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, IEEE, 2011, pp. 2178–2183.

[10] Fong, T., Nourbakhsh, I., Kunz, C., Flückiger, L., Schreiner, J., Ambrose, R., Burridge, R., Simmons, R., Hiatt, L. M., Schultz, A., et al., "The peer-to-peer human-robot interaction project," *In Proceedings of AIAA Space*, 2005.

[11] Fong, T., Kunz, C., Hiatt, L. M., and Bugajska, M., "The human-robot interaction operating system," *In Proceedings of the 1st ACM SIGCHI/SIGART conference on Human-robot interaction*, ACM, 2006, pp. 41–48.

[12] Jonker, C. M., Van Riemsdijk, M. B., and Vermeulen, B., "Shared mental models," *Coordination, organizations, institutions, and norms in agent systems vi*, Springer, 2011, pp. 132–151.

[13] Espinosa, A., Kraut, R., Slaughter, S., Lerch, J., Herbsleb, J., and Mockus, A., "Shared mental models, familiarity, and coordination: A multi-method study of distributed software teams," *ICIS 2002 Proceedings*, 2002, p. 39.

[14] Mathieu, J. E., Heffner, T. S., Goodwin, G. F., Salas, E., and Cannon-Bowers, J. A., "The influence of shared mental models on team process and performance." *Journal of applied psychology*, Vol. 85, No. 2, 2000, p. 273.

[15] Stout, R. J., Cannon-Bowers, J. A., Salas, E., and Milanovich, D. M., "Planning, shared mental models, and coordinated performance: An empirical link is established," *Human Factors*, Vol. 41, No. 1, 1999, pp. 61–71.

[16] Orasanu, J., "Shared mental models and crew decision making," , 1990.

[17] Lee, M., Johnson, T., and Jin, M. H., "Toward understanding the dynamic relationship between team and task shared mental models as determinants of team and individual performances," *International journal of information technology and business management*, Vol. 8, No. 1, 2012, pp. 1–14.

[18] Yen, J., Fan, X., Sun, S., Hanratty, T., and Dumer, J., "Agents with shared mental models for enhancing team decision makings," *Decision Support Systems*, Vol. 41, No. 3, 2006, pp. 634–653.

[19] Goodrich, M. A., and Yi, D., "Toward task-based mental models of human-robot teaming: A bayesian approach," *International conference on virtual, augmented and mixed reality*, Springer, 2013, pp. 267–276.

[20] Nikolaidis, S., and Shah, J., "Human-robot teaming using shared mental models," *ACM/IEEE HRI*, 2012.

[21] Ososky, S., Schuster, D., Jentsch, F., Fiore, S., Shumaker, R., Lebiere, C., Kurup, U., Oh, J., and Stentz, A., "The importance of shared mental models and shared situation awareness for transforming robots from tools to teammates," *Unmanned Systems Technology XIV*, Vol. 8387, International Society for Optics and Photonics, 2012, p. 838710.

[22] Fan, X., and Yen, J., "Modeling cognitive loads for evolving shared mental models in human–agent collaboration," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, Vol. 41, No. 2, 2011, pp. 354–367.

[23] 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.

[24] Gervits, F., Warne, C., Downs, H., Eberhard, K., and Scheutz, M., "Exploring Coordination in Human-Robot Teams in Space," *AIAA SPACE and Astronautics Forum and Exposition*, 2017, p. 5309.

[25] Smith, T., Barlow, J., Bualat, M., Fong, T., Provencher, C., Sanchez, H., and Smith, E., "Astrobee: A new platform for free-flying robotics on the international space station," *Int. Symp. on AI, Robotics, and Automation in Space (iSAIRAS)*, 2016.

[26] Steinfeld, A., Fong, T., Kaber, D., Lewis, M., Scholtz, J., Schultz, A., and Goodrich, M., "Common metrics for human-robot interaction," *Proceedings of the 1st ACM SIGCHI/SIGART conference on Human-robot interaction*, ACM, 2006, pp. 33–40.

[27] Olsen, D. R., and Goodrich, M. A., "Metrics for evaluating human-robot interactions," *Proceedings of PERMIS*, Vol. 2003, 2003, p. 4.

[28] Hart, S. G., "NASA-task load index (NASA-TLX); 20 years later," *In Proceedings of the human factors and ergonomics society annual meeting*, Vol. 50, 2006, pp. 904–908.

[29] Sellers, J. M., "Team Workload Questionnaire (TWLQ): Development and Assessment of a Subjective Measure of Team Workload," 2013.

[30] Taylor, R., "Situational Awareness Rating Technique(SART): The development of a tool for aircrew systems design," *In Proceedings of Situational Awareness in Aerospace Operations (AGARD)*, 1990.

[31] Harriott, C. E., "Workload and task performance in human-robot peer-based teams," Ph.D. thesis, Vanderbilt University, 2015.

[32] Cox-Fuenzalida, L.-E., "Effect of workload history on task performance," *Human Factors*, Vol. 49, No. 2, 2007, pp. 277–291.

[33] Scheutz, M., "ADE: Steps toward a distributed development and runtime environment for complex robotic agent architectures," *Applied Artificial Intelligence*, Vol. 20, No. 2-4, 2006, pp. 275–304.

[34] Rehnmark, F., Currie, N., Ambrose, R. O., and Culbert, C., "Human-robot teaming in a multi-agent space assembly task," *Automation Congress, 2004. Proceedings. World*, Vol. 15, IEEE, 2004, pp. 69–74.

[35] Bualat, M. G., Abercromby, A., Allan, M., Bouyssounouse, X., Deans, M. C., Fong, T., Flückiger, L., Hodges, K. V., Hurtado Jr, J., Keely, L., et al., "Robotic recon for human exploration: Method, assessment, and lessons learned," *Analogs for planetary exploration*, 2011, pp. 117–135.

[36] Deans, M., Fong, T., Lee, P., Hodges, K., Helper, M., Landis, R., Riley, S., Bualat, M., Pacis, E., and Kobayashi, L., "Robotic scouting for human exploration," *AIAA Space 2009 Conference & Exposition*, 2009, p. 6781.