

MIT Open Access Articles

*Computing the Effects of Operator Attention
Allocation in Human Control of Multiple Robots*

The MIT Faculty has made this article openly available. **Please share**
how this access benefits you. Your story matters.

Citation: Crandall, Jacob W., Mary L. Cummings, Mauro Della Penna, and Paul M. A. de Jong. "Computing the Effects of Operator Attention Allocation in Human Control of Multiple Robots." IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans 41, no. 3 (May 2011): 385-397. © Copyright 2011 IEEE

As Published: <http://dx.doi.org/10.1109/TSMCA.2010.2084082>

Publisher: Institute of Electrical and Electronics Engineers (IEEE)

Persistent URL: <http://hdl.handle.net/1721.1/81173>

Version: Final published version: final published article, as it appeared in a journal, conference proceedings, or other formally published context

Terms of Use: Article is made available in accordance with the publisher's policy and may be subject to US copyright law. Please refer to the publisher's site for terms of use.



Computing the Effects of Operator Attention Allocation in Human Control of Multiple Robots

Jacob W. Crandall, Mary L. Cummings, *Member, IEEE*, Mauro Della Penna, and Paul M. A. de Jong

Abstract—In time-critical systems in which a human operator supervises multiple semiautomated tasks, failure of the operator to focus attention on high-priority tasks in a timely manner can lower the effectiveness of the system and potentially result in catastrophic consequences. These systems must integrate computer-based technologies that help the human operator place attention on the right tasks at the right times to be successful. One way to assist the operator in this process is to compute where the operator's attention should be focused and then use this computation to influence the operator's behavior. In this paper, we analyze the ability of a particular modeling method to make such computations for effective attention allocation in human–multiple-robot systems. Our results demonstrate that it is not sufficient to simply compute and dictate how operators should allocate their attention. Rather, in stochastic domains, where small changes in either the endogenous or exogenous environment can dramatically affect model fidelity, model predictions should guide rather than dictate operator attentional resources so that operators can effectively exercise their judgment and experience.

Index Terms—Attention allocation, human-performance modeling, human–robot interaction, multirobot teams, supervisory control.

I. INTRODUCTION

A COMMON theme in many computing systems is for users (or operators) to simultaneously manage multiple semiautomated software agents, robots, or other system components. In these systems, which include human–robot systems, transportation systems, power systems, and other everyday computer systems, the way an operator allocates attention among these various tasks can greatly impact a system's effectiveness. This is particularly true in time-critical missions in which failure of the operator to attend to high-priority tasks in a timely manner can have catastrophic consequences.

Manuscript received September 7, 2009; revised April 3, 2010; accepted June 12, 2010. Date of publication November 21, 2010; date of current version April 15, 2011. This work was supported by the Massachusetts Institute of Technology Lincoln Laboratory. The review of this paper was arranged by Editor T. Tsuji.

J. W. Crandall is with the Computing and Information Science Program, Masdar Institute of Science and Technology, Abu Dhabi, United Arab Emirates (e-mail: jcrandall@masdar.ac.ae).

M. L. Cummings is with the Department of Aeronautics and Astronautics, Massachusetts Institute of Technology, Cambridge, MA 02139-4307 USA (e-mail: missyc@mit.edu).

M. Della Penna is with the Della Penna Autotrasporti s.r.l., 80147 Napoli, Italy (e-mail: mauro@dellapenna.it).

P. M. A. de Jong is with the Department of Aerospace Engineering, Delft University of Technology, 2628 CN Delft, The Netherlands (e-mail: p.m.a.dejong@tudelft.nl).

Color versions of one or more of the figures in this paper are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/TSMCA.2010.2084082

However, effective attention management in these multi-dimensional, uncertain, and high-workload situations is difficult given the operators' cognitive limitations. To mitigate the negative effects of these limitations, computer-based technologies can help operators manage their attentional resources. In this paper, we present and analyze methods for improving operator attention allocation in human–multiple-robot systems (HMRSs) in which highly autonomous robots perform command and control tasks under the direction and supervision of a single human operator.

Three related approaches have been suggested and studied to improve operator attention allocation in HMRSs. In the first approach, visualizations representing the status, plans, and progress of the robots in the system are provided to the operator via a graphical user interface (GUI). Such visualizations implicitly tell the operator which tasks need to be performed and when the operator should perform them. For example, timeline displays for unmanned aerial-vehicle systems can be used to present a schedule of anticipated events [1]. The operator can use this display to infer and investigate which tasks to perform and when to perform them [2].

Warning systems are a second well-studied methodology for improving operator attention allocation in HMRSs. A typical warning system detects potential critical events and then explicitly alerts the operator of those events via visual, auditory, or haptic signals. In addition to potential use in HMRSs [3], [4], such warning systems have been used and studied in many systems involving human interaction with automation, including nuclear power plants [5], aviation [6], [7], and automobiles [8], [9].

In this paper, we consider a third approach for improving operator attention allocation in which the system explicitly suggests or dictates where the operator's attention should be focused at any given time. This is done by using a model of the HMRS to compute a utility-maximizing *operator attention allocation scheme* (OAAS), or *operator scheduling policy* [10]–[13]. An OAAS specifies where human attention should be allocated in all possible situations. This approach has two potential benefits. First, as people are often poor schedulers [14], particularly under time pressure [15], ensuring that an operator attends to the tasks with the highest utility (or priority) can significantly increase the system's performance. Second, aiding the operator in attention allocation can potentially reduce the amount of effort it takes for the operator to choose a task to perform, thus reducing the operator's workload. This, in turn, can lead to additional performance increases.

Gaining these potential benefits, however, is challenging. To increase the system's performance and potentially lower

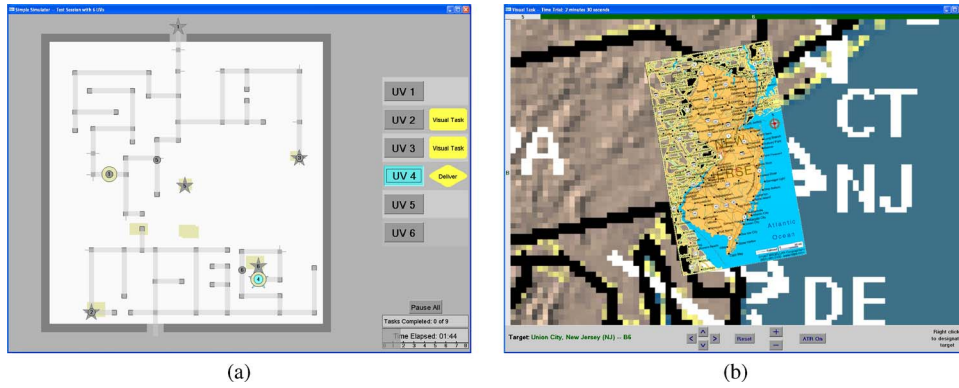


Fig. 1. Two-screen human-robot interface used in RESCU.

operator workload, the model of the HMRS must be able to effectively compute the effects of OAASs on the system's effectiveness. Furthermore, once a desirable OAAS is computed, effective human-robot-interface technologies must be constructed to help the operator attend to the recommended tasks. Failure to accurately model the effects of OAASs or to convey the needs of the system through the human-robot interface will likely lead to lower system performance and increased operator workload [2], [16].

Thus, a system seeking to improve operator attention allocation using this approach must address two questions. First, how can an effective OAAS be computed? Second, once computed, how should the system use the knowledge of this OAAS to help the operator attend to the most appropriate task? Answering this second question includes determining whether the operator or the automation is ultimately responsible for selecting the task that the operator performs [17], [18].

In this paper, we address these questions using the HMRS and user study described in Section II. In Section III, we review a previously introduced computational modeling method for HMRSs [19] and introduce how this model can be used to compute the effects of OAASs. We also show how the model can be used to estimate an "optimal" OAAS under certain assumptions. In Section IV, we analyze the appropriateness of these assumptions. In Section V, we describe methods for using the optimal OAAS to assist the operator in allocating his or her attention among various tasks. We present the results of a second user study evaluating the resulting HMRS in Section VI and discuss the lessons learned in Section VII.

II. USER STUDY I: BACKGROUND

We used the *Research Environment for Supervisory Control of Unmanned Vehicles (RESCU)* software test bed to evaluate the modeling and interaction technologies described in this paper. In this section, we give a brief overview of RESCU; additional details are given in previous work [20]. We also present results on operator attention allocation in RESCU.

A. Software Test Bed

An operator of an HMRS commonly assists in performing a set of abstract tasks. These tasks include mission planning and

replanning, robot path planning and replanning, robot monitoring, sensor interpretation, and target designation. RESCU was designed to capture these tasks in a time-critical mission.

In RESCU, an operator supervises multiple simulated robots in performing a search-and-rescue mission. The operator is tasked with identifying and gathering 22 tokens from a building, which is represented as an (initially) unknown maze, within an 8-min time period. The locations of the tokens are given to the system. The mission goal is to collect as many of these tokens as possible during this time period while ensuring that all robots are out of the building when time expires. Specifically, the operator is tasked with maximizing the following objective function:

$$\text{Score} = \text{TokensCollected} - \text{RobotsLost} \quad (1)$$

where *TokensCollected* is the number of tokens collected during the 8-min period and *RobotsLost* is the number of robots in the building when time expires.

A token is collected using a three-step process.

- 1) A robot moves to the location of a token in the building. This step requires the operator to be involved at some level in mission planning, target designation, path planning and replanning, and robot monitoring.
- 2) The robot "picks up" the token. In real-world systems, the operator would likely need to perform visual tasks such as identifying the token from imagery or interpreting other sensor data. In RESCU, this burden on the operator is simulated by requiring the operator to locate a designated city on a computer-based Google-Earth-style map of the U.S. As with scanning video imagery, this task requires the operator to dedicate cognitive resources to a *visual search task*.
- 3) The robot carries the token out of the building via one of two exits. This step requires the operator to monitor the robot and assist in path planning and replanning.

The operator uses a two-screen display to oversee and help perform the RESCU mission. On the left screen [Fig. 1(a)], the locations of the robots and tokens are displayed in the building. A partial map of the building is also displayed. This map is a combination of the map created by each robot as they move about the world. The right screen [Fig. 1(b)] displays the Google-Earth-style map used in the visual search task.

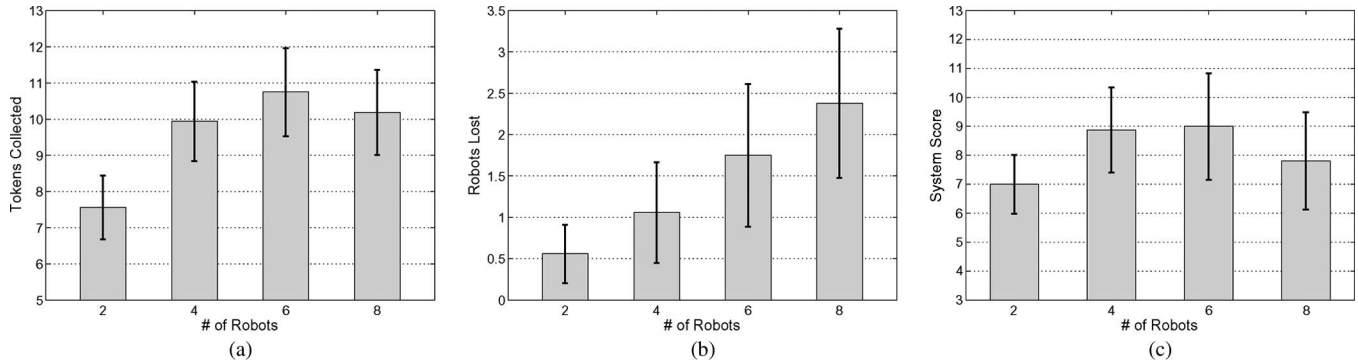


Fig. 2. Average number of (a) tokens collected, (b) robots lost, and (c) system score in RESCU. Error bars show 95% confidence intervals.

In RESCU, the operator interacts with a single robot at a time. Once the operator has selected a robot, by clicking on a button corresponding to that robot, the operator can perform three tasks for that robot. These tasks are as follows.

- 1) *Goal assignment*—The operator specifies the robot’s destination, typically a token or building exit, by dragging a goal marker to the desired location. The robot then navigates through the building, using Dijkstra’s algorithm with estimated path costs, in search of this destination. The robot’s projected path is displayed on the screen.
- 2) *Replanning*—If desired, the operator modifies the selected robot’s target destination or modifies the path that the robot intends to follow to its destination.
- 3) *Visual Search Task*—Once the robot finds a token, the operator helps the robot “pick up” the token.

To assist the operator in determining which robot to service, a visual alerting system [yellow message in Fig. 1(a)] notifies the operator when a robot 1) is not assigned a task, 2) needs assistance “picking up” a token, or 3) in danger of being left in the building in the last minute of the mission.

B. Operator Attention Allocation in RESCU

RESCU was initially used to analyze how changes in team size, robot-autonomy characteristics, and the human–robot interface affect the HMRS’s performance [19], [20]. In this paper, we are interested in how OAASs, or the way operators allocate their attention among multiple robots, affect HMRS’s effectiveness. We begin by analyzing observed operator attention allocation in a previous RESCU user study.¹

1) *Experimental Setup*: The user study was a within-subject study with 16 participants. The independent variable was robot team size, which had four levels: two, four, six, and eight robots. The dependent variable was the number of tokens collected and robots lost and, for the purposes of this effort, the OAASs used by the participants.

Each participant was trained on all aspects of the RESCU test bed. They then completed three comprehensive practice

sessions after which the four test sessions were administered. The order that the participants saw each team size was counter-balanced to offset order effects.

2) *Results*: The average performance of these HMRSs is shown in Fig. 2, which demonstrates that the average number of tokens collected peaked at about six robots, while the number of robots lost consistently increased with team size. Given the objective function in (1), system effectiveness was highest when the team consisted of between four and six robots. Adding additional robots to the team after six robots appears to decrease the HMRS’s effectiveness. On average, operators were unable to effectively manage more than six robots, although some individual operators could. This same trend was observed in a similar, but separate, 12-subject study [20].

However, this decrease in system effectiveness with increasing numbers of robots need not occur. With eight-robot teams, operators could simply utilize fewer of the available robots to increase system effectiveness to levels observed in six-robot teams. While three operators implemented this strategy, most did not. Thus, the decrease in system effectiveness in teams with many robots can be traced, at least in part, to the way that operators allocated their attention among the robots.

To help understand the operators’ attention-allocation behavior in these larger teams, we consider operator *strategy profiles*, which depict the number of goal assignment, replanning, and visual search tasks (payload operations) the operators performed in each minute. Fig. 3(a) and (b) shows the average strategy profile across all operators for six- and eight-robot teams. The strategy profiles are fairly similar for both six- and eight-robot teams except for two revealing differences. First, in eight-robot teams, the average operator sent more than six robots into the building in the first minute. While this is not surprising since there were eight robots to control rather than just six, this result illustrates that at least some operators were unable to identify that they would score higher, on average, using only six robots rather than eight.

The second main difference between the strategy profiles for the six- and eight-robot teams occurred in the last minute, where Fig. 3(b) shows a sharp increase in replanning tasks in the eight-robot condition. In this condition, operators tended to send too many robots into the building. They were then forced to try to correct this mistake in the last minute by sending robots back out of the building before they gathered a token. This is

¹Previously published work describing this user study [19] focused on developing a predictive model for human–robot team performance and did not examine strategies in any way.

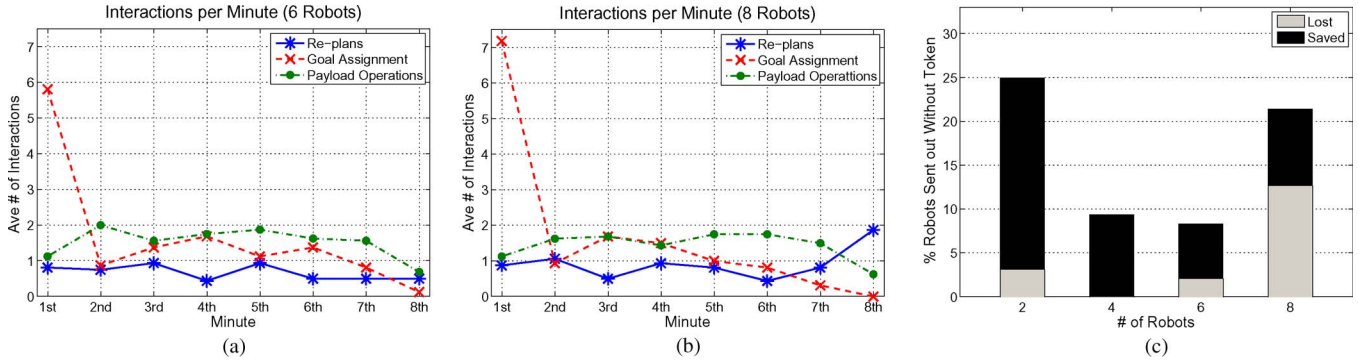


Fig. 3. Average operator strategy profile for (a) six- and (b) eight-robot teams. (c) Reactive operator behavior in the eighth minute.

shown in Fig. 3(c), which plots the average percentage of robots sent out of the building without a token in their last interaction with the operator. The figure shows that operators tended to remove more robots from the building without gathering a token in the two- and eight-robot conditions than in the four- and six-robot conditions. This shows that operators sent more robots into the building than could collect tokens when they managed two- or eight-robots. While operators typically corrected this mistake with two robots, they were unable to do so in a timely manner when they managed eight robots. This resulted in more robots lost.

In short, some operators made time-critical errors in attention allocation that reduced the HMRS's effectiveness. This behavior is reminiscent of human behavior observed by Sheridan and Tulga [14], where operators could not determine optimal scheduling rules. In the remainder of this paper, we describe and evaluate how a detailed model of the HMRS can potentially be used to improve operator attention allocation in RESCU. In the next section, we present a modeling method for computing which robot the operator should service at any given time. In Section V, we discuss methods for helping operators to allocate their attention to the recommended robots.

III. MODELING THE EFFECTS OF OAASs

Modeling the effects of operator attention allocation in supervisory control systems has been addressed in past work [14], [21]–[23]. Additionally, several methods for computing operator scheduling strategies in human–unmanned vehicle systems have been proposed [10]–[12]. However, we are not aware of a generalized and thoroughly tested method for computing the effects of unobserved OAASs in HMRSs nor of work which links such models with the human–robot interface.

Previous studies have demonstrated that a stochastic discrete-event simulation can provide reasonably good predictions of how changes in team size, robot autonomy, and human–robot-interface characteristics alter system effectiveness [19], [20], [24]. In this paper, we evaluate the ability of this same modeling method to estimate the effects of OAASs on system effectiveness in RESCU. We first give an overview of the modeling method and describe how it can be used to compute the effects of OAASs on system effectiveness. We then describe how the resulting models can be used to compute an “optimal” OAAS and discuss the application of this computation to RESCU.

A. Modeling Human–Multirobot Systems

Crandall *et al.* [19] model an HMRS with the four-tuple $\mathcal{M} = (\mathcal{II}, \mathcal{NI}, \mathcal{ST}, \mathcal{OAAS})$ of stochastic structures. We briefly describe each stochastic structure

- 1) *Interaction impact.* The stochastic process $\mathcal{II}(\sigma)$, where σ is the system state at the beginning of the human–robot interaction, describes how the robot's status (or state) changes over time as it interacts with the operator. $\mathcal{II}(\sigma)$ implicitly specifies the length of a single human–robot interaction.
- 2) *Neglect impact.* The stochastic process $\mathcal{NI}(\sigma)$ describes how the robot's status changes in the absence of human–robot interactions, given that the robot's last interaction with the operator ended in system state σ .
- 3) *Switching time.* The probability distribution $\mathcal{ST}(\sigma)$ describes how long it takes for the operator to determine which robot to service given the system state σ .
- 4) *OAAS.* The structure $\mathcal{OAAS}(\sigma)$ is a probability distribution over robots. This probability distribution defines which robot the operator services given the system state σ .

Once these four structures are determined for a particular HMRS, a discrete-event simulation can be used to simulate the HMRS and, in turn, estimate its effectiveness. When the four structures are estimated by observational data of a particular HMRS, previous studies have shown that the estimated system effectiveness computed by the model, denoted by $U(\mathcal{M})$, matches the observed system effectiveness [19]. Thus, the model is able to describe the behavior of the HMRS.

B. Computing the Effects of Operator Attention Allocation

The model can be used to predict how changes in the OAAS will alter the effectiveness of HMRS. Let $\mathcal{M}^O = (\mathcal{II}^O, \mathcal{NI}^O, \mathcal{ST}^O, \mathcal{OAAS}^O)$ denote the model formed from observations of an HMRS in RESCU. Our goal is to determine how well the system would have performed had the operators allocated their attention among the robots differently (i.e., when operators use some alternate OAAS, denoted as \mathcal{OAAS}'). If we assume that \mathcal{II}^O , \mathcal{NI}^O , and \mathcal{ST}^O would not be altered by a change in the OAAS, the resulting model of the new HMRS would be $\mathcal{M}' = (\mathcal{II}^O, \mathcal{NI}^O, \mathcal{ST}^O, \mathcal{OAAS}')$. Then, $U(\mathcal{M}')$ is a prediction of how well the system would perform when

the operator selects robots according to $\mathcal{O}AAS'$ rather than $\mathcal{O}AAS^O$.

The accuracy of the estimated change in system effectiveness due to a shift in the OAAS relies on the important assumption that changing from $\mathcal{O}AAS^O$ to $\mathcal{O}AAS'$ does not affect \mathcal{II}^O , \mathcal{NI}^O , or \mathcal{ST}^O . While previous studies showed that changes that affected \mathcal{NI} and \mathcal{II} did not significantly alter the other structures [19], the assumption might not hold for changes in the OAAS. We reevaluate the integrity of this assumption in Sections IV and VI.

C. Computing $\mathcal{O}AAS^*$

Our task is to identify how operators should allocate their attention among robots in order to obtain high system effectiveness. Let Ω be the set of possible OAASs. Then, formally, we want to compute $\mathcal{O}AAS^* \in \Omega$, such that $U(\mathcal{M}^*) \geq U(\mathcal{M}') - \varepsilon$, for all $\mathcal{O}AAS' \in \Omega$ and some small $\varepsilon \geq 0$. We begin by identifying the set Ω .

Recall that an OAAS specifies how human attention is allocated to the robots in the team in all situations, or system states. In RESCU, we define system state with respect to 1) the individual states of all robots in the team and 2) mission time, defined in the continuous interval $[0, 8]$ min. Since mission time is continuous, there are an infinite number of possible OAASs.

We identify a reasonable subset of these OAASs over which we seek to find $\mathcal{O}AAS^*$ by making two simplifying assumptions the effects of which we analyze in Section IV. First, the time component of system state can be reduced by discretizing the mission into a finite set of time periods. In RESCU, we divide the mission time into eight discrete periods, one corresponding to each minute of the 8-min scenario.

Second, in constructing Ω , we also limit the number of possible probability distributions that $\mathcal{O}AAS(\sigma)$ can take on to those probability distributions that place all weight on robots in a particular state. These distributions can be expressed as preference orderings over robot states, which specify the order that the operator services the robots.

In addition to reducing the size of Ω , a preference ordering over robot states is easier for people to understand than a probability distribution for each of an infinite number of system states. However, the “optimal” preference ordering for a particular minute could depend on the OAAS used in previous minutes. We evaluate the impact of this assumption in Section IV.

In RESCU, these assumptions mean that each OAAS in Ω consists of a sequence of eight preference orderings over the five robot states defined in Table I. As an example, consider Table II, which shows an OAAS as a series of eight preference orderings. In the table, $X > Y$ denotes that the operator attends to robots in state X before attending to robots in state Y . The OAAS in Table II specifies that the operator should give priority to robots in state R in the first minute. If no robot is in state R , the operator should attend to a robot in state A , etc. In the second minute, the operator first attends to robots in state I . Once there are no robots in state I , the operator services robots in state A , and so on.

TABLE I
FIVE ROBOT STATES USED TO COMPUTE OAASS IN RESCU

State	Explanation
A – Assignment	The robot is outside the building waiting for an assignment
I – Idle	The robot is idle in the building, but not at the location of a token
P – Payload Ready	The robot is ready to “pick up” a token.
R – Re-plan	The robot is navigating the building, but is either not taking the likely shortest path to its destination, or could benefit from being re-assigned to a different destination
G – Good Progress	The robot is efficiently moving toward its assigned destination

TABLE II
 $\mathcal{O}AAS^*$ FOR EIGHT-ROBOT TEAMS

Minute	Preference Ordering
1 st	$R > A > I > P > G$
2 nd	$I > A > P > R > G$
3 rd	$P > I > R > A > G$
4 th	$A > I > P > R > G$
5 th	$I > P > A > R > G$
6 th	$R > I > P > A > G$
7 th	$I > R > P > A > G$
8 th	$I > P > R > A > G$

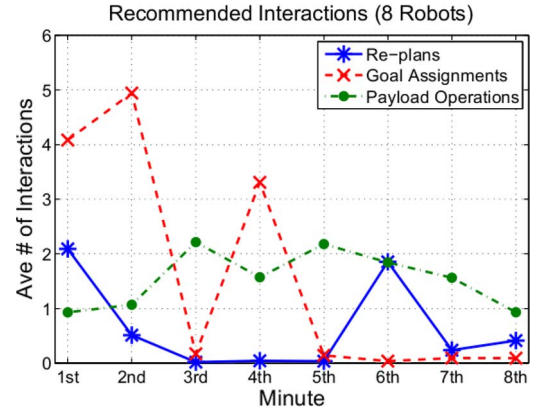


Fig. 4. Predicted strategy profile for following $\mathcal{O}AAS^*$.

If we assume that robots in state G (Good Progress) always have the lowest priority, there are 24 possible preference orderings over robot states in RESCU. Thus, the set Ω consists of 24^8 OAASs. We used a genetic algorithm on a population of 35 OAASs to estimate $\mathcal{O}AAS^*$ from Ω . After each generation, the two OAASs with the highest fitness were kept in the population for the next generation. The remaining population of the subsequent generation was randomly selected (according to fitness level) from the previous population and then altered through mutation (50%) and crossover (50%). Alternate parameter values had little effect on the results.

D. $\mathcal{O}AAS^*$ in RESCU

Using the structures \mathcal{II}^O , \mathcal{NI}^O , and \mathcal{ST}^O observed for eight-robot teams in the first user study (Section II) and the method described in the previous section, we computed $\mathcal{O}AAS^*$ for eight-robot teams in RESCU. This OAAS is shown in Table II. The model-predicted strategy profile for following $\mathcal{O}AAS^*$ is shown in Fig. 4.

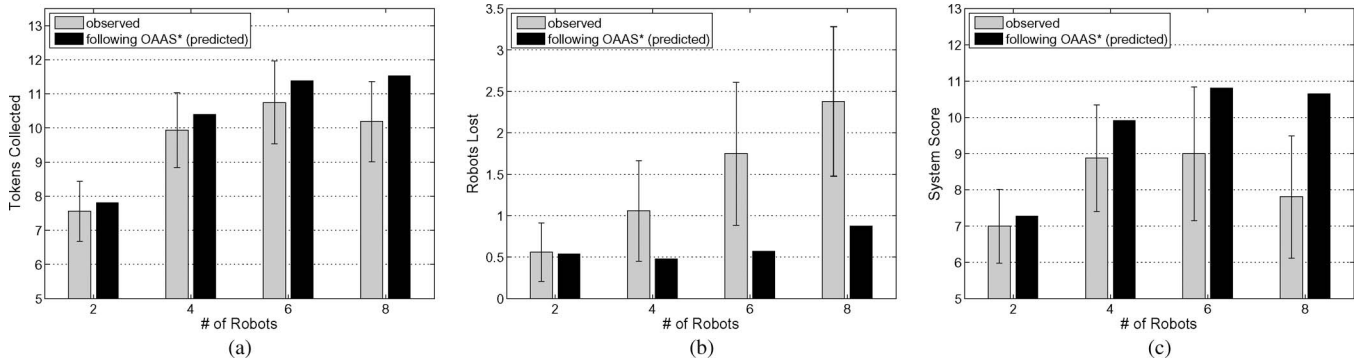


Fig. 5. Predicted system effectiveness for following $\mathcal{O.A.A.S}^*$ compared with observed system effectiveness. Error bars show 95% confidence intervals.

In the first minute, $\mathcal{O.A.A.S}^*$ specifies that the operator should attend to robots in the replanning state R . When no robots are in this state, the operator should perform goal assignment (i.e., attend to robots in state A). Since all robots are initially in state A , most interactions in the first minute begin with robots in state A . However, if any robot enters state R while searching for their designated goals, the operator should switch attention to that robot. Thus, in the first minute, most of the interactions are with robots in state A and R . In the second minute, robots in state I are given the highest priority. However, since there are typically not many robots in state I at this point, most of the robot selections are to robots in state A . This results in the remaining robots being sent into the building (interactions with robots in state A), after which robots that are ready to “pick up” tokens are serviced.

While discretizing mission time causes sudden changes in operator behavior from minute to minute, the strategy profile of $\mathcal{O.A.A.S}^*$ in Fig. 4 provides several interesting insights. First, for eight-robot teams, $\mathcal{O.A.A.S}^*$ gives low priority to sending robots into the building (i.e., servicing robots in state A) in the final 3 min. This recommendation agrees with experimental observations made in Section II-B, which show that operators sent too many robots into the building in the sixth and seventh minutes, causing them to spend extra time replanning in the last minute. The model predicts that following $\mathcal{O.A.A.S}^*$ would substantially reduce this problem.

Second, $\mathcal{O.A.A.S}^*$ specifies that the operator should give less priority to replanning in minutes two through five as compared with what they did in the first user study [Fig. 3(b)]. Rather, in these minutes, $\mathcal{O.A.A.S}^*$ suggests that operators should perform tasks that the robots cannot perform themselves, such as goal assignment and payload operations. However, by the sixth minute, $\mathcal{O.A.A.S}^*$ recommends that the operator should give high priority to replanning and rerouting robots in order to ensure that each robot in the building can “pick up” a token and exit the building before time expires.

Third, $\mathcal{O.A.A.S}^*$ suggests that operators should give high priority to replanning in the first minute. This is contrary to the operator behavior in the first user study, as participants typically dedicated the first minute to sending robots into the building. From a performance standpoint, it is not clear why it would be beneficial to give high priority to replanning in the first minute. Regardless of whether the model is “right” or “wrong” in this regard, discrepancies between what operators

believe should be done and what $\mathcal{O.A.A.S}^*$ suggests creates an interesting dynamic. We revisit this issue in Section VI.

The model predicts that the eight-robot HMRS would have substantially higher performance if operators had followed $\mathcal{O.A.A.S}^*$ rather than $\mathcal{O.A.A.S}^O$ (Fig. 5). For eight-robot teams, the model predicts that using $\mathcal{O.A.A.S}^*$ would, on average, result in the team gathering more than one extra token per session while losing less than half as many robots. If accurate, the model predicts that the system score [(1)] would increase by nearly three points if operators had followed $\mathcal{O.A.A.S}^*$ in the eight-robot condition.

IV. ANALYSIS OF MODELING ASSUMPTIONS

The model’s accuracy depends on the appropriateness of the assumptions and simplifications it utilizes, including time-discretization granularity, preference orderings, the invariance of $\mathcal{I}I^O$ and $\mathcal{S}T^O$ given the changes in OAAS, and adherence to the stated mission-objective function. In this section, we analyze the effects of these assumptions on $\mathcal{O.A.A.S}^*$ and $U(\mathcal{O.A.A.S}^*)$, the HMRSs’ predicted performance when the operator follows $\mathcal{O.A.A.S}^*$. To do so, we compute the *adjusted* $\mathcal{O.A.A.S}^*$, the optimal OAAS when the assumption is violated. We then compare $U(\mathcal{O.A.A.S}^*)$ to $U(\text{adjusted } \mathcal{O.A.A.S}^*)$, $U(\mathcal{O.A.A.S}^O)$, and $U(\text{Random})$ (the estimated system performance when operators service robots at random).

A. Granularity of Time Discretization

Recall that we divided the mission time into eight discrete time periods to compute $\mathcal{O.A.A.S}^*$. Would more time periods allow for a substantially more effective OAAS? To answer this question, we computed the *adjusted* $\mathcal{O.A.A.S}^*$ and $U(\text{adjusted } \mathcal{O.A.A.S}^*)$ using between 2 and 16 time periods. Fig. 6(a) shows that dividing the mission time into more than eight periods does not produce a substantially more effective OAAS. Thus, given that a finer discretization is more likely to cause overfit due to the sparsity of the data, dividing mission time into eight time periods appears to be ideal.

B. Preference Orderings

The optimization model proposed in the previous section defines OAASs by a set of preference orderings, i.e., one for each minute. However, a preference ordering conveys the same

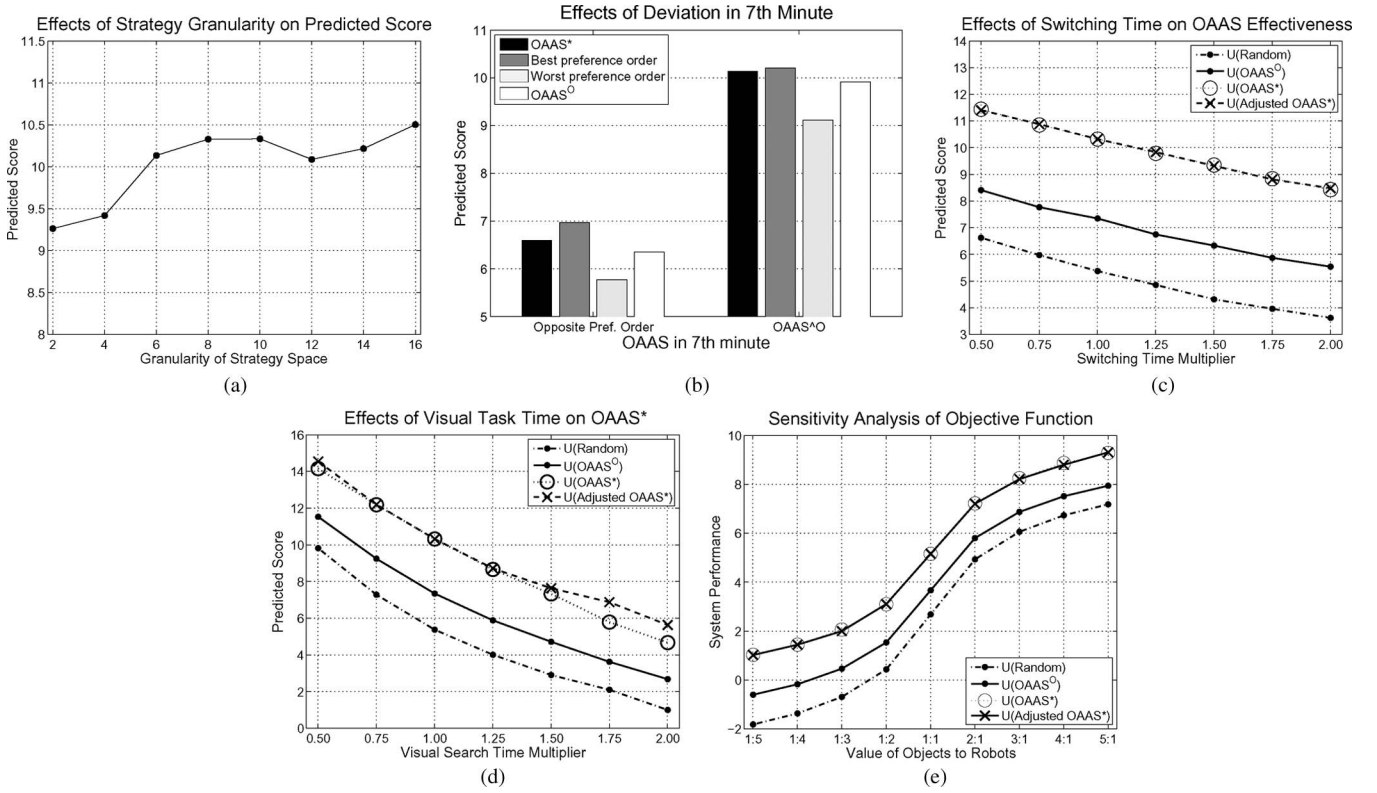


Fig. 6. Effects of (a) preference orderings, (b) discretization granularity, (c) switching times, (d) interaction times, and (e) the objective function on $\mathcal{O}AAS^*$.

robot selection for a system state consisting of one robot in state s_1 and seven robots in state s_2 as a system state with seven robots in state s_1 and one robot in state s_2 . Since preference orderings do not fully distinguish between system states, the best preference ordering for the eighth minute could potentially depend on the OAAS used in the previous minutes.

To better understand the limitations of using preference orderings, we consider the example situation in which operators deviate from $\mathcal{O}AAS^*$ during the seventh minute. We then use the model to assess the “optimality” of $\mathcal{O}AAS^*$ given this deviation by computing the preference ordering that produces the highest system score in the eighth minute (*best preference order*) given the deviation. For comparative purposes, we also consider two other eighth-minute OAASs: 1) the preference ordering that would produce the worst system score in the eighth minute (*worst preference order*) and 2) the operators’ modeled behavior from the first user study ($\mathcal{O}AAS^O$).

We consider two seventh-minute deviations. First, we consider the situation in which operators follow the opposite preference ordering from $\mathcal{O}AAS^*$ ($A > P > R > I$) in the seventh minute. This is the largest possible seventh-minute deviation from $\mathcal{O}AAS^*$ and would produce a very distinct system state entering the eighth minute than if no deviation occurred. Second, we consider the situation in which the operators use $\mathcal{O}AAS^O$ in the seventh minute. This represents a smaller deviation from $\mathcal{O}AAS^*$ and approximates how operators allocate their attention when they follow their own devices.

Fig. 6(b) shows the predicted system scores of the four eighth-minute OAASs given both seventh-minute deviations. In both cases, the model found a preference ordering that

outperforms $\mathcal{O}AAS^*$, which confirms that the best preference ordering is dependent on the OAAS used in previous minutes. However, the figure shows that following $\mathcal{O}AAS^*$ is still quite effective given the smaller, and more likely, seventh-minute deviation ($\mathcal{O}AAS^O$). Additionally, for both deviations, the model predicts that selecting robots according to $\mathcal{O}AAS^*$ is better than selecting robots according to $\mathcal{O}AAS^O$.

C. Invariance to Changes in Switching and Interaction Time

To compute $\mathcal{O}AAS^*$ and $U(\mathcal{O}AAS^*)$, we assumed that changing OAAS does not cause a change in II , NI , and ST . However, since a change in the OAAS might cause a change in the other processes, particularly ST and II , it is useful to analyze the potential consequences.

These consequences are predicted and shown in Fig. 6(c) and (d). Fig. 6(c) shows that, while switching time is predicted to have a substantial impact on the effectiveness of the system, $\mathcal{O}AAS^*$ is still the best OAAS when the switching time is halved or doubled. Fig. 6(d) shows that changes in visual-search-task times would also cause a substantial change in system effectiveness. However, unlike switching time, large changes in visual-search-task times make $\mathcal{O}AAS^*$ suboptimal. Thus, operators that are either much slower or much faster than average in performing the visual search task should follow alternate OAASs.

D. Objective Function

Throughout this paper, we have assumed that operators maximize the objective function given in (1), where the value of a

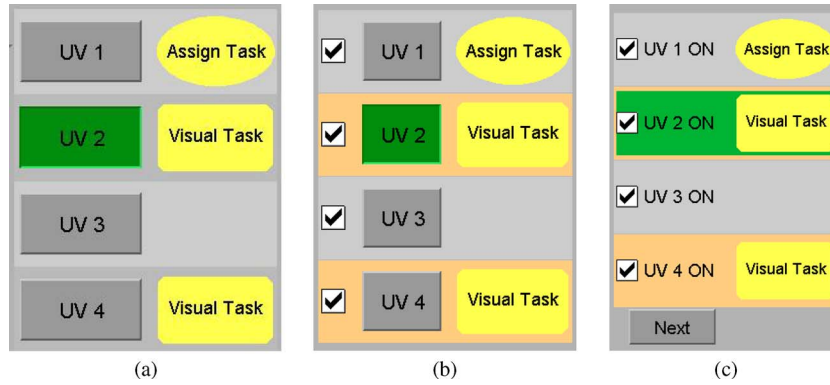


Fig. 7. GUIs for each robot-selection mechanism in RESCU. Recommendations are highlighted in orange in the Guided and Auto Modes. (a) Manual. (b) Guided. (c) Auto.

robot and a token are equivalent. However, operators may value robots more or less than tokens. Under such conditions, is it desirable to still follow $\mathcal{O}AAS^*$; or would an alternate OAAS produce better results?

Let c_T and c_R be the comparative value of tokens to robots. Then, let the system score be given by

$$\text{Score} = c_T * \text{TokensCollected} - c_R * \text{RobotsLost}. \quad (2)$$

For simplicity, we normalize c_T and c_R so that $c_T + c_R = 1$.

Fig. 6(e) shows the performance of the various OAASs for various ratios $c_T : c_R$. The figure shows that, while the system's predicted score changes based on the ratio $c_T : c_R$, the model does not find an OAAS that outperforms $\mathcal{O}AAS^*$ (computed for $c_T = c_R$). This suggests that, in RESCU, an operator is not forced to choose between maximizing tokens collected at the expense of robots lost (or vice-versa). Rather, an effective OAAS involves sending just enough robots into the building that each robot can collect and remove a token from the building before time expires.

In summary, while violations of the assumptions may cause $\mathcal{O}AAS^*$ to be suboptimal in some cases, these predictions indicate that it remains a relatively good OAAS. However, violations of the assumptions are likely to compromise the accuracy of the model's predictions of system score. We revisit these predicted phenomena in Section VI by way of user study.

V. MANAGING OPERATOR ATTENTION ALLOCATION

While following $\mathcal{O}AAS^*$ can potentially enhance system effectiveness in RESCU, it is not clear how knowledge of this scheme should be used to alter operator attention allocation *in situ*. In the first user study, the operator clicked on the button corresponding to the desired robot. This selection mechanism, which we refer to as *Manual Mode*, is illustrated by the portion of the GUI shown in Fig. 7(a) for a four-robot team. In the Manual Mode, a list of buttons is provided corresponding to each robot (labeled “UV” for *unmanned vehicle* in the GUI), with the currently selected robot highlighted [robot 2 in Fig. 7(a)]. Messages next to the buttons indicate the next tasks that the operator should likely perform for each robot.

Given the knowledge of $\mathcal{O}AAS^*$, alternate robot-selection mechanisms can be created to improve operator attention allocation. In this section, we describe two of these selection mechanisms for RESCU: *Auto Mode* and *Guided Mode*. We evaluate the effectiveness of these two selection mechanisms in the next section.

A. Auto Mode—Automating Robot Selection

One way to ensure that an operator conforms with $\mathcal{O}AAS^*$ is to automate the robot-selection process. We designed the *Auto Mode* to implement this robot-selection mechanism. In the Auto Mode, rather than directly selecting a robot, the operator clicks a button labeled “Next” to select a new robot to service [Fig. 7(c)]. The computer then automatically selects a robot for the operator to service based on $\mathcal{O}AAS^*$. This is done by selecting a robot with the highest priority (based on the robots' states) according to the current preference ordering specified by $\mathcal{O}AAS^*$. In the case that multiple robots are in the same high-priority state, a robot is chosen randomly. If the operator does not want to give any new commands to a selected robot, he or she can temporarily “turn-off” a selection by unchecking the check box next to that robot's label.

This selection mechanism has two potential advantages. First, it could lower the operator's workload since the operator would no longer need to spend time determining which robot to service. Second, given that $U(\mathcal{M}^*) > U(\mathcal{M})$, the system should be better able to schedule the operator's time, on average, than the operator (Fig. 5). Hence, system effectiveness is expected to improve if the automation is responsible for determining where operator attention should be focused.

While these potential advantages are enticing, this approach also has a number of potential disadvantages. First, removing the human from the robot-selection process can lower the operator's situation awareness [25]. This could potentially increase the amount of time it takes the operator to service a robot [26], [27], thus increasing the operator's workload rather than decreasing it [28]. Second, for any number of reasons, the model's utility estimates may be incorrect. In such situations, $\mathcal{O}AAS^*$ could lead to lower levels of effectiveness than if operators were left to their own devices.

B. Guided Mode—Recommending Robot Selection

Another potential method for leveraging knowledge of $\mathcal{O}AAS^*$ to improve OAASs is via a management-by-consent approach. This level of automation has been used successfully in various tasks performed in other studies of systems with multiple unmanned aerial vehicles [29], [30]. In management-by-consent, the system recommends courses of action, but the operator can decide whether to follow or not the recommendations. The *Guided Mode* implements this selection mechanism.

In the Guided Mode, the system uses $\mathcal{O}AAS^*$ to recommend which robots (determined by the robots' states) the operator should service by highlighting the suggested robots [in orange; Fig. 7(b)]. As with the Manual Mode, the operator chooses which robot to service by clicking on the button corresponding to that robot. Thus, the operator can decide whether to follow or not $\mathcal{O}AAS^*$. If the operator believes that the recommendations are in error, he or she can simply ignore them. Additionally, the operator can “turn off” recommendations by unchecking the check box next to the button of the corresponding robot. Once the button is unchecked, the robot is no longer highlighted, and subsequent recommendations are displayed.

Suggesting, but not enforcing, robot selections lowers the risks associated with having an inaccurate model since the operator can potentially distinguish between successful and unsuccessful recommendations. However, even suggesting but not enforcing undesirable OAASs can potentially decrease performance due to automation bias [31] and mistrust [32].

VI. USER STUDY 2

We conducted a second user study in RESCU using eight-robot teams to evaluate 1) how well the model predicted the effects of $\mathcal{O}AAS^*$ on system effectiveness and 2) how well the various robot-selection mechanisms promoted successful OAASs. In this section, we outline the experimental protocol of this user study and then present and discuss the results.

A. Experimental Setup

The user study was a single factor within-subject study. The independent variable was the interface mode, which had three levels: Manual, Guided, and Auto. Each user managed an eight-robot team in each interface mode in the same RESCU scenarios used in the first user study. The order that the users saw each mode was counterbalanced to offset ordering effects.

The dependent variables for the study were the system score [1], including number of tokens collected and robots lost, and the participants' OAASs. We also collected subjective information, including the participants' perceived workload and their qualitative assessment of the robot selections recommended by the system.

The following procedure was followed for each subject. First, the subject was trained on each aspect of the original RESCU system and then performed a full practice scenario. Second, the subject was introduced to one of the three interface modes and performed a full practice scenario on that interface mode. Third, the subject performed a complete test mission using the

new interface. Fourth, the subject answered several subjective questions about their experience. The subject then repeated steps two through four using the other two modes.

Twelve undergraduate students, graduate students, and post-doctoral associates participated in the experiment. Six of the subjects were female and six were male. The subjects were between the ages of 19 and 32, with a mean age of 23.2 years. None of the subjects had prior experience with RESCU.

B. Results

This user study was designed to answer three questions. First, did the model accurately predict the effects of using $\mathcal{O}AAS^*$? Second, how did the different robot-selection mechanisms alter OAASs and which robot-selection mechanism was the most successful? Third, what were the participants' perceptions of the different selection mechanisms?

1) *Predictive Accuracy*: Since the Auto Mode uses $\mathcal{O}AAS^*$ to select robots for the operator to service, we compared the system effectiveness predicted by the model ($U(\mathcal{M}^*)$) with the system effectiveness observed in the Auto Mode to evaluate the model's ability to predict how changes in OAASs affect system effectiveness. The initial predicted performance did not match the system effectiveness observed in the Auto Mode. This was due in large part to a system upgrade to the computer used in the first user study, which resulted in the mouse scroll wheel being more sensitive than in the first user study. As a result, the average time an operator spent on visual search tasks decreased from 20 to 15 s (75%). As predicted by Fig. 6(d), this led to a substantial increase in system effectiveness. While relatively trivial, this experimental “mistake” illustrates how difficult it is to develop robust predictive models. Minor changes in the environment or the system itself can alter system effectiveness to the point that a predictive model becomes quite unreliable.

Fortunately, we can incorporate the new search times into \mathcal{II}^O to evaluate the model's ability to predict the effects of OAAS in the absence of this unintended system change. The remainder of the results in this paper include this revised \mathcal{II}^O . We also note that $\mathcal{O}AAS^*$ was still the best OAAS the model found for this revised system [Fig. 6(d)].

Comparisons of the observed number of tokens collected and robots lost in the Auto Mode with model predictions are shown in Fig. 8(a) and (b), where the model's predictions are labeled *Predicted*. The figures show that the model significantly overpredicted the number of tokens collected. Additionally, while the predicted number of robots lost was within the 95% confidence interval, the predicted value is still less than half the observed number of robots lost.

The inaccuracy of the predictions can be traced, in part, to an incorrect model of \mathcal{II} . Users took, on average, 1.6 s longer in the Auto Mode to service a robot than they did in the first user study (in which \mathcal{II}^O was modeled). The longer interaction times in the Auto Mode appear to have been caused by the altered robot-selection mechanism. Since users did not decide which robot they serviced, this sometimes caused them to spend more time gaining awareness of the selected robot's situation, which, as predicted by Billings [26] and

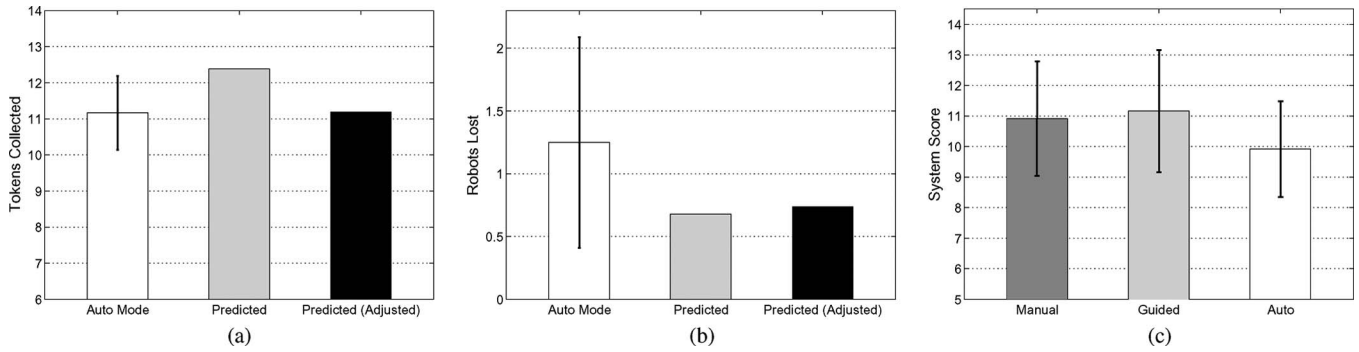


Fig. 8. Predicted system effectiveness compared with observed system effectiveness in the Auto Mode as measured by (a) tokens collected and (b) robots lost. (c) Observed system score in the user study.

Parasuraman *et al.* [27], led to longer interaction times than in the Manual Mode. Thus, the assumption of an unchanged II^O as stipulated in Section III-B was violated, thus compromising the model's predictive ability.

If we change the model's estimate of II to reflect the observed interaction times in the Auto Mode, we get the predictions labeled *Predicted (Adjusted)* in Fig. 8(a) and (b). These predictions for both tokens collected and robots lost fall well within the 95% confidence intervals, which indicates that if the model's estimates of II , NI , and switching times ST are accurate, the model can adequately predict the effects of OAASs on system effectiveness. However, when these estimates are incorrect, the model's predictions of system effectiveness are likely to be inaccurate, although the model still predicts that $OAAS^*$ would be a good OAAS if we attribute the increased interaction times to switching costs [Fig. 6(c)].

Thus, the model's reliance on observed data to estimate II implies that it cannot reliably predict the effects of changes in the OAAS. To be sufficiently robust, the model must somehow account for how human-robot interactions alter other aspects of the system. However, since the model does give reasonably good predictions given correct estimates of II , it has potential as a high-fidelity predictive tool if the effects of OAASs on II can be anticipated. While anticipating the effects of OAASs on II is a topic of future work, it is a nontrivial problem since the degrees of freedom in OAASs for command and control systems is extremely large.

2) *Effects of Selection Mechanisms on OAASs*: The second question addressed by this user study concerned how the robot-selection mechanism affected the system. While the model predicts that $OAAS^*$ was theoretically optimal given the set Ω and the modeled structures II^O , NI^O , and ST^O , following it did not lead to a higher score. In fact, users had higher scores in the Manual and Guided Modes, in which they often deviated from $OAAS^*$, than in the Auto Mode [Fig. 8(c)], although an analysis of variance shows no statistically significant difference in performance among the three modes ($F(2, 33) = 0.50$, $p = 0.609$).

That the Auto Mode did not outperform the other two modes can be linked, in part, to interaction time. Interaction times in the Auto Mode were, on average, about 1.6 s longer than in the Manual Mode and about 1.0 s longer than in the Guided Mode. As discussed previously, the model predicts that these

differences in interaction times have a substantial impact on the system's effectiveness [Fig. 8(a) and (b)].

Given the differences between $OAAS^*$ and $OAAS^O$ (the OAAS observed in the Manual Mode), it is interesting to study the OAASs used by operators in the Guided Mode, where users were free to follow or ignore the recommendations made by $OAAS^*$. While users did appear to follow some of the recommendations, they did not always do so. To see this, consider Fig. 9(a), which shows the percentage of time that the users' behavior corresponded with the model's recommended selections. In the Manual Mode, users' selections matched the model's recommendations about 50% of the time, which is just higher than random behavior (30% correspondence). Meanwhile, as expected, user selections in the Auto Mode corresponded with the recommendations more than 95% of the time. (Recall that users could turn off recommendations in the Auto Mode, which is why correspondence is not 100% in this condition.) In the Guided Mode, user selections corresponded to model recommendations about 60% of the time. Thus, the user's selections in the Guided Mode were more similar to those observed in the Manual Mode than in the Auto Mode.

The effects of recommendations in the Guided Mode are further shown in Fig. 9(b), which plots the correspondence of robot selections to recommendations in each minute. From the fourth minute to the end of the mission, the correspondence of users' selections to model recommendations in the Guided Mode mirrored that of the Manual Mode, except that the correspondence in the Guided Mode was shifted upward about 10%–20%. Thus, while users appeared to be biased by the recommendations, they tended to follow their own judgement.

3) *User Perceptions*: While observed system effectiveness and operator attention allocation are crucial metrics of any system, one cannot discount the role of user perception. While a system might produce good results, it will not likely become successful unless it gains user acceptance. A postexperiment questionnaire provided insights into the participants' attitudes toward the various robot-selection mechanisms.

Seventy-five percent of the users indicated that they did not feel that the recommendations were ideal or important. For example, one participant said that he completely ignored the recommendations in the Guided Mode because they were "confusing." Several other users commented that, while they did not always follow the recommendations in the Guided Mode, the recommendations sometimes drew their attention to

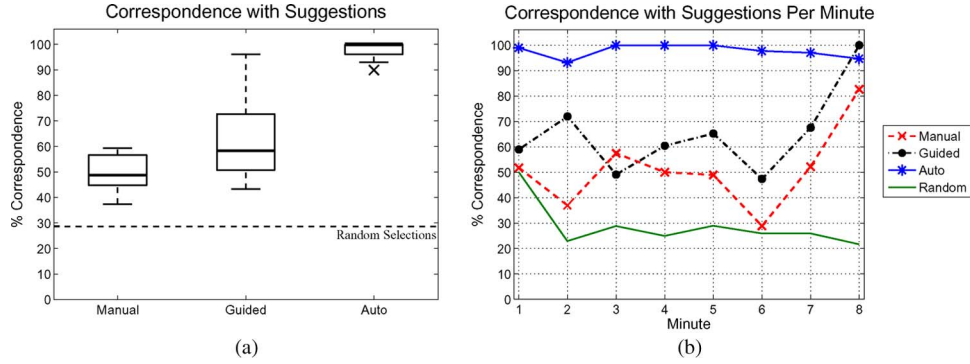


Fig. 9. (a) Average correspondence of robot selections with $\mathcal{O}AAS^*$. The dotted line marks random correspondence. (b) Correspondence per minute.

a robot that required servicing that they otherwise might have missed. Another participant determined that he did not need to follow the recommendations because the penalty for not doing so was not severe.

An analysis of how often users chose to “turn off” the recommendations in the Auto Mode shed additional light on users’ acceptance of the automation. Recall that the Auto Mode allowed users to “turn off” recommendations for a given robot. Once this was done, a subsequent set of recommendations was provided. In the Auto Mode, three of the 12 users (or 25%) chose to turn off various recommendations throughout the course of a mission. This implies that they did not believe the recommendations were worthy of consideration. Some of the nine subjects that never “turned off” recommendations indicated in postexperiment discussions that they would have “turned off” recommendations in the Auto Mode if they had remembered how to do so.

Participants were also asked to rank the three modes according to their preferences. Eight of the 12 participants (67%) in the study preferred the Guided Mode the most. They liked that the Guided Mode allowed them to service whichever robot they desired. Additionally, several users said that the recommendations alerted them of robots that needed to be serviced that they otherwise might have missed. In comparison, eight of the 12 users (67%) liked the Auto Mode the least. Many of the users expressed frustration that they were not allowed to select a robot that was not recommended.

However, several operators appreciated that the Auto Mode reduced their workload. Their intuition is validated statistically. After each mission in the study, each participant was asked to rank his mental workload during the mission on the scale one to five. An ordered logit model, specifically, proportional odds [33], shows a statistical difference in this measure of subjective workload ($\chi^2(2) = 6.98, p = 0.0305$). The odds of having higher subjective workload was lower for the Auto Mode compared with the Guided Mode ($\chi^2(1) = 9.84, p = 0.002$) and the Manual Mode ($\chi^2(1) = 5.46, p = 0.020$). Thus, while the Auto Mode did frustrate many of the users, it lowered their perceived workload.

VII. CONCLUSION AND FUTURE WORK

In this paper, we have discussed and analyzed operator attention allocation in HMRSs in which a human supervises

multiple semiautomated and time-critical tasks. In particular, we have focused on the role that predictive optimization models can play in improving operator attention allocation in HMRSs. Our experience highlights two important findings

- 1) Predictive models for computing and dictating optimal allocation of operator attentional resources in time-critical command and control settings are likely to fail.
- 2) Robust predictive models for identifying interventions for human attention guidance are necessary for the success of HMRSs.

The seemingly contradictory nature of these findings highlights the tricky balance that designers of HMRSs must achieve to help operators effectively manage attentional resources.

A. Optimization Models for Attention Allocation Will Fail

Despite our best efforts over a number of years, our predictive model was not sufficiently robust to predict the effects of altering the operator-attention-allocation strategy in HMRSs. While it is possible that more reliable models can be developed, the real world is sufficiently complex that other models will eventually fail, particularly in domains with significant exogenous uncertainty. In such situations, it is essential that the system be designed so that operators can adequately compensate for such failures.

Under the assumption of a correct estimate of the other aspects of the system, including \mathcal{II} , \mathcal{NI} , and \mathcal{ST} , our results indicate that such models give reasonably good predictions of the effects of OAASs on system effectiveness. While this is a good start, it appears that the assumption can be very difficult to meet. In many instances, the act of altering OAASs will induce changes in human-robot interactions, as demonstrated by user behavior in the Auto Mode of our study. Thus, while the modeling method potentially offers a framework for developing predictive models capable of predicting the effects of changes in OAASs, it is not sufficiently robust to reliably dictate successful OAASs. To be sufficiently robust, the model must anticipate how changes in one aspect of the system will affect other aspects of the system. This is an important, yet challenging, area of future work.

In addition to propagation of effects, predictive models for attention allocation must also account for other unforeseen changes in the system. For example, in our study, a system

upgrade produced a small change in the system that we, as system designers, did not anticipate but that had substantial impact on the effectiveness of the system. Given the relatively rapid changes that can occur in the software-intensive environment of multivariate HMRSs, as well as the highly dynamic environments and missions, this is a critical lesson learned.

The effects of dictating that operators follow *OAAS** resulted in a number of undesirable outcomes. First, dictating operator attention allocation did not produce an increase in system effectiveness. Second, operators became frustrated when they were forced to service robots they did not want to service. This finding means that dictating operator attention could have substantial negative effects if applied to safety-critical HMRSs operated over long periods of time. It also has implications for systems that provide an *a priori* set of strategies which require humans to subscribe to a predefined set of actions that may inadvertently overconstrain the operator and the system.

Thus, our results show that it is not enough to propose a computational model to successfully harness operator attentional resources in time-critical command and control HMRSs [10]–[13]. Rather, computational models must be evaluated in conjunction with human interactions with the system. It may be that such models are most useful for investigating human–computer architectures and developing operator decision support aids that guide as opposed to dictate human behavior.

B. Models for Guiding Operator Attention Are Necessary

Despite the brittleness of computational models in computing successful OAASs, we maintain that they are necessary to build successful HMRSs and other human–computer systems in real-world domains. While human judgment is critical in HMRSs, observed OAASs and the associated outcomes in RESCU clearly illustrate that operators are unable to effectively allocate their attention in time-critical command and control settings. Operators attempting to control or manage multiple robots or any heterogeneous set of tasks in time-pressured environments need assistance in allocating their attentional resources.

A recommendation system incorporating both human judgment and model recommendations appears to have potential for improving operator attention allocation in time-pressured HMRSs. While the user study showed no statistically significant difference in system effectiveness between the Auto and Guided Modes, users preferred the Guided Mode. This result mirrors the findings of Ruff *et al.* [29] in which management-by-consent was the preferred method in supervisory control of multiple unmanned vehicles. In the Auto Mode, users were often frustrated that they could not service the robot of their choice. On the other hand, while users believed that the suggestions made by the model were suboptimal in the Guided Mode, many of the users felt that they could still make good use of them. This is significant since highly robust predictive HMRS models are still likely to have moments of failure. In such situations, it is essential that operators can determine whether to follow the recommendations.

In short, rather than focus on determining “optimal” operator attention allocation, a more productive modeling effort involves using a model to identify OAASs that are “good enough” [34], ensuring that model assumptions are not overly constraining, and that the system allows operators flexibility in determining whether to follow or not automated recommendations.

REFERENCES

- [1] M. L. Cummings and P. J. Mitchell, “Automated scheduling decision support for supervisory control of multiple UAVs,” *AIAA J. Aerosp. Comput. Inf. Commun.*, vol. 3, no. 6, pp. 294–308, 2006.
- [2] M. L. Cummings, A. S. Brzezinski, and J. D. Lee, “The impact of intelligent aiding for multiple unmanned aerial vehicle schedule management,” *IEEE Intell. Syst.—Special Issue on Interacting with Autonomy*, vol. 22, no. 2, pp. 52–59, Mar./Apr. 2007.
- [3] B. D. Donmez, M. L. Cummings, and H. D. Graham, “Auditory decision aiding in supervisory control of multiple unmanned aerial vehicles,” *J. Human Factors Ergonom.*, vol. 51, no. 5, pp. 718–729, Oct. 2009.
- [4] B. Donmez, H. D. Graham, and M. L. Cummings, “Assessing the Impact of Haptic Peripheral Displays for UAV Operators,” *Humans Autom. Lab.*, MIT, Cambridge, MA, Tech. Rep. HAL2008-02, 2008.
- [5] M. Wieland, L. Griesser, and C. Kuendig, “Seismic early warning system for a nuclear power plant,” in *Proc. 12th World Conf. Earthquake Eng.*, 2000, pp. 1–8.
- [6] J. E. Veitengruber, G. P. Boucek, Jr., and W. D. Smith, “Aircraft Alerting Systems Criteria Study. Volume 1: Collation and Analysis of Aircraft Alerting Systems Data,” Dept. Transp., Federal Aviation Admin., Washington, DC, Tech. Rep. FAA-RD-76-222, 1977.
- [7] V. Ahlstrom and G. Panjwani, “Auditory Alarms in the Airway Facilities Environment,” Dept. Transp., Federal Aviation Admin., Washington, DC, Tech. Rep. DOT/FAA/CT-TN04/04, 2004.
- [8] J. D. Lee, J. D. Hoffman, and E. Hayes, “Collision warning design to mitigate driver distraction,” in *Proc. SIGCHI Conf. Human Factors Comput. Syst.*, 2004, pp. 65–72.
- [9] R. Parasuraman, P. A. Hancock, and O. Olofinboba, “Alarm effectiveness in driver-centered collision-warning systems,” *Ergonomics*, vol. 40, no. 3, pp. 390–399, Mar. 1997.
- [10] P. Scerri, K. Sycara, and M. Tambe, “Adjustable autonomy in the context of coordination,” in *Proc. AIAA 3rd Unmanned Unlimited Tech. Conf., Workshop Exhibit*, 2004, pp. 1–13.
- [11] K. Savla, C. Nehme, T. Temple, and E. Frazzoli, “On efficient cooperative strategies between humans and UAVs in a dynamic environment,” presented at the AIAA Guidance, Navigation Control Conf., Honolulu, HI, 2008, AIAA 2008-6841.
- [12] K. Savla, T. Temple, and E. Frazzoli, “Human-in-the-loop vehicle routing policies for dynamic environments,” in *Proc. IEEE Conf. Decision Control*, 2008, pp. 1145–1150.
- [13] H. Sohn, J. D. Lee, D. L. Bricker, and J. D. Hoffman, “A dynamic programming algorithm for scheduling in-vehicle messages,” *IEEE Trans. Intell. Transp. Syst.*, vol. 9, no. 2, pp. 226–234, Jun. 2008.
- [14] T. B. Sheridan and M. K. Tulga, “A model for dynamic allocation of human attention among multiple tasks,” in *Proc. 14th Annu. Conf. Manual Control*, 1978, pp. 343–358.
- [15] N. Moray, M. Dessouky, B. A. Kijowski, and R. Adapahtya, “Strategic behavior, workload, and performance in task scheduling,” *Hum. Factors*, vol. 33, no. 6, pp. 607–629, Dec. 1991.
- [16] J. P. Bliss and S. A. Acton, “Alarm mistrust in automobiles: How collision alarm reliability affects driving,” *Appl. Ergonom.*, vol. 34, no. 6, pp. 499–509, Nov. 2003.
- [17] M. A. Hearst, “Mixed-initiative interaction,” *IEEE Intell. Syst. Appl.*, vol. 14, no. 5, pp. 14–23, Sep./Oct. 1999.
- [18] T. B. Sheridan and W. L. Verplank, “Human and Computer Control of Undersea Teleoperators,” *Man-Machine Lab.*, MIT, Cambridge, MA, 1978, Tech. Rep.
- [19] J. W. Crandall, M. L. Cummings, and C. Nehme, “A predictive model for human–unmanned vehicles systems,” *J. Aerosp. Comput. Inf. Commun.*, vol. 6, no. 11, pp. 585–603, 2009.
- [20] J. W. Crandall and M. L. Cummings, “Identifying predictive metrics for supervisory control of multiple robots,” *IEEE Trans. Robot.*, vol. 23, no. 5, pp. 942–951, Oct. 2007.
- [21] S. Mau and J. Dolan, “Scheduling to minimize downtime in human–multirobot supervisory control,” in *Proc. Workshop Plan. Scheduling Space*, 2006, pp. 268–277.

- [22] S. Mau and J. Dolan, "Scheduling for humans in multirobot supervisory control," in *Proc. Int. Conf. Intell. Robots Syst.*, 2007, pp. 1637–1643.
- [23] H. Neth, S. S. Khemlani, B. Oppermann, and W. D. Gray, "Juggling multiple tasks: A rational analysis of multitasking in a synthetic task environment," in *Proc. 50th Annu. Meeting Human Factors Ergonom. Soc.*, 2006, pp. 1142–1146.
- [24] B. D. Donmez, C. Nehme, and M. L. Cummings, "Modeling workload impact in multiple unmanned vehicle supervisory control," *IEEE Trans. Syst., Man, Cybern. A, Syst., Humans*, vol. 40, no. 6, pp. 1180–1190, Nov. 2010.
- [25] D. B. Kaber and M. R. Endsley, "The effects of level of automation and adaptive automation on human performance, situation awareness and workload in a dynamic control task," *Theor. Issues Ergonom. Sci.*, vol. 5, no. 2, pp. 113–153, Mar. 2004.
- [26] C. E. Billings, *Aviation Automation: The Search for a Human-Centered Approach*. Mahwah, NJ: Lawrence Erlbaum Associates, 1997.
- [27] R. Parasuraman, T. B. Sheridan, and C. D. Wickens, "A model of types and levels of human interaction with automation," *IEEE Trans. Syst., Man, Cybern. A, Syst., Humans*, vol. 30, no. 3, pp. 286–297, May 2000.
- [28] L. Bainbridge, "Ironies of automation," *Automatica*, vol. 19, no. 6, pp. 775–779, 1983.
- [29] H. A. Ruff, S. Narayanan, and M. H. Draper, "Human interaction with levels of automation and decision-aid fidelity in the supervisory control of multiple simulated unmanned air vehicles," *Presence*, vol. 11, no. 4, pp. 335–351, Aug. 2002.
- [30] M. L. Cummings and P. J. Mitchell, "Predicting controller capacity in remote supervision of multiple unmanned vehicles," *IEEE Trans. Syst., Man, Cybern. A, Syst., Humans*, vol. 38, no. 2, pp. 451–460, Mar. 2008.
- [31] M. L. Cummings, "Automation bias in intelligent time critical decision support systems," in *Proc. AIAA 1st Intell. Syst. Tech. Conf.*, Chicago, IL, 2004, pp. 33–40.
- [32] J. D. Lee and N. Moray, "Trust, self-confidence, and operators' adaptation to automation," *Int. J. Human-Comput. Stud.*, vol. 40, no. 1, pp. 153–184, Jan. 1994.
- [33] M. E. Stokes, C. S. Davis, and G. G. Koch, *Categorical Data Analysis Using the SAS System*, 2nd ed. Cary, NC: SAS Publishing, 2002.
- [34] H. A. Simon, *The Sciences of the Artificial*, 3rd ed. Cambridge, MA: MIT Press, 1996.



Jacob W. Crandall received the B.S., M.S., and Ph.D. degrees in computer science from Brigham Young University, Provo, UT, in 2001, 2004, and 2006, respectively.

He was a Postdoctoral Associate with the Department of Aeronautics and Astronautics, Massachusetts Institute of Technology, Cambridge. He is currently an Assistant Professor with the Computer and Information Science Program, Masdar Institute of Science and Technology, Abu Dhabi, United Arab Emirates. His research interests

include human-machine interaction, multiagent learning, game theory, robotics, and interactive artificial learning.



Mary L. Cummings (M'03) received the B.S. degree in mathematics from the U.S. Naval Academy, Annapolis, MD, in 1988, the M.S. degree in space systems engineering from the Naval Postgraduate School, Monterey, CA, in 1994, and the Ph.D. degree in systems engineering from the University of Virginia, Charlottesville, in 2003.

A naval officer and military pilot from 1988 to 1999, she was one of the Navy's first female fighter pilots. Her previous teaching experience includes instructing for the U.S. Navy with the

Pennsylvania State University, University Park, and as an Assistant Professor with the Engineering Fundamentals Division, Virginia Polytechnic Institute and State University, Blacksburg. She is currently an Associate Professor with the Department of Aeronautics and Astronautics, Massachusetts Institute of Technology, Cambridge. Her research interests include human supervisory control, human-uninhabited vehicle interaction, bounded collaborative human-computer decision making, decision support, information complexity in displays, and the ethical and social impact of technology.



Mauro Della Penna was born in Naples, Italy, in 1984. He received the B.S. degree in aerospace engineering from University Federico II, Naples, in 2006 and the M.S. degree in dynamics and control of aerospace vehicles from the Delft University of Technology (TU Delft), Delft, The Netherlands, in 2009.

In 2008, he was involved in research at Massachusetts Institute of Technology, Cambridge concerning supervisory control. In the same year he was Student Assistant with TU Delft. In 2009, he was

part of a team at Entropy Control in San Diego that developed a prototype of a haptic steering safety system. He is currently the CEO of his family business Della Penna Autotrasporti s.r.l., Naples, that safely moves tourists from all over the world across European roads. In 2010, TU Delft patented the haptic system developed in his M.S. degree thesis.



Paul M. A. de Jong received the M.Sc. degree with a thesis in control space analysis of continuous-descent approaches for aircraft, from the Delft University of Technology (TU Delft), Delft, The Netherlands, in 2009. He is currently working toward the Ph.D. degree at the Faculty of Aerospace Engineering, TU Delft, where he is working on aircraft noise and gaseous emission reduction during approach.

As a graduate student, he spent four months as a Visiting Student with the Humans and Automation Laboratory, Massachusetts Institute of Technology, Cambridge, with Prof. M. Cummings, where he worked on predictive modeling of human supervisory control of unmanned vehicles. His interests include air-traffic management, software development, game theory, simulations, human-machine interaction, and aircraft control.