A PREDICTIVE MODEL FOR HUMAN-UNMANNED VEHICLE TEAMS

PROGRESS REPORT

J. W. CRANDALL
M. L. CUMMINGS

MASSACHUSETTS INSTITUTE OF TECHNOLOGY*

PREPARED FOR MIT LINCOLN LABORATORY

HAL2007–07

JULY 2007

Humans and Automation Laboratory

http://halab.mit.edu email: halab@mit.edu

*MIT Department of Aeronautics and Astronautics, Cambridge, MA 02139

Abstract

Advances in automation are making it possible for a single operator to control multiple unmanned vehicles (UVs). To effectively build such teams, predictive models must be developed that (a) describe the behavior of the human-UV team and (b) predict how alterations in team composition and system design will affect the team's overall effectiveness. In this report, we describe a predictive model for a human-UV team consisting of a single operator and multiple independent UVs. Through a series of user studies, we analyze the ability of this model to predict how changes in UV team size, interface design, and UV autonomy would alter the team's effectiveness. We also use the model to predict where human attention "should" be focused, and outline a (future) user study for validating these predictions.

1 Introduction

For the foreseeable future, unmanned vehicle (UV) technologies will require the assistance of human operators to perform complex and often dangerous tasks. Current UV platforms require multiple operators to control a single UV. This need for significant manpower is expensive and sometimes ineffective. As a result, it is desirable to invert this ratio so that a few operators control many UVs in order to (a) reduce costs, (b) extend human capabilities, and (c) improve human-UV system effectiveness. To achieve this goal, additional research must address many issues related to the human operator, the UVs, and the interactions between them.

When a human interacts with multiple UVs, many questions must be answered, including: How many UVs should there be in the team? What human-UV interaction methodologies are appropriate for the given human-UV team, mission, and circumstances? What autonomy levels should the UVs employ, and when (if ever) should changes in these autonomy levels be made? What aspects of a system should be modified to increase the team's overall effectiveness?

Models of the various components of human-UV teams can be constructed. Appropriate combinations of the "right" models can then be used to construct predictive tools that can answer these questions, even, potentially, for previously unmeasured situations or design modifications.

In this paper, we propose a set of such models as well as a method to combine the models so that we can predict various system characteristics of unmeasured conditions. This set of models is derived from a theoretical analysis of human, multi-UV teams, which we previously presented in [2, 3]. We review these theoretical foundations in Section 2. We then formally describe the models and a methodology for combining them to form a complete model of the team's behavior (Section 3). In Sections 4 and 5, we present a user study to (a) demonstrate how the models can be constructed and (b) show their predictive power. We then use the models to estimate the system effectiveness of other unobserved conditions (Section 6). We end with a discussion of our future research directions (Section 7).

Throughout the paper, we make the following assumptions. First, we assume homogeneous UV capabilities. However, the principles and theories discussed in this paper also apply to teams with heterogeneous UV capabilities, though additional issues will need to be considered for those teams. Second, we assume the team has only a single human operator. While realistic human-UV teams are likely to have multiple human operators, we make this assumption to simplify the problem. Third, we confine our discussion to human-UV teams in which the UVs perform independent tasks. This is also a very strong assumption, and it does not even completely apply to the simulated human-UV teams we consider in this paper. However, the independence assumption, again, simplifies the problem. Furthermore, our predictive models still functions reasonably well in many situations in which the independence assumption is not met.

2 Theoretical Foundations

The dynamics of a single human, multi-UV team under the previously mentioned assumptions are depicted in Figure 1. Associated with each UV are the two control loops of supervisory control [9]. In the upper control loop, the human interacts with the UV via the interface. We call measures of this function *interaction efficiency (IE)*. In the lower control loop, the UV acts in its world. We call the effectiveness of this behavior

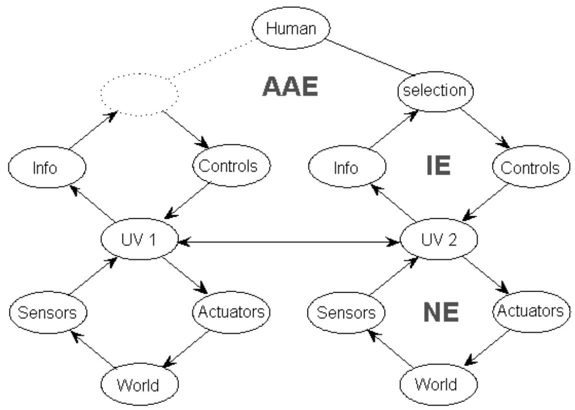


Figure 1: A single-human, multi-UV team can be separated into three separate components: attention allocation efficiency, interaction efficiency, and neglect efficiency.

neglect efficiency (NE). In sequential control, the upper control loop cannot be closed for all UVs in the team. As such, the human must decide which UV to interact with at any given time. *Attention allocation efficiency (AAE)* measures how effectively human attention is allocated between the UVs in the team.

Figure 1 also shows a connecting link between UVs in the team. This link captures the notion that interactions between UVs can have a significant impact on the team. This impact could be made manifest in measures of IE, NE, and AAE, or it could potentially be defined by a fourth function that needs to be measured in order to thoroughly evaluate the team. However, when UVs perform independent tasks (as we assume in this paper), this link has no effect on the behavior of the team.

While we separate IE, NE, and AAE into three separate aspects (or metric classes), they are not independent; a failure in one aspect of the system can very well lead to a failure in another aspect of the system. However, we believe that each system component can be modeled separately. The separate models can then be recombined to construct the behavior of the entire team.

3 Models and a Predictive Tool

In this section, we define behavioral models for each component of the team (IE, NE, and AAE). Second, we discuss how the models can be constructed by observing the human-UV team. Third, we describe how these individual models can be combined to form a predictive tool.

3.1 Behavioral Models

We use four different mathematical structures from the three metric classes to model the behavior of the agents (human and automated) in the team. The IE and NE components of the system can each be described with a single stochastic structure. We describe the behavior of the human with respect to attention allocation with two separate stochastic structures. We describe each model in turn.

3.1.1 A Model of Interaction Efficiency

Since IE measures how human-UV interactions change a single UV's behavior, we must model the behavior of the UV as it interacts with the human operator. This behavior can be modeled as a random sequence, which we refer to as *interaction impact* (II). Let T be the mission time in which the human-UV interaction begins, and let s be the UV's state at time T. Then, $II(s, T)$ is a random sequence that describes the sequence of the UV's future states while the interaction persists. Thus, a sample drawn from $II(s, T)$ is a sequence of UV states for all $T \leq t \leq T + \tau$, where τ is the length of the human-UV interaction.

This model of UV behavior relies on a number of assumptions. First, the model makes the first-order Markov assumption that the random sequence is well defined given (a) the time that the interaction began and (b) the UV's state at time T (it does not depend on past UV states). At least for the limited situation we consider in this paper, this assumption does not appear to significantly detract from the predictive ability of the model (as demonstrated by the results presented in Sections 5). However, if desired, the first-order assumption could be exchanged for an n-order Markov assumption to potentially improve the model's accuracy, though doing so drastically increases the amount of data we need in order to model the process. We note that the accuracy of the Markov assumption is determined in large part by the set of UV states we consider.

A second assumption of this model is that UV behavior does not depend on factors such as the mission task being performed, the number of such tasks the team is asked to perform, and the complexity of the world in which the UVs operate. Obviously, variations in any of these aspects are likely to cause alterations in II [1]. Thus, our model applies only to the specific situation being measured, though in many situations, we should be able to adequately estimate how circumstances will vary the behavioral model. We estimate some of these alterations for particular situations in Sections 5 and 6 (though we leave validation of these estimates to future work).

Third, the time variable T is used to model time-critical missions. This variable makes provisions for the fact that agent behaviors may change as mission time advances. When a mission is not time-critical, we anticipate that this variable could be discarded, thus simplifying the models.

3.1.2 A Model of Neglect Efficiency

The behavior of a UV in the absence of human attention can also be modeled as a random sequence, which we call *neglect impact* (NI). Let T be the time the user stopped interacting with the UV, and let s be the state of the UV at that time. Then, $NI(s, T)$ stochastically describes the infinite sequence of UV states thereafter in the absence of future human-UV interactions. The same modeling assumptions made for II apply to NI.

3.1.3 Models of Attention Allocation Efficiency

After the human completes an interaction with a UV, (s)he must determine which UV to attend to next. Relevant behavioral characteristics of this process include (a) how long the human takes to make this decision (called *switching time* (ST)) and (b) which UV the human selects (called *selection strategy* (SS)). Let $S = (s_1, \cdots, s_N)$ be the *joint state* of the N UVs in the team, where s_i is the state of UV i. Then, switching time is modeled by the distribution $ST(S, T)$, where T is the time the previous human-UV interaction ended. The selection strategy is modeled by the distribution $SS(S, T)$, where T is the time of the selection.

Under these modeling assumptions, switching time is a combination of two kinds of time periods. First, it consists of the time it takes for the operator to (a) orient to the circumstances of the UVs in the team, (b) select a UV to service next, and (c) carry out the necessary steps to select that UV. Second, it consists of any time in which the operator chooses to not service any of the UVs in the team when (s)he believes none of the UVs need servicing. In such situations, the operator simply monitors the UVs' progress, etc.

These models infer similar modeling assumptions to those we made for II and NI.

3.2 Building the Models

Rather than construct parametric models of II, NI, ST, and SS, we approximate them with the raw data samples collected from observing the operations of a human-UV team. A probability distribution over these samples for each (S, T) pair provides a description of the behavior of the team with respect to each of these system components. We now define these probability distributions for each structure.

3.2.1 Constructing $II(s,T)$

Each sample $x \in II$ is a particular sequence of UV states observed during a human-UV interaction. Each sample x is labeled with the tuple (x_T, x_s), where x_T was the time in the mission that the human-UV interaction began and x_s was the UV's state at time x_T. The labels are used to define a probability distribution over the samples for each tuple (s,T), which in turn defines the random sequence $II(s,T)$. Formally, the probability that sample x will be "chosen" is

$$Pr(x|s,T) = \frac{w_x^{II}(s,T)}{\sum_{y \in II} w_y^{II}(s,T)}, \tag{1}$$

where $w_x^{II}(s,T)$ is a weight given by

$$w_x^{II}(s,T) = \begin{cases} exp\left(\frac{-(x_T - T)^2}{2v^2}\right) & \text{if } s = x_s \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

The parameter v is a positive constant.

3.2.2 Constructing NI

The stochastic behavior of a UV while being neglected can be simulated by deriving a probability distribution over data samples in the same way as we described for II. In practice, this is not necessary, since we can use the sample $y \in NI$ that succeeded the sample of II that was selected. In the case that the Markov assumption holds, these two methodologies are mathematically equivalent. In the case that the Markov assumption is violated, selecting the sample that followed the interaction sample provides a more accurate estimate of UV behavior.

As in II, a sample in NI consists of a sequence of UV states. Each sample in NI defines the behavior of the UV in the absence of human-UV interactions. However, since human-UV interactions do occur, each observed sample is incomplete. Thus, we must estimate how the remaining subsequence of UV states would play out had the interaction not occurred. For example, suppose the user chose to interact with a UV after the UV had been neglect for τ seconds. Then, the UV's states in the sample are fully observable for the first τ seconds, but must be estimated for time $\tau + 1$ onward. In many instances, estimates of the unobserved portion of the sample can be estimated reasonably well, particularly when the dynamics of the environment are well understood (as in the case of the simulated test-bed we consider in the next section). However, in complex real world situations, these estimates are more difficult to make. While this is an important issue, it must be solved in a case by case manner. Thus, we do not fully address this issue in this paper.

3.2.3 Constructing SS

A sample $x \in SS$ is a selection of a UV (made by the operator) at a specific mission time x_T when the joint state was x_S. Since these samples consider the joint state of the UVs rather than just an individual UV's state, the state space can be quite large. As a result, there are often few (or no) relevant samples for each (S,T) pair.

In order to reduce the state space, we adopt an alternate representation of joint state. When UVs are homogeneous in capabilities, it does not matter which UV is in which state, but only how many UVs are in each state.[1] Thus, we can re-specify joint state as $S = (\sigma_1, \cdots, \sigma_M)$, where σ_i is the number of UVs in state i (mapping the states to the real numbers), and M is the number of UV states. This representation significantly reduces the state space. We can further reduce the state space by considering only the boolean representation of S. In this representation, we consider only if there is a UV in state s or not. Formally,

[1] Depending on how UV states are defined, this statement can hold true for the heterogeneous case as well.

let $S(\sigma_j)$ denote the j^{th} component of the joint state S and let $\kappa_j(S)$ be a boolean representation of $S(\sigma_j)$. Formally,

$$\kappa_j(S) = \begin{cases} 1 & \text{if } S(\sigma_j) > 0 \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

This alternate representation of joint state significantly reduces the joint state space. Furthermore, we can successfully interpolate between joint states by defining a similarity metric between joint states. Let $\Phi(S_1, S_2)$ denote the similarity of joint state S_1 to joint state S_2. Let q be the probability that, over all $j \in [1, M]$, if $S_1(\sigma_j) > 0$, then $S_2(\sigma_j) > 0$. Thus, q is given by

$$q = \frac{\sum_{i=1}^{M} \kappa_i(S_1)\kappa_i(S_2)}{\sum_{j=1}^{M} \kappa_j(S_1)} \tag{4}$$

Then, formally, the similarity of S_1 to S_2 is given by

$$\Phi(S_1, S_2) = q^C \tag{5}$$

where C is a positive constant. Higher values of C make differences in joint states more salient.

The weight that a sample contributes to the probability distribution $SS(S, T)$ is a combination of similarity in the joint states S and x_S and the times T and x_T. Formally,

$$w_x^{SS}(S, T) = \Phi(S, x_S) \cdot exp\left(\frac{-(x_T - T)^2}{2v^2}\right). \tag{6}$$

Thus, the probability that a sample $x \in SS$ will be selected given a joint state S and a mission time T is:

$$Pr(x|S, T) = \frac{w_x^{SS}(S, T)}{\sum_{y \in SS} w_y^{SS}(S, T)}. \tag{7}$$

In addition to the assumptions we have previously discussed, this definition of the distribution $SS(S, T)$ makes one other limiting assumption. This assumption is that it does not matter how many UVs are in a given state, as long as there is at least one. This assumption is obviously limiting, since the operator's selection strategy could definitely vary based on the number of UVs that are in a joint state. We leave this issue to future work.

3.2.4 Constructing ST

A sample $x \in ST$ is a time measurement taken at a specific mission time x_T when the joint state was x_S. The time measurement, however, is not completely straight-forward (as it was observed) due to the fact that the switching time is composed of both time to select a UV and monitoring time. When switching time is comprised of only the former event, then the observed elapse in time between the end of the last interaction and the beginning the next interaction is sufficient to define the switching time. However, in situations in which the user chooses to monitor the UVs before making a selection, we must consider the event (or events) that separate monitoring time from selection time. Thus, in this latter case, the time measurement is a function of elapsed time from some event, and not the elapsed time since the last interaction was completed. The mechanism for identifying an event that triggers the switch from monitoring the team to selecting a UV is context dependent. We describe this mechanism for a particular situation (the human-UV team we consider in this paper) in Section 4.3.2.

The probability distribution over samples in ST for a joint state S and mission time T is defined similarly to that of SS.

1. Set $T = 0$, determine NI sample for each UV
2. Repeat
   (a) $T = T + t_{\text{switch}}$, where t_{switch} is drawn from $ST(S, T)$
   (b) Update the joint state S
   (c) Select a UV k to service using $SS(S, T)$
   (d) Select a sample x from $II(s, T)$ for UV k
   (e) $T = T + \tau$ (τ is the length of sample x)
   (f) Update the joint state S
   (g) Select a sample x from $NI(s, T)$ for UV k

Algorithm 1: Outline of the discrete event simulation. The distributions $II(s, T)$, $NI(s, T)$, $SS(S, T)$, and $ST(S, T)$ are defined in Section 3.2.

3.3 A Model of Team Behavior

The models we have just described can be used to estimate the behavior of a human-UV team using a discrete event simulation, which is outlined in Algorithm 1. In step 1 of the algorithm, system variables are initialized. In step 2(a), the human takes time t_{switch} to determine which UV to service, during which time we observe the UVs' state transitions. In steps 2(c)-(d), UV k is selected for servicing and a sample is drawn from $II(s, T)$. UV k acts according this sample for the next time period of length τ, while the other UVs continue to act according to their samples of NI. When the interaction is completed, a behavior sample for UV k is drawn from $NI(s, T)$. The process repeats until the human-UV team ceases operating.

4 A User Study

To demonstrate the ability of these models (and the associated method for combining them), we conducted a user study. From the user study, we constructed the aforementioned models. These models can be used to estimate the behavior of the system when various components of the system change. In this section, we describe the software test-bed and the experimental procedure used in the user study. In subsequent sections, we analyze the predictive ability of the models. We also describe the system specific parameters needed to generate the aforementioned models.

4.1 Software Test-bed

The software test-bed is called RESCU (*R*esearch *E*nvironment for *S*upervisory *C*ontrol of *U*nmanned-vehicles). We describe the RESCU in three parts: the mission performed by the human-UV team, the human-UV interface, and the UVs' behavior.

4.1.1 Mission

Across many mission types, operators of human-UV teams assist in performing a set of common tasks. These common tasks include mission planning and re-planning, UV path planning and re-planning, UV monitoring, sensor analysis and scanning, and target designation. Each of these tasks can be performed using various levels of automation [10].

In designing RESCU, we sought to capture each of these tasks in a time-critical situation. The human-UV team (which consisted of the participant and simulated UVs) was assigned the task of removing as many objects as possible from the maze in an 8-minute time period. At the end of 8-minutes, the maze "blew up," destroying all UVs and objects that remained in it. Thus, in addition to collecting as many objects as possible, a user needed to ensure that all UVs were out of the maze when time expired.

An object was removed from the maze (i.e., collected) using a three-step process. First, a UV moved to the location of the object (i.e., target designation, mission planning, path planning, and UV monitoring).

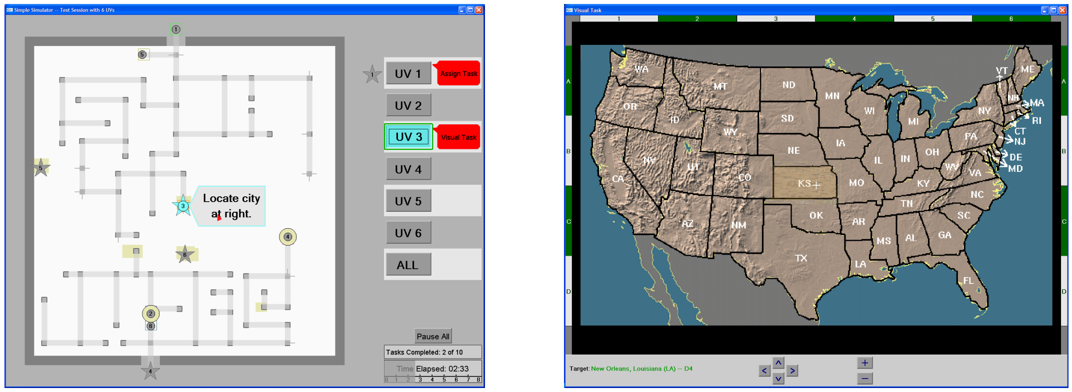


Figure 2: Two displays used in the experiment (on separate monitors). The display at left shows the map of the maze, the locations of the UVs, their destinations, the locations of the objects to be gathered, etc. The display at right shows the interface for the abstract visual task.

Second, the UV "picked up" the object (i.e., sensor analysis and scanning). In the real world, performing such an action might require the human operator to assist in identifying the object in video or laser data. To simulate this task, we asked users to identify a city on a map of the mainland United States using *Google*™ *Earth*-style software. Third, the UV carried the object out of the maze via one of two exits.

The mission also had the follow details:

- At the beginning of the session, the UVs were positioned outside of the maze next to one of two entrances.
- The form of the maze was initially unknown. As each UV moved in the maze, it created a map which it shared with the participant and the other UVs.
- The objects were randomly spread through the maze. Initially, the human-UV team could only see the positions of six of the objects. In each minute of the session, the locations of two additional objects were shown. Thus, there were 22 possible objects to collect during a session.
- The participant was asked to maximize the following objective function:

$$Score = ObjectsCollected - UVsLost, \tag{8}$$

  where $ObjectsCollected$ was the number of objects removed from the area during the session and $UVsLost$ was the number of UVs remaining in the area when time expired. This objective function makes two assumptions. First, the value of a UV is equivalent to the value of an object. Obviously, altering these utilities should cause alterations in team behavior. Second, all objects have the same priority. In many applications, different objects will have different priorities, and these priorities could vary over the course of a mission. Such variations in priority can be easily implemented in RESCU. However, to make it easier to validate the model, we assign all objects the same constant priority. We leave the more complex case (in which priorities differ and vary) to future work.

4.1.2 Interface

The human-UV interface for RESCU was the two-screen display shown in Figure 2. On the left screen, the map of the maze was displayed, along with the positions of the UVs and (known) objects in the maze. The right screen was used to locate the cities.

The participant could only interact with one UV at a time. When a user desired to interact with a particular UV, (s)he clicked a button on the interface corresponding to that UV (labeled UV1, UV2, etc.).

Once the participant selected the UV, (s)he could direct the UV by designating a goal location and modifying the UV's intended path to that goal. Designating a goal for the UV was done by dragging the goal icon corresponding to the UV in question to the desired location. Once the UV received a goal command, it generated and displayed the path it intended to follow. The participant was allowed to modify this path using the mouse.

To assist the operator in determining which UVs needed input, warning indicators related to a particular UV were displayed next to its corresponding button. There were four kinds of warning indicators:

- Assign Task Indicator – Displayed when the UV had arrived at its goal and needed the user's attention to proceed.
- Visual Task Indicator – Displayed when the UV had arrived at an object it was designated to collect.
- Time Warning – Displayed during the last minute of a session if the UV was still in the maze and had not been told to leave the maze.
- Deliver Object Indicator – Displayed to indicate that the UV had picked up an object, but had not been directed to leave the maze.

If no status or warning was reported, the system determined that the UV was satisfactorily progressing on its assigned task.

4.1.3 UV Behavior

The UVs' map of the maze took the form of an undirected graph. Each edge of the graph was an ordered pair (u, v) representing a connection between vertices u and v in the graph. Associated with each edge was a weight indicating the cost for a UV to move along that edge. Since the maze was not fully known, a UV had to choose between (a) moving along the shortest path of the known maze to its user-specified goal and (b) exploring the unknown portions of the maze in hopes of finding a shorter path. To make this decision, a UV assumed that an unmapped edge from a known vertex v led directly to the goal position with a cost equal to the Manhattan distance from v to the UV's goal, plus some cost of exploration (C_E). The UV then used Dijkstra's algorithm on the resulting graph to determine the path it intended to follow.

Using this approach, the constant C_E determines the degree to which the UVs explore the unknown maze. Higher values of C_E result in less exploration. We used a small value of C_E for a UV that was searching for an object, and a higher value for a UV that was carrying an object. Since users sometimes felt that the resulting behavior was undesirable, they were allowed to modify a UV's path if they desired.

4.2 Experimental Procedure

Following training on all of the functions of the system and after completing a comprehensive practice session, each user participated in six eight-minute sessions. In each of the first four sessions, a different number of UVs (2, 4, 6, or 8) were allocated to the team. In the last two sessions, the experimental conditions (i.e., number of UVs in the team) of the first two session were repeated. The conditions of the study were counter-balanced and randomized. The participants were paid \$10 per hour; the highest scorer also received a \$100 gift certificate.

Twelve people (one professor, ten students, and one other person from the community) between the ages of 19 and 44 years old (mean of 27.5) participated in the study. Of these twelve participants, eight were U.S. citizens, two were Canadian, one was Hispanic, and one was Egyptian. Three of the participants were female and nine were male.

4.3 Model Parameters

The model definitions described in Section 3 require a number of instantiation specific parameters. We specify values for these parameters in this subsection, including UV states, event triggers (for modeling switching times), and weight parameters.

4.3.1 UV State Specifications

As we mentioned previously, the models requires a specification of states the UVs can be in. This set of states should have three properties. First, it should adequately capture and distinguish between the possible situations the UVs can encounter. Second, a set of states should adequately capture and distinguish between the users' perceptions of the UVs' situations. Third, the number of UV states should be as small as possible in order to make the model efficient.

We make two observations about the interaction between these three desirable properties. First, a UV's status may differ significantly from the user's perception of its status. Thus, it may be desirable to identify two sets of states: one for UV behavior (II and NI) and one for user behavior (SS and ST). Second, a larger set of states can often differentiate between the various situations encountered by the human-UV team than a smaller set of states. Therefore, a trade-off exists between making the model computationally efficient (via less states) and accurate (via more states).

Our work in identify sets of states is still in progress. In this paper, we describe a particular set of states that we have found to be relatively useful (both for user perception and UV behavior), though not completely correct. One main focus of future work (to be carried out within the next five months) will be to identify more satisfactory sets of UV states.

To this point, we have considered a UV to be in one of 13 states at any given time. A UV could be outside the maze, in the maze on an object, lost (the UV could not possibly leave the maze before time expired), in one of six states when searching for an object in the maze, or in one of four states when exiting the maze carrying an object. UV states when searching for an object or exiting the maze depended on (a) whether the UV was taking the shortest path to its goal and (b) how long it would take the UV to reach its goal.

For the purpose of this report, we give names to each of the thirteen possible UV states and briefly describe each. The names of the states are based on the letters 'O' (for outside), 'X' (for exiting the maze), 'R' (for ente*r*ing the maze), and 'V' (the UV was on an object waiting for the user to perform the *v*isual task).

- **O_ST** – A UV that was outside the maze.
- **XC_ST** – A UV that was within 5 seconds of leaving the maze carrying an object.
- **V_ST** – A UV that was located at the position of the object that the user had assigned to it.
- **RC_ST** – A UV that was within 5 seconds of arriving at the position of the object that the user had assigned to it.
- **XS_ST** – A UV that was (a) in the maze, (b) not moving (i.e., sitting), and (c) carrying an object.
- **RS_ST** – A UV that was (a) in the maze, (b) not moving (i.e., sitting), and (c) searching for an object..
- **RSB_ST** – A UV that was (a) searching for an object, (b) not taking a direct course to its user-specified goal (i.e., *B*ad behavior), and (c) would arrive at the location of its user-specified object within 20 seconds (as designated by the 'S' for short distance).
- **RMB_ST** – A UV that was (a) searching for an object, (b) not taking a direct course to its user-specified goal (i.e., *B*ad behavior), and (c) would arrive at the location of its user-specified object in 20 to 60 seconds (as designated by the 'M' for medium distance).
- **RLB_ST** – A UV that was (a) searching for an object, (b) not taking a direct course to its user-specified goal (i.e., *B*ad behavior), and (c) would arrive at the location of its user-specified object in 60 seconds or more (as designated by the 'L' for long distance).
- **RG_ST** – A UV that was searching for an object and taking taking a direct course (shortest route) to its user-specified goal (i.e., *G*ood behavior).

- **XG_ST** – A UV that was carrying an object and was taking the shortest path out of the maze (i.e., *G*ood behavior).
- **XB_ST** – A UV that was carrying an object, but was not taking the shortest path out of the maze (i.e., *B*ad behavior).
- **L_ST** – A UV that could not possibly leave the maze before time expired (i.e., the maze blew up).

This set of UV states seems to do a relatively good job of distinguishing between the situations the UVs encounter while keeping the number of states relatively small. Research is underway to determine what other state spaces would more accurately represent users' perceptions of the UVs' states.

4.3.2 Trigger Events

In order to model switching times from user data, we must define so called trigger events, or events that tend to make the user change from monitoring the UVs to selecting a UV to service (Section 3.2.4). For the user study in question, a trigger event is a change in a UV's state while the user is between interactions (i.e., after the user finishes servicing a UV and before (s)he selects another UV to service).

4.3.3 Weight Parameters

Equations 2, 4, and 6 require values for two free parameters: v and C. $v = 30$ seconds and $C = 40$ were selected experimentally to provide good fit to the data obtained from this first user study (from the 12 subjects). However, experimental comparisons with other values of v and C show relatively little change in the estimates calculated by our models in RESCU. In future work, we will study the impact of these variables in more detail.

5 Results

In-depth analysis of empirical results describing the success of the human-UV teams in this user study are given in [3]. In this section, we analyze the ability of our models (generated by data from this user study) to make predictions about the human-UV team's effectiveness in RESCU. The models are constructed from observations of the human-UV team in a single condition (called the *measured condition*). The predictive models must then predict the system's effectiveness in *unmeasured conditions*. We consider situations in which conditions change based on (a) the number of UVs in the team (Section 5.2) and (b) the available decision support tools and UV artificial intelligence (Section 5.3). Before doing so, however, we analyze the ability of the models to duplicate the behavior of the human-UV team in the measured condition. We do so by comparing the estimates of system effectiveness made by Algorithm 1 with the actual average system effectiveness observed in the user study.

5.1 Duplicating Observed Results

Figure 3 compares the estimates made by Algorithm 1 (an average of 10,000 trials) to the actual observed results from the user study (the mean and standard error are shown). The figure shows comparisons of number of object collected (Figure 3a) and number of UVs lost (Figure 3b). The y-axis is the effectiveness of the UV team and the x-axis is the observed condition (with respect to UV team size).

For all team sizes, estimates of number of objects collected are within the standard error of actual number of objects collected. In the case of number of UVs lost, the estimates also compare favorably with the observed results, though the estimates are outside of the standard error for 2- and 8-UV teams. That the estimates made by our predictive models are reasonably close to the actual observed results indicates that, indeed, the models do capture the behavior of the human-UV team reasonably well.

While the estimates are reasonably good, they are in all cases pessimistic. That is, the models predict fewer number of objects collected and more UVs lost than the actual observed means show. This seems to

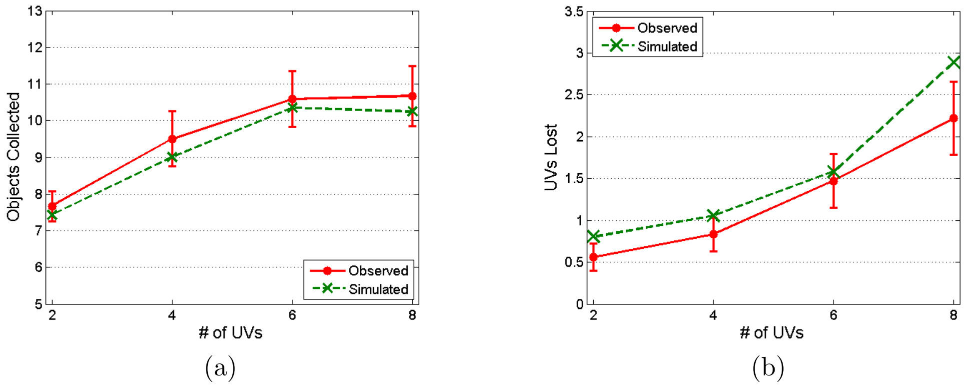


Figure 3: Average (a) number of objects collected and (b) number of UVs lost simulated values plotted against average observed values from the user study.

indicate that the models are not capturing all aspects of the agents' (both human and UV) behaviors. This is likely due to (a) the set of UV states we are considering and (b) the simplifying assumptions of joint state. However, the differences between the estimates and the observed results are relatively small. As such, we conclude that the models do a reasonably good job of capturing the team's behavior (though future work in these areas is still necessary).

5.2 Predicting Effects of Changes in UV Team Size

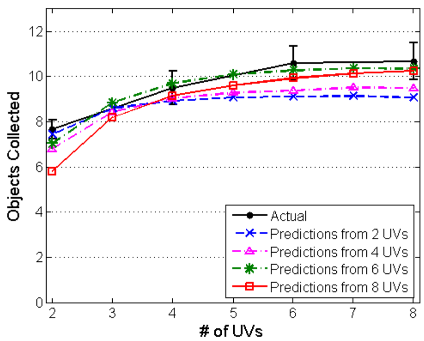
One way to improve the effectiveness of a human-UV team is to alter the number of UVs in the team. Additionally, the number of UVs in the team will often change due to UV failures, UVs being passed between teams, etc. Thus, knowing how the effectiveness and behavior of the team changes based on the number of UVs in the team is desirable.

These estimates can be generated by (a) constructing the models II, NI, SS, ST from observations the human-UV team in the measured condition and (b) running Algorithm 1 using these models using the the number of UVs in question (rather than the number of UVs in the measured condition). The resulting estimates are summarized in Figure 4. These results are presented and discussed in [3]. As such, we do not replicate these results herein. We note only that, while not perfect, the resulting predictions do capture the trends of the observed results. Predictions tended to vary based on the measured condition, however, since the number of UVs in the measured condition altered the neglect efficiency of the UVs and the selection strategies of the users. Future work should address how to predict these changes so as to provide improved predictive tools.

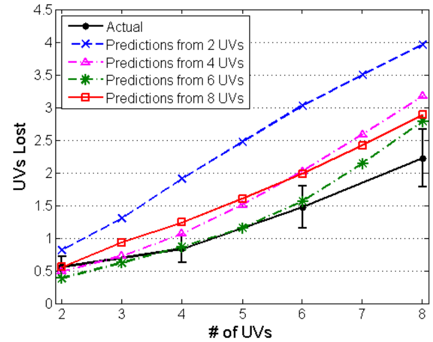
5.3 Predicting Effects of System Design Modifications

In addition to changing the number of UVs in a team, a human-UV team's effectiveness can potentially be improved by altering the human-UV interface or increasing the capabilities of the UVs (both in software and hardware). In this section, we consider the ability of our models (formed by observing results from the 12-subjects in the user study described in Section 4) to predict the effectiveness of such modifications. We compare the resulting predictions with the observed results from a second user study for 4- and 8-UV teams.

In this section, we consider only modifications to components of the system associated with IE and NE. That is, we consider how altering the individual UVs' behaviors (via altered UV autonomy and user input) affect the team's overall effectiveness. Figure 5a illustrates conceptually how such design modifications can affect a single UV's instantaneous performance. In the figure, the y-axis is the UV's instantaneous

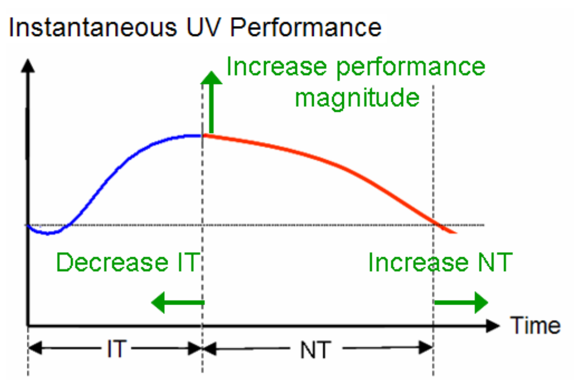


(a)

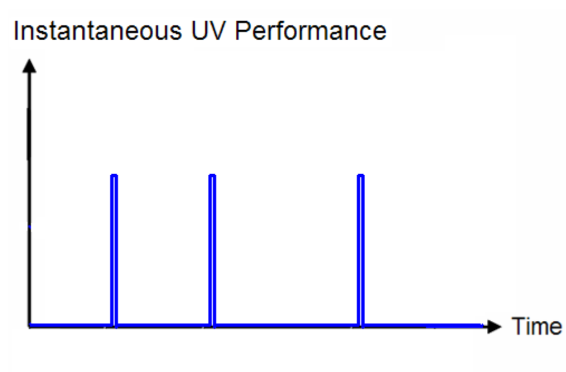


(b)

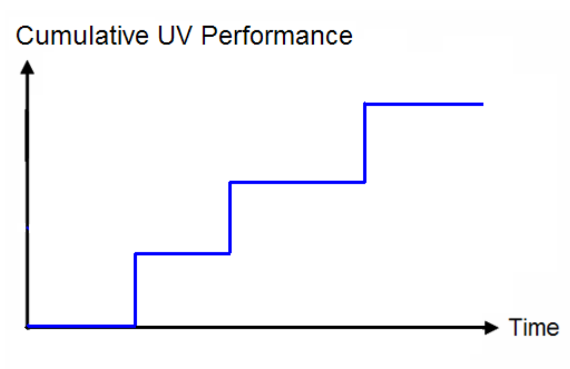
Figure 4: Predictions of objects collected (left) and robots lost (right) compared to the sample means obtained in the user study. Actual refers to the mean (and standard error) of observed scores in the user study and *Predictions from N Robots* shows the predictions (for all team sizes shown along the x-axis) from the N-robot measured condition. Each prediction is the average of 10,000 samples. This figure was presented in [3].



(a)



(b)



(c)

Figure 5: (a) Diagram of how $II(s,T)$ and $NI(s,T)$ of UVs can be modified to potentially improve a human-UV team's overall effectiveness. The blue curve represents a hypothetical expected value of a UV's performance while interacting with the human user. The red curve represents a hypothetical expected value of the UV's performance while being neglected. (b) UV instantaneous performance as a sum of delta functions. (c) Cummulative UV performance as a step function.

performance, which is determined by mapping the UV's behavior to some performance function[2]. On the left side of the time line (time t_0), the user begins interacting with the UV. In such situations, the UV's expected performance will (hopefully) increase until interactions cease. Thus, the blue portion of the curve in the figure represents the UV's average performance during human-UV interactions. Once the user ceases interacting with the UV at time $t_0 + IT$ (where IT is *interaction time*), the average performance of the UV decreases over time. This notion is depicted on the right half of the curve, where the average UV performance during neglect times is plotted in red. The variable NT (for neglect time) denotes the amount of time it takes for the UV's average performance to drop below some acceptable performance threshold [8].

Figure 5a illustrates three ways that the system's effectiveness can potentially be improved. First, it can be increased by decreasing IT [1]. This allows more time for the user to attend to other important tasks. This is likely to be achieved by improving the user interface to allow the operator to perform her/his tasks with respect to that UV more quickly. Alternately, the team's effectiveness could be improved by increasing each UV's NT [1]. This is done by increasing a UV's ability to perform effectively in the absence of user assistance, such as giving the UVs the ability to generate their own goals when the user does not provide direction. A third way to improve the system's effectiveness is to increase the magnitude of the curve. This could be achieved by equipping the UV with better hardware, control software, and artificial intelligence, or altering the system so that the human user gives better input to the UV. However, we note that such a modification might also have the effect of increasing or decreasing IT and NT. Thus, the performance benefits from "increasing the magnitude" of such a curve may be unclear.

In the previous discussion, we assumed that UV performance can be measured continuously. In practice, this can be difficult to do since "rewards" are only received at discrete points in time (e.g., when a target is located). In such cases, instantaneous UV performance can be expressed as a sum of delta functions, as shown in Figure 5b. Such a performance function means that cumulative UV performance can be expressed as a step function (Figure 5c). One future area of research will be defining UV performance models that are linked to effects based outcomes (such as targets imaged, etc.).

We note that changes in performance magnitude can be expressed in Figure 5b as increases in the height of the delta function (e.g., finding targets with higher priorities) or the frequencies of the delta functions (e.g., finding more targets). These increases in performance magnitude would be expressed in Figure 5c as higher and more frequent steps in the performance function.

In this section, we consider design improvements designed to decrease IT and increase NT.[3] We propose methods for achieving both of these objectives and assess the ability of our models to predict the behavior of the UV team under these modifications. We do so by hypothesizing how the individual models we constructed by observing the *original* human-UV team (from the first user study described in Section 4) will change given these design modifications. We then run Algorithm 1 using the resulting models, and compare these results to the results of a second user study. We leave considerations of methodologies for increasing the magnitude of UV performance to future work.

In the remainder of this section, we describe two system modifications and predict how these modifications will affect the behavioral models of the various entities in the team. We then describe a second user study in which users play RESCU with the modified systems. Finally, we compare the results of the user study to the estimates generated by the models.

5.3.1 Adding Automatic Target Recognition (ATR)

Several participants in the original (first) user study indicated that they thought that the team's effectiveness was largely dependent on how long it took them to perform the locate-a-city task (to "pick up" an object).

[2]In this representation, a UV's performance at any time t is a function of how well its current actions translate into current and future rewards (see, for example, [1]). In practice, this performance function could be determined by some credit assignment algorithm. In our models, we do not use such a mechanism as we consider individual UV performance conceptually (and not in practice)

[3]We leave analysis of system modifications designed to increase individual UV performance magnitude to future work. This future work should consider increasing UV performance by (a) performing subtasks more effectively and efficiently and (b) performing subtasks linked to tasks with higher priorities.

Thus, one potential system modification is to develop decision support that can help the user perform this task more efficiently. Potential decision support for this task would include an automated visioning system to assist the user in locating the cities. This visioning system could suggest to the human operator potential candidate cities on the map. Accurate suggestions made by the visioning system could significantly reduce the time it takes the human to perform this task.

In our implementation, an automatic target recognition (ATR) algorithm suggested two candidate cities on the map to the user. These suggestions were imposed on the map as blinking red boxes around the suggested cities.[4] One of these two suggestions was correct about 70-75% of the time.

To predict how this ATR assistance will change the human-UV team's effectiveness, we must determine which aspects of the human-UV team will be affected and in what way they will change. Since ATR relates to the length of human-UV interactions, it will affect II. II can be modified to reflect this change by substituting the new distribution of locate-a-city search times in place of the old search times. Since this new distribution is unknown, we must either estimate what we think it would be (or what we would like it to be), or run a small user study to model the new distribution of search times. This user study would come at a cost, though it would be significantly less expensive than a user study testing the entire team.

In order to address this need, we used locate-a-city search times observed in the user study we will describe shortly. By running Algorithm 1 on the resulting models (we predicted no change in NI, ST, or SS) and accounting for the error manifest in Figure 3, we predicted a 13% increase in number of objects collected for both 4- and 8-UV teams. These predictions, along with predictions for number of UVs lost, are shown in Figure 6.

5.3.2 Increasing UV Autonomy

In the first user study (with the original, unenhanced, human-UV team), UVs spent long periods of time waiting for user input while the user attended to the needs of other UVs in the team. Thus, the team's effectiveness could potentially be increased if the UVs' autonomy (in terms of initiative) were increased. Along these lines, we implemented a second system modification in which UVs automatically selected a new goal when they were left idle by the user. Specifically, we used a management-by-exception (MBE) level of automation in which a UV that was left idle, but not on an object in the maze, waited 15 seconds for the user to intervene. If the user did not intervene, the UV automatically moved to the nearest unassigned object (if the UV was searching for an object) or the nearest exit (if the UV was already carrying an object). Additionally, if the user did not intervene, UVs automatically chose to exit the maze via the (estimated) nearest exit in the final 45 seconds of a session.

The MBE enhancement addresses the behavior of UVs when they are neglected. Thus, we predicted that it would affect NI; we assumed that a UV that generates its own goals would behave as if the user specified the new goal. After making these changes in our model of NI (we, again, assumed the other models did not change), and accounting for the error manifest in Figure 3, we predicted 16% and 18% increases in objects collected for 4- and 8-UV teams, respectively. The complete set of predictions are shown in Figure 6.

5.3.3 Experimental Procedure: User Study 2

We used the same procedure in this second user study as we used in the first user study except that only 4- and 8-UV teams were tested. Additionally, users participated in only four test sessions (rather than six); two with four UVs and two with eight UVs. Twenty-four people between the ages of 19 and 49 years old participated in the study; the mean age was 23.8 years old. The participants were randomly divided into two groups of equal size. One group used the ATR-enhanced system and the second group used the MBE-enhanced system.

We used the same training procedure and test scenarios in the second user study as we did in the first. Thus, while the number of sessions and combinations of UV team sizes are slightly different in this second

[4]This is really automation-aided target recognition (AATR) rather than ATR. Thus, in future work we will refer to this methodology as AATR to avoid confusion with standard ATR.

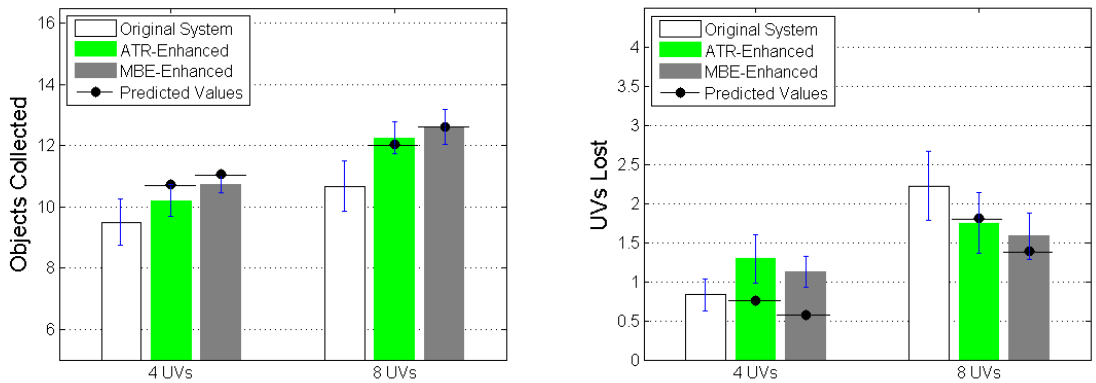


Figure 6: Comparison of observed values to predicted values for (left) number of objects collected and (right) number of UVs lost.

study, we believe that these differences had negligible effect on the results.[5]

5.3.4 Results and Discussion

Figure 6 compares the effectiveness of the original (unenhanced) human-UV team (from the 12 subjects in the first user study; we hereby refer to this as the *original* team) with the ATR-enhanced and MBE-enhanced teams. The predicted values are also shown. We discuss results for the ATR-enhanced team and the MBE-enhanced team separately. We then reflect on the implications of our findings.

ATR-Enhanced Team. The ATR assistance decreased mean locate-a-city search times by about 4.7 seconds (down from 19 seconds in the original team). As a result, the teams in the user study collected 7% and 15% more objects in the 4- and 8-UV teams, respectively (see Figure 6). 8-UV teams also lost less UVs. Surprisingly, however, the 4-UV ATR-enhanced team lost more UVs on average than the original team.

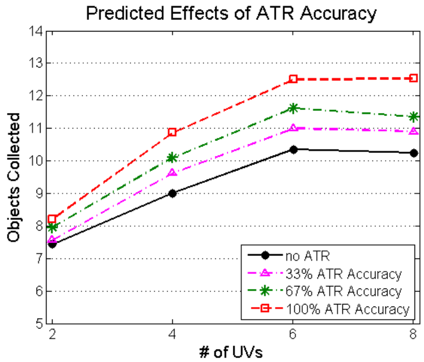
As we mentioned previously, our model predicted that the ATR-enhanced team would collect 13% more objects in both 4- and 8-UV teams. It also predicted slight decreases in the number of UVs lost in both cases. These predictions are all close to the observed results, except for the number of UVs lost in the 4-UV case. We conclude from these results that the model gave reasonably good predictions of how the ATR enhancement affected the team's effectiveness.

MBE-Enhanced Team. Figure 6 shows that the model also made reasonable predictions of how the MBE enhancement changed the team's effectiveness. Results from the user study show the MBE-enhanced team collecting 13% and 18% more objects for 4- and 8-UV teams, respectively. As with the ATR-enhanced team, the number of UVs lost increased in the 4-UV case, but decreased in the 8-UV case. Again, these results are very similar to those predicted by our model (with the exception of number of UVs lost in the 4-UV case).

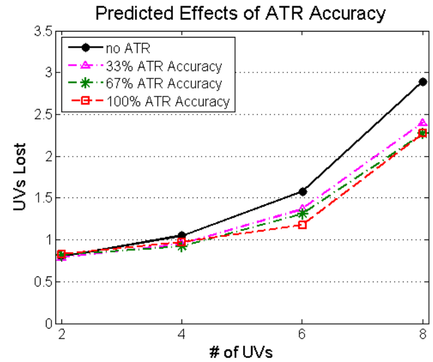
The loss of more UVs in the 4-UV case for both ATR and MBE enhancements is not statistically significant. In fact, much of the increase in UVs lost in both conditions can be attributed to a single user. Still, the fact that the trend exists for both human-UV team's suggests that this point needs further investigation. We are currently running a larger, more comprehensive, user study to further investigate this matter.

In summary, in RESCU, our model was able to predict with reasonable accuracy how the ATR and MBE enhancements changed the team's effectiveness. Additionally, in all cases, the model predicts slightly higher increases for the MBE-enhanced team than the ATR-enhanced team. This trend is also present in the

[5] We are currently running a more complete user study to test this hypothesis.



(a)



(b)

Figure 7: Predictions of the effects of ATR accuracy on (a) number of objects collected and (b) number of UVs lost.

observed data from the second user study. This result further highlights the ability of our model to predict the effects of modifications to the human-UV team.

6 What else can the model predict?

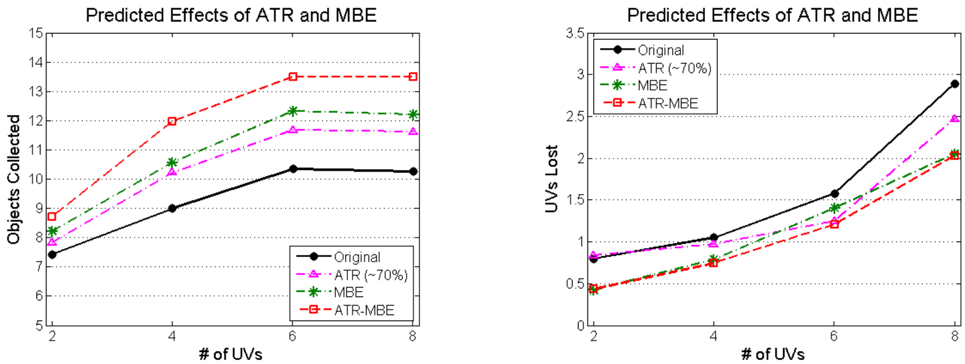
Our model accurately predicted that both the ATR and MBE enhancements would produce moderate increases in the team's effectiveness (between 12% and 18% increases in number of objects collected). If these increases in effectiveness would not be satisfactory to the end-user, we could consider other potential modifications to the human-UV team and predict how these modifications would change the team's effectiveness. These modifications could include altering ATR accuracy, combining the ATR and MBE enhancements, increasing the magnitude of individual UV effectiveness, and providing decision support for operator selection strategies. Since the behavioral model we have constructed of the team has shown to be a reasonably good predictor of how other changes would affect system effectiveness, we use the model to predict the effects of several of these system modifications. In each case, we modify the individual models that were constructed from the data from the first user study (with the original team) to estimate the effects of the design modifications. We leave evaluations of the accuracy of these predictions to future work, some of which is currently being carried out.

Specifically, we now consider the (potential) effects of altering ATR accuracy, combining the ATR and MBE enhancements, and providing decision support for operator selection strategies.

6.1 Predicted Effects of ATR Accuracy

In the previous section, we assumed a visioning system that could suggest the correct cities about 70-75% of the time. Such accuracy may or may not be possible. Indeed, in more complicated domains, attaining this accuracy would be impossible with current technologies. Additionally, it would be interesting to know whether further increases in ATR accuracy would be beneficial. As such, we use the predictive models to estimate how ATR accuracy affects system effectiveness. The results for all team sizes are given for various ATR accuracies in Figure 7.

We discuss two trends from the figure. First, as expected, the model's predictions show a nearly linear relationship between ATR accuracy and number of objects collected for each team size (Figure 7a). However, the same trend is not present in the case of number of UVs lost, as Figure 7b shows that just some ATR accuracy (e.g., 33%) decreases the number of UVs lost almost as much as full ATR accuracy. The reason for this result is not clear; we leave further analysis on this matter to future work.



(a) (b)

Figure 8: Predictions of the effects of ATR and MBE on (a) number of objects collected and (b) number of UVs lost.

Second, since the ATR enhancement essentially causes a reduction in operator workload, we would expect it to have a greater impact when the operator's workload is high. Since the number of UVs in the team affects operator workload, we would expect the ATR enhancement to bring about the highest increase in effectiveness in larger UV teams. This is, indeed, a trend manifest in Figure 7, especially in the case of number of UVs lost.

Our modifications to the model to account for varying ATR accuracies do not account for several important behavioral issues. First, the model does not consider some potential effects of varying ATR accuracy. For example, users are likely to develop higher levels of mistrust in a visioning system that suggests the correct city only a third of the time as opposed to two-thirds of the time. This mistrust can potentially lead the user to completely ignore the ATR suggestions, which would result in no increase in effectiveness. Second, the decrease in expected IT achieved by increasing ATR accuracy could change operator's selection strategies. The models used to generate the results shown in Figure 7 do not consider such changes.

Another limitation of these results is that users receive only a small time penalty for selecting the wrong city. In many applications, identifying the wrong target can have disastrous repercussions. Since decision support systems such as this ATR enhancement can lead to a higher number of such errors (due to automation bias [4]), the benefits achieved by lowering IT could be offset (and superseded) by the decrease in user performance. This information is available in the data we have collected and is an item of future work.

6.2 Predicted Effects of Combining the ATR and MBE Enhancements

Figure 10 gives the model's estimates of how combining ATR and MBE would affect the number of objects collected (Figure 8a) and the number of UVs lost (Figure 8b). The figure also shows the same predictions for a human-UV team with both ATR (with about 70% accuracy) and MBE enhancements. In the case of number of objects collected, the figure shows a nearly additive effect. However, the model estimates that the number of UVs lost would be equal to the minimum of what the ATR- and MBE-enhancements achieve separately. We are currently validating these predictions in the larger, more comprehensive user study we are currently running.

6.3 Predicted Effects of "Optimal" Selection Strategies

Up to this point, we have considered only modifying the system to improve IE and NE. We now consider how improving AAE could improve the team's effectiveness. In particular, we focus on how improving operator selection strategies (through improved interfaces and decision support systems) could possibly improve the team's effectiveness.

In this subsection, we first consider a computational methodology for computing an "optimal" selection strategy. We then discuss these selection strategies for each team size. Next, we discuss the estimated impact of using these selection strategies. Finally, we discuss the issues and dilemmas for providing decision support tools to assist users in improving their selection strategies.

6.3.1 Computational Methodology

Let Ω be the set of possible selection strategies. The effectiveness of any selection strategy $\omega \in \Omega$ can be computed by Algorithm 1 using the models of II, NI, and ST as they were calculated from the first user study for the original team. The "best" selection strategy, which we denote ω^*, could then be determined if we ran Algorithm 1 using each $\omega \in \Omega$. Unfortunately, since a complete selection strategy specifies a probability distribution function for all (S, T) pairs, the number of possible selection strategies is infinite, meaning that computing ω^* in this way is impossible.

However, we can consider a reasonable subset of Ω by considering *preferences orderings over UV states*, and then compute the best selection strategy from this subset of strategies. A preference ordering over UV states specifies an order to which the user prefers to (and does) service UVs. Analysis of user behavior from the user studies shows that this preferences ordering changes over time. For example, users are more likely to service a UV in state O in the first four minute than in the last minute.

If we assume a finite number of UV states (denoted $|S|$), the number of preference orderings over these states is finite (given by $|S|!$). Thus, if we allow preference orderings to change only at distinct points in time, the number of selection strategies we consider is also finite. Analysis of users' selection strategies in first user study show that user strategies tend to change the most in the second half (last 3 or 4 minutes) of the user study in RESCU. Thus, we consider selection strategies in which the preference orderings remain constant in the first four minutes, but (can) shift preference orderings at (1) the end of the 4th minute, (2) the end of the 6th minute, and (3) the end of the 7th minute. These changing points, while based on observations of user's behaviors in RESCU, are a point of future work.

In Section 4.3.1, we defined 13 different UV states. However, this means that we would have to consider over 1.5×10^{39} different selection strategies. Thus, we must simplify further, which we can do by alternatively considering only five different states, denoted O (for those UVs outside the maze or almost outside the maze), S (for sitting UVs in need of a goal assignment), M (for UVs moving in maze but in possible need of user attention), V (for UVs waiting on their designated objects or almost on their designated objects), and G (UVs in that the user probably should not service; i.e., they are *good* to be left alone). State O consists of the UV states **O_ST** and **XC_ST**. State S consists of the states that **XS_ST** and **RS_ST**. State M consists of the the states **RLB_ST**, **XB_ST**, **RMB_ST**, and **RSB_ST**, state V consists of the states **V_ST** and **RC_ST**. State G consists of the other three states **RG_ST**, **L_ST** and **XG_ST**.

This grouping of UVs states was done for a number of reasons. First, for the UVs to be useful, users *must* interact with UVs with states in the groupings O, V, and S. However, the reasons for interacting with a UV in each of these groups are fundamentally different. As such, they should be kept separate. UVs in grouping M (regardless of which state they are in) have alternate needs than UVs, though it is not necessarily essential that users ever interact with these UVs as they are likely to eventually reach their goals on their own. Alternately, there does not appear to be many reasons for a user to interact with a UV in state G – user interactions are not going to increase UV performance.

If we consider that UVs in group G should never receive operator attention unless there are no UVs in the other four states, we need consider only 24 preference orderings, meaning we need only consider a little over 330,000 possible UV selection strategies. In the case that multiple UVs are in the same state (in the discrete event simulation), the order that the UVs are serviced is dictated by the order in which the UV states appear in the previous paragraph. UV states listed first in the paragraph are given the highest preferences. Additionally, a UV in state M was not allowed to be serviced twice in a row.

Let $\hat{\Omega}$ be the set of selection strategies just defined, and let $\hat{\omega}^*$ be the selection strategy that yields the highest score (on average) with respect to Equation (8). Then, using the method described, $\hat{\omega}^*$ can be estimated for each team size using Algorithm 1.

Min	Pref. Order
1^{st}	$M > S > O > V$
2^{nd}	$M > S > O > V$
3^{rd}	$M > S > O > V$
4^{th}	$M > S > O > V$
5^{th}	$M > O > V > S$
6^{th}	$M > O > V > S$
7^{th}	$M > S > V > O$
8^{th}	$V > O > M > S$

(a) 2-UV Teams

Min	Pref. Order
1^{st}	$S > V > M > O$
2^{nd}	$S > V > M > O$
3^{rd}	$S > V > M > O$
4^{th}	$S > V > M > O$
5^{th}	$O > S > V > M$
6^{th}	$O > S > V > M$
7^{th}	$S > V > M > O$
8^{th}	$S > V > M > O$

(b) 4-UV Teams

Min	Pref. Order
1^{st}	$S > O > V > M$
2^{nd}	$S > O > V > M$
3^{rd}	$S > O > V > M$
4^{th}	$S > O > V > M$
5^{th}	$S > M > V > O$
6^{th}	$S > M > V > O$
7^{th}	$M > V > O > S$
8^{th}	$S > M > V > O$

(c) 6-UV Teams

Min	Pref. Order
1^{st}	$S > O > V > M$
2^{nd}	$S > O > V > M$
3^{rd}	$S > O > V > M$
4^{th}	$S > O > V > M$
5^{th}	$S > V > M > O$
6^{th}	$S > V > M > O$
7^{th}	$M > V > S > O$
8^{th}	$M > V > S > O$

(d) 8-UV Teams

Table 1: Predicted "optimal" selection strategies for each UV team size. $X > Y$ denotes that state X is preferred to state Y. State G is not shown in the table since it is always given lowest priority.

6.3.2 "Best" Selection Strategies

The selection strategies in $\hat{\Omega}$ that yield the highest average scores (with respect to Equation (8)) using Algorithm 1 for each team size are given in Table 1. Some other selection strategies yield scores nearly as high as these scores, however these preference orderings gave the highest average scores (out of 1,000 simulations). We now briefly discuss these preference orderings for each team size.

2-UV Teams. When managing 2-UV teams, Table 1a says that, in the first six minutes, users should place highest priority on UVs in state M (UVs that were not moving directly toward their goals). Thus, the model estimates that attention to local navigational control and/or task re-assignment produces the best behavior in the UV team. Furthermore, up until the final two minutes of the scenario, users should send UVs into the maze before they help users "pick up" objects.

4-UV Teams. Table 1 shows that the "best" preference orderings for 4-UV teams differ from those of 2-UV teams. Interestingly, the models recommend that users should help UVs "pick up" objects before they should send UVs into the maze (when such a decision exists). This preference ordering changes after the fourth minute, as users are then advised (in the fifth and sixth minute) to give highest priority to UVs that need to be sent into the maze (grouping O). Users are then advised to help UVs find objects, pick them up, and exit the maze in the 7^{th} and 8^{th} minutes.

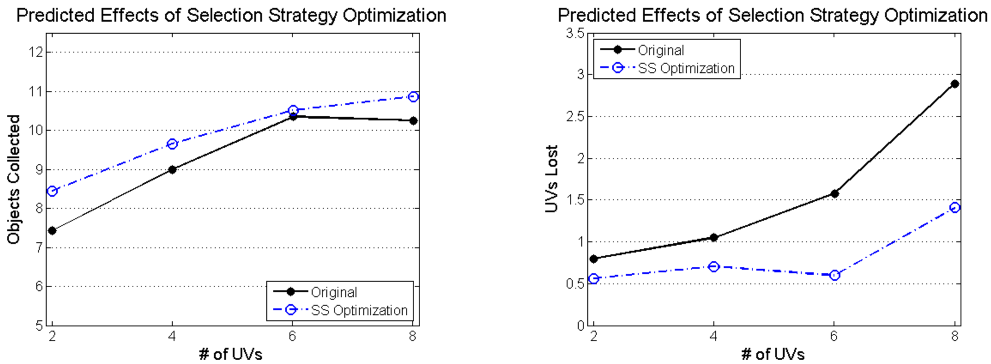
Additionally, in 4-UV teams, the model indicates that less attention (than in 2-UV teams) should be given to local navigational control and/or task re-assignment, as is indicated by the lower priority of state M in the preference orderings. Such a difference in desirable preference orderings between 2- to 4-UV teams is expected, as higher workload situations mean that a user must take a more distant role in managing UV behavior.

6-UV Teams. In 6-UV teams, Table 1c advocates that, in the first four minutes, users should give highest priority to those UVs that do not have any tasks to perform, and lowest priority to UVs that are searching for objects or exiting the maze. The table estimates that a user should place less emphasis on sending UVs back into the maze in the second half of the scenario. This strategy leads to a large drop in number of UVs lost from actual results from the first user study (we discuss this result in the next section).

8-UV Teams. Suggested selection strategies for 8-UV teams are very similar to those of 6-UV teams.

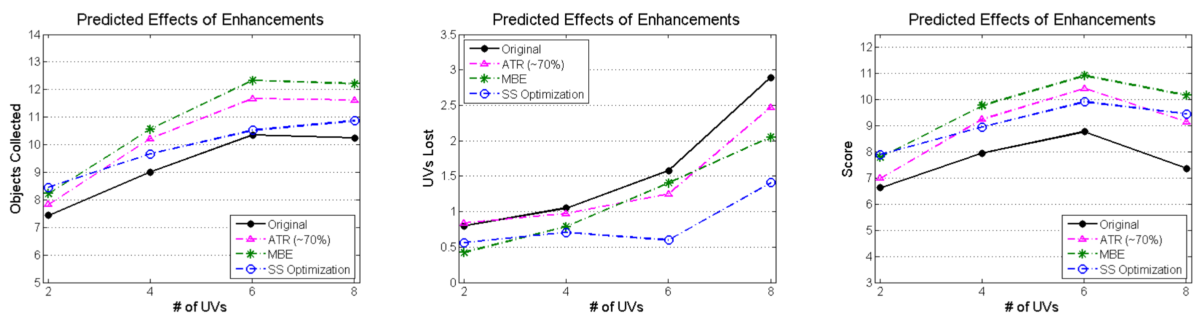
6.3.3 Impact on Team Effectiveness

Figure 9 shows estimates of the average number of objects collected and UVs lost if users were to follow the selection strategies dictated by the preference orderings given in Table 1. While improving selection



(a) (b)

Figure 9: Predictions of the effects of "optimal" UV selection strategies on (a) number of objects collected and (b) number of UVs lost.



(a) (b) (c)

Figure 10: (a) Predictions of number of objects collected, (b) predictions of number of UVs lost, and (c) predictions for team score (objects collected minus UVs lost) for various system modifications.

strategies can increase the number of objects the human-UV team collects, the figure shows that its largest benefit is in number of UVs lost, especially for large UV teams. For 6- and 8-UV teams, the model estimates that following these selection strategies would result in half as many UVs being lost as the user's actual strategies.

Comparisons of how improvements in selection strategies affect the team's effectiveness with the other design modifications we have considered are shown in Figure 10. The figure shows that the model predicts that improvements in selection strategies will not do as well at increasing number of objects collected as the ATR- or MBE-enhanced systems. However, Figure 10(b) does show that improving selection strategies is the best way (of the methods considered) to decrease system error with respect to losing UVs for larger UV teams. In fact, controlling for adequate selection strategies appears to be a good way to decrease the number of UVs lost for teams with more UVs (Figure 10(c)). For this reason, it would be desirable to provide decision support that would help users manage their selection strategies (i.e., their attention), a possibility which we now consider.

6.3.4 A Decision Support Tool for Operator Attention Allocation

There are two possible approaches we can use to improve operator attention allocation (or selection strategies). First, the human-UV interface can be improved so as to implicitly suggest where the user should direct her/his attention. For example, the interface could communicate to the user the expected amount of time it will take for a UV to exit the maze from its current location. This could help users to better plan when to remove UVs from the maze. Alternately, a more explicit kind of decision support tool could be developed that uses the "best" selection strategies found by the model (and shown in Table 1) to recommend to the user which UVs to service at any given time. It is this approach we consider in the remainder of this section.

Figure 9 gives our model's estimates of how a decision support tool for UV selection (which we call *the recommender*) could potentially improve system effectiveness. However, depending on how the user reacts to recommendations, increases in team effectiveness could be more or much less than these estimates show. Issues that appear likely to affect the user's reaction to the recommender include (a) trust issues, (b) level of automation, and (c) situation awareness. We discuss each in turn.

Trust. Because the models used to generate the "best" strategies (a) used a coarse UV state representation and (b) considered only preference orderings over UV states rather than considering the complete joint state of the UVs, adherence to recommendations made by the recommender will sometimes lead to poor performance. Previous research has shown that erroneous recommendations can lead to distrust, which has significant implications on system effectiveness (see, for example, [7]). Alternately, in time-critical situations such as RESCU, automated recommendations are known to cause automation bias, a phenomena in which users blindly follow automated recommendations [4]. Thus, to be effective, the recommender must be implemented in such a way that it induces appropriate trust levels in the user.

Level of Automation. Sheridan and Verplank outline ten ways (called levels of automation) for dividing work between the human and the automation [10]. At one extreme (the lowest level of automation), the human is left to do all the work. In the case of UV selection, this is the level of automation employed by the original human-UV team described in this paper. On the other extreme (the highest level of automation), the automation acts without user input. In the case of UV selection, this would be equivalent to the recommender determining (without human input) which UV the user should service at any given time. The model suggests that such an approach would result in the system effectiveness levels shown in Figure 9. However, since the models are imperfect, the system could ultimately perform worse. Moreover, completely automating UV selection could decrease the user's situation awareness, an issue that we discuss shortly.

Alternately, the recommender system could be implemented at intermediate levels of automation by letting the user and recommender share responsibility for selecting UVs for the user to service. Ideally, such an approach would combine the strengths of both the human and the recommender to lead to system effectiveness that is higher than either could achieve alone. However, it also has potential to decrease system effectiveness if done incorrectly [5].

Situation Awareness. Research has shown that changes in level of automation alter the user's situation awareness (see, for example, [6]). Thus, yet another issue we must consider as we develop various versions of the recommender is how human performance is affected by these changes in situation awareness. For example, lack of situation awareness caused by automated influence in the process of selecting UVs could lead to decreases in interaction efficiency, which could potentially offset the any gains contributed by such a decision support tool.

Over the next five months, we plan to develop several instantiations of the recommender. These instantiations of the recommender will employ various levels of automation for the selection of UV's for the operator to service. We will then evaluate these instantiations of the recommender via another user study.

7 Future Work

Throughout this paper, we have referred to many areas of future work. We now outline the aspects of this future work we hope to accomplish within the next five-plus months. We also outline other significant areas of future research we would like to perform in the coming years.

7.1 Next Five Months

The remainder of the research we plan to perform under the current research grant falls into four categories. We describe each briefly.

- We are currently conducting a user study involving between 60-80 participants. This user study is being run (under our supervision) by an MIT UROP. The study expands on the original user studies and will provide a cleaner set of data from which we can verify the results described in this paper. In this user study, we are also testing a human-UV team with both the ATR and MBE enhancements. We believe that this user study will provide data that will not only be helpful for the current research questions we are addressing, but that it will also help to answer other future research questions related to operator control of multiple UVs.
- As we mentioned in Section 4.3.1, we are seeking to identify sets of UV states to construct a more accurate model of the behavior of human-UV teams in RESCU. We anticipate that improved sets of UV states will give the model more descriptive and predictive power.
- In Section 6.3, we described a potential decision support tool for improving operator selection strategies. We plan to develop several possible implementations of the decision support tool, and then run a user study to test the effectiveness of the resulting systems. We anticipate that this work will be carried out in part by a visiting student from the University of Delft (under our supervision).
- While we have tried to be thorough in our research to this point, we believe that there are a number of assumptions and descriptions that can be improved. In the next five months, we hope to solidify these points more thoroughly.

7.2 Other Future Work

While the research we have performed in this project answers some interesting questions, many questions remain unanswered. We would like to continue to pursue these questions in future research. We mention of few of these research agendas.

- To this point, we have evaluated the models we have created in simulation only. To better understand their predictive power, they should be validated in real-world systems.
- We have made a number of limiting assumptions throughout the course of our research. First, we have assumed that the team contains only a single operator. It is likely that most teams of the future will have multiple operators supervising many UVs. Second, we considered only systems in which UVs performed independent tasks. While there are many situations in which UVs do perform independent tasks, many human-UV teams require UVs to work in teams. Third, we have only considered teams with homogeneous UV capabilities. Fourth, we have assumed that all tasks were of equal priority. Future work should extend the models we have created to remove these assumptions.
- Future research should include improved analysis of human interaction with automated vision systems, human-automation interaction with automated path planners, and human interaction with other kinds of machine learning algorithms.

References

[1] J. W. Crandall, M. A. Goodrich, D. R. Olsen Jr., and C. W. Nielsen. Validating human-robot systems in multi-tasking environments. *IEEE Transactions on Systems, Man, and Cybernetics – Part A: Systems and Humans*, 35(4):438–449, 2005.

[2] Jacob W. Crandall and M. L. Cummings. Developing performance metrics for the supervisory control of multiple robots. In *Proceeding of the 2nd Annual Conference on Human-robot Interaction*, pages 33–40, 2007.

[3] Jacob W. Crandall and M. L. Cummings. Identifying predictive metrics for supervisory control of multiple robots. *IEEE Transactions on Robotics*, 2007. Accepted.

[4] M. L. Cummings. Automation bias in intelligent time critical decision support systems. In *AIAA 1st Intelligent Systems Technical Conference*, pages 33–40, September 2004.

[5] M. L. Cummings, A. S. Brzezinski, and John D. Lee. The impact of intelligent aiding for multiple unmanned aerial vehicle schedule management. *IEEE Intelligent Systems: Special Issue on Interacting with Autonomy*, Mar/Apr 2007.

[6] David B. Kaber and Mica R. Endsley. The effects of level of automation and adaptive automation on human performance, situation awareness and workload in a dynamic control task. *Theoretical Issues in Ergonomics Science*, 5(2):113–153, 2004.

[7] John D. Lee and Neville Moray. Trust, self-confidence, and operators' adaptation to automation. *International Journal of Human-Computer Studies*, 40(1):153–184, January 1994.

[8] D. R. Olsen and M. A. Goodrich. Metrics for evaluating human-robot interactions. In *NIST's Performance Metrics for Intelligent Systems Workshop*, Gaithersburg, MA, 2003.

[9] T. B. Sheridan. *Telerobotics, automation, and Human Supervisory Control*. The MIT Press, 1992.

[10] T. B. Sheridan and W. L. Verplank. Human and computer control of undersea teleoperators. Technical report, Man-Machine Laboratory, Massachusetts Institute of Technology, Cambridge, MA, 1978.