On Human-Assisted Multiagent-Based Planning

Dr. Leah Kelley and Dr. Michael Ouimet
SPAWAR Systems Center Pacific
53560 Hull Street, San Diego, CA 92152-5001
USA

kelleylc@spawar.navy.mil ouimet@spawar.navy.mil

ABSTRACT

One of the goals of the U.S. Third Offset Strategy is to develop a truly collaborative human-machine fighting force. Currently, the state of the art in the field is a single soldier or sailor providing direct supervisory control of a machine agent, requiring the soldier's full attention. A truly collaborative force requires capable machine agents able to generate and execute courses of action without constant, direct human supervision. At the same time, the human teammates must still be able to affect the machine agents' decision-making, and vice versa. This paper addresses the topic of human-machine interaction, communication and cooperation applied to Artificial Intelligence planning.

Automated machine-based agents interact with the world in two ways: they react to external stimuli or deliberately execute previously generated courses of action. Reactive behaviours, such as a ground vehicle swerving to avoid obstacles, happen in extremely short time frames and address short term needs. Not only is there little time for human intervention during execution, there is also little need, since the decision making complexity for execution of short term, immediate goals (e.g. avoid the obstacle in the path) is low. In contrast, generation of deliberative courses of action is more complex, and requires goal-based decision making: given a causal model of the environment, a set of objectives, and a belief about its current state, a machine-based agent can formulate a course of action – a plan. This form of goal-based decision-making is called "Planning" by the Artificial Intelligence (AI) community.

The AI Planning community has developed planning algorithms over the past several decades. Many of these algorithms work well when applied to well-constrained planning problems, where assumptions of turn taking and a static world hold. Even games as complex as Heads-Up No-Limit Texas Hold 'Em poker are well-constrained when compared with the dynamic, uncertain warfighting environment. When the current planning algorithms are applied to real-world, military-relevant problems, the domain space the planning agent must reason over explodes such that its planner can't find a solution at all, or by the time it finds a solution, the time to execute it has passed.

In this research, we hypothesize that human knowledge can be injected into the AI planning process to help pare down the search space and assist the machine agents' planning by providing additional intelligence in the form of constraints and goals dynamically during plan generation and execution. We start with a simple representation of a naval-relevant planning problem and systematically increase complexity by removing underlying assumptions that make the problem tractable. As we remove assumptions, we find ways to inject human intelligence into the planning process, enabling the machine agent to reason over more complex scenarios. We apply this method to multi-agent mobile target search scenarios in simulation, and have developed ways for a human user to interact with several AI agents' planning without requiring the user to understand the complex mathematics behind the planning algorithms. This paper addresses our methodology, development of a human-user interface, and development of an experimental framework for measuring the effectiveness of human intelligence injection into AI planning through human subject trials.

1.0 INTRODUCTION

Recently, the U.S. Military has expressed a desire for a fully collaborative human-machine fighting force, and also seeks to find ways to enable a single or small group of soldiers or sailors to oversee many machine agents, e.g. in a robotic swarm [1]. Currently, at best a single soldier or sailor oversees a single machine agent, and in many cases a machine agent requires several humans for oversight and mission support. In this paradigm, the human operator is required to make very low-level decisions for the agent, such as supplying a fine waypoint trajectory for its motion, or even tele-operating it. The human also has to closely monitor the machine's behaviour, and may be required to take emergency action on the machine's behalf. Such tight supervision is not practical when overseeing more than a few agents [2].

Several technical advances are needed in order to transition from today's single robot-single human operator paradigm to a truly collaborative multi-machine, multi-human force. These include, but are not limited to, advances in machine sensing, perception and reasoning, and advances in command and control interfaces for facilitating human interaction with and supervision of multiple agents. This research focuses on enhancing machine reasoning and decision-making by incorporating human decision-making into machine goal-based reasoning, as well as on effective means of communicating machine agents' decision-making and reasoning to human operators for effective teaming.

When a machine agent interacts with the world, it can respond to external stimuli, as a mobile robot avoids obstacles or as a digital assistant searches for, and then answers, a query. However, if a machine agent is to achieve a longer-term goal, such as reach a destination across a room, efficiently perform a series of tasks, or play a game such as Chess, it has to generate a course of action to achieve its goal. This goal-based reasoning process is called "Planning" by the Artificial Intelligence (AI) community. If humans are to effectively team with machine agents, and are to effectively supervise many machine agents, they need to be able to influence the machines' planning processes and vice versa.

AI Planning is an active area of research and has been over the past several decades [3, 4]. Much of the work has been focused on domain-independent planning algorithms that are capable of solving wide varieties of planning problems in different application domains. To use such planning algorithms, one must provide the planner with a model of the application domain, a planning problem definition, a goal to achieve and an initial starting condition. This requires that the user not only understand the application domain and the problem, but must also have understanding of how to formulate them so that the solver can interpret and solve the problem. Additionally, many of the existing planners require that the domain be modelled using discrete representations of variables, that the assumption that the domain remains static during planning holds, that the agent knows all relevant facts about the domain and that the agent's sensing is perfect. Though the AI planning community continues to improve planning algorithms and has addressed some of the above limitations, they still cannot handle all real-world complexities, especially as found in highly dynamic, uncertain domains relevant to successful warfighting.

1.1 Research Approach

This research develops a methodology to mature AI planning algorithms to make them applicable to realistic, military-relevant scenarios. Given a simplified version of a military-relevant problem, we apply an appropriate AI planning algorithm to it, and see how it performs. If the algorithm is able to generate a plan for the simplified problem, we remove one of the simplifying assumptions and re-run the planner. If the planner is still successful, we remove another simplifying assumption. We continue removing assumptions until the planner is unable to generate a solution. At this point, we then find ways to inject human intelligence/help into the planning so that it is able to generate solutions. We then remove another assumption, find where the planner is unsuccessful again, and find additional ways to inject human intelligence and help into the planning process. Using this systematic approach, we enable AI planners to address more realistic planning problems (see Figure 1).

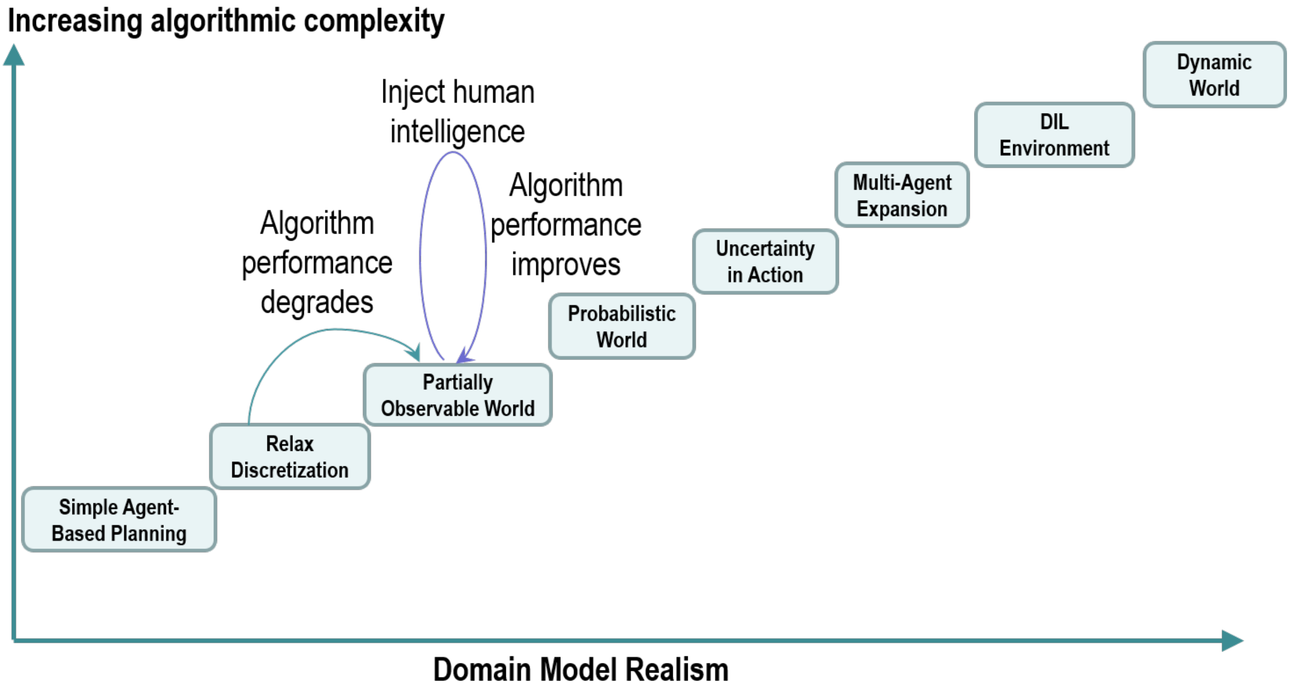

Figure 1: Assumption removal and human intelligence injection process

This research also addresses methods for communicating the AI agent's goals, beliefs and needs to a human supervisor so that s/he can assist the planning agent. Unlike previous work in semi-autonomous systems, which focuses on methods to assist a robot during task execution [5], this work has similarity with mixed-initiative planning [6-8], where a human and AI agent interact in order to jointly form a plan before task execution. The mixed-initiative planning work cited focuses on tight interactions between the machine and the human, where the machine agent poses a series of questions to a human and explicitly asks for human input, and hence the need to create complex domain models is circumvented by having the human present in the planning as an active participant. However, as with tight supervisory robot control fielded today, a tight-knit and highly interactive planning approach doesn't scale to situations where one human must supervise many machine AI agents, as in swarming applications. This research seeks to find ways to facilitate human assistance to the AI agents without requiring the agents to explicitly request help, and also without requiring tight supervision of each agent from a human operator, so decision-making is less tightly coupled than in mixed initiative planning. The goal is to find ways to couple the human abilities of quick decision-making under highly dynamic situations with the computational strengths of AI planning agents so that together humans and machines can accomplish complex tasks neither can do (either well, quickly or at all) without help from the other. To facilitate such cooperation and interaction, the agents' goals, beliefs and needs have to be displayed to the human in an easily interpretable manner, and the human needs to be able to provide information in a form the machine agents can understand and use.

The rest of this paper is organized as follows. Section 2 describes the current state of the art of AI Planning, as well as our prior work on injecting human intelligence into the planning process of a single AI agent in a naval-relevant scenario. Section 3 introduces our notion of hierarchical planning and describes the extension of our single human-single agent approach to a single human-multi-AI agent persistent search scenario. Section 4 describes our approach to and development of a human subject test and experimental framework.

As of the time of writing, the team is still awaiting permission to officially begin the human subjects test. Section 5 concludes the paper and discusses expected future outcomes and directions.

2.0 BACKGROUND AND PRIOR WORK

2.1 State of the Art of AI Planning

The AI community has developed several types of planning algorithms and techniques over the last several decades, with primary emphasis on developing domain-independent planners that can solve problems in a wide variety of application domains. Early methods, often referred to as "Classical Planning," use situational calculus to find a set of transitions from an initial state to a goal state given a domain model that encodes all possible state transitions using some form of formal logic, such as prepositional or first order logic [3]. These methods assume the world is static, time is atomic, only the agent causes changes to the state of the world, the agent's actions are deterministic, the agent has complete domain knowledge, the domain is not affected by external influences (i.e. if the initial conditions are completely known, then the action sequences generated will be completely and correctly predicted), and the planning goals are known. Such assumptions are valid even for complex games such as Chess and Go, but do not hold for many real-world problems.

State-space search methods have been used to solve classical planning problems. Such methods include tree-based and graph-based search, where states an agent and the environment are junctions or nodes, and transitions between states are "branches" or "edges." Search algorithms can start from either the initial state or goal state to build plans (forward search and backward search, respectively), and can either explore down one path (depth first) or explore all transitions from the starting node (breadth-first). The planning community has developed heuristics for choosing between alternatives, in order to speed up plan generation. A* search is one well-known best-first search method [3]. Other heuristic search methods include hill-climbing, simulated annealing, genetic algorithms and belief-state search for partially observable environments. Though early heuristic functions were encoded by hand, methods exist to generate heuristics for particular problems automatically by analysing the domains [4].

Planning problems can be formulated as constraint satisfaction problems (CSPs). In constraint satisfaction, a solver seeks to find an assignment of permissible values to a set of variables such that the assignment satisfies all of the constraints that dictate how the variables interact and/or combine. CSPs can be defined using prepositional logic, Boolean formulas or mathematical formulas. Search methods used for classical planning can be applied to constraint satisfaction problems. Additional methods for solving CSPs include constraint propagation, backtracking, back-jumping and model checking [3]. Though methods such as model checking can solve large CSPs, such problems still require assumptions of determinism (no uncertainty) and static domains with no external influences.

Domain-specific planning algorithms highly tailored to particular applications and problems also exist. For example, Google's Optimization Tools contain specialized algorithms that solve combinatorial optimization problems such as Traveling Salesman Problems, knapsack problems, linear assignment problems, bin packing problems, and vehicle routing problems, as well as more general constraint and linear programming solvers [9]. In order to use such solvers, one must cast one's planning problem into the particular form required by the planning algorithm, and the same assumptions used when encoding the domain must apply to any new planning problems one tries to solve.

The AI community has also developed planning methods that are able to cope with more complex scenarios. Hierarchical Task Networks (HTNs) divide complex planning problems into smaller and smaller sub-problems until sub-problems are fully decomposed into sets of primitive actions. Sub-problems, called higher level actions (HLAs), may be realizable by various combinations of primitive actions. When solving HTNs, a planner may reason over the higher-level actions, letting an agent determine the primitives to use during solution execution [3]. The Simple Hierarchical Ordered Planner (SHOP) and SHOP2 HTN planners generate plans with steps in their execution order, reducing complexity [10], and have been extended to handle nonlinear and continuous effects [11] as well as user preferences [12]. Case-based planning methods use databases of previously generated plans to form solutions to new planning problems. This method is applicable to planning problems that exist within one application domain, and where new planning problems are assumed to be similar to previously solved problems. Reuse of previous solutions is intended to reduce computational complexity [13]. However, the assumption that new problems are similar to old problems is not necessarily valid.

Planning methods that handle uncertainty have been developed, and include contingency planning, Markov Decision Process (MDPs) and Partially Observable Markov Decision Process (POMDPs) iteration, and inference over statistically modelled processes. In contingency planning, the domain is modelled as an "And-Or" graph and online planners generate solutions using depth-based search, random walk, hill-climbing or Learning Real-Time A* (LRTA*) [3]. Fully observable, stochastic environments can be represented as Markov Decision Processes (MDPs), assuming uncertainties in the environment, sensory data and effects of agent actions can be quantified [3, 14-15], and optimal solutions to problems in such representations can be found using dynamic programming. POMDPs are extensions of MDPs to domains in which an agent cannot fully observe the environment, hence its current state is also represented as a probability distribution. Methods for handling POMDP planning include back-chaining [14], an extension of value iteration [15], and branch-and-bound and gradient ascent methods [16]. Inference reasoning algorithms have been used to solve partially observable stochastic processes modelled as Bayesian Non-Parametric Models (BNPs) [17]. Finally, methods for planning for imperfect information games have been developed, and have been successful at defeating all human opponents of Heads-Up No-Limit Texas Hold 'Em poker [18], using a combination of planning over abstraction (blueprint strategy computation), real-time sub-game solving, and self-improvement game play. Note that even in imperfect information games, the assumption of a static world while planning still holds.

Model-free Reinforcement Learning algorithms have had recent success in playing videogames, starting with DQN on a variety of Atari games [19]. This method directly learns from experience and does not have any explicit planning or model of the games. DQN has best performed on games that were more tactical than strategic though where there is a rich reward signal or reactions are key. However, recent work by OpenAI on Dota 5v5 has shown the algorithm PPO can play a complex, team-based adversarial game in real-time with partial observations [20, 21]. Even though no planning is being done, the agent is able to play competitively at a game where long-term strategy, teamwork, and tactics are all important by learning from previous data and only using the most recent states as the input to the action selectors. This approach is promising since it doesn't require explicit domain modelling in advance. However, it becomes difficult for a human to collaboratively work with an agent when s/he doesn't understand the agent's decision-making process.

As problem complexity increases, more complex domain models with large state space representation are required, regardless of the methods one intends to apply to solve the planning problem. When considering large state spaces, planning algorithms require significant time to perform computations or, in some cases, cannot generate plans at all. Plan generation is PSPACE-hard even when the set of reachable states is finite – to generate the shortest set of transition actions, the number of states that must be considered from initial to goal state is exponential in in the size of the planning problem description [4]. Planning computational complexity can range from constant time to NEXPTIME-complete, with increasing complexity corresponding to fewer simplifying assumptions [22]. This research addresses the challenge of applying the state of the art planning techniques, especially those that handle some degree of uncertainty, to naval-relevant problems that are not well constrained.

2.2 Single Human – Single Agent Cooperative Planning

As a starting point for this research, we selected an underwater object search task as a representative naval-relevant scenario, and applied the systematic relaxation approach described above to it. In its most simplified version, the underwater search task is a recasting of the "Wumpus World" game [3], where the Wumpus is replaced by an object of interest and pits are replaced by underwater obstacles. Note that unlike the Wumpus game, there is no treasure and here the AI agent simply wants to find the object and interact with it. The simplest version takes place on a square grid, and the agent knows where it starts, all locations of obstacles and the object of interest and has perfect sensing and motion. Under these circumstances, planning is easily accomplished using A* search or other tree or graph-based search methods, and no human assistance is necessary. However, if this search were to be expanded into 3-dimensional space and the agent's roll, pitch and yaw were also considered, the number of states would become too large for tree-based methods to find optimal paths in reasonable time and one would need to employ heuristics to get real-time but sub-optimal results.

We were able to remove several simplifying assumptions to this scenario without needing human assistance, simply by changing planning algorithms and domain representations. By casting the search problem (still in 2 dimensional space) as an MDP, the problem becomes a simultaneous localization and mapping (SLAM) problem in addition to an object search. The SLAM problem has been well studied in the robotics and AI communities [23]. In SLAM, the agent does not have to have perfect sensing, motion, or knowledge of obstacle locations, and can still successfully navigate and localize, while building a representation of the environment. This representation is more realistic since sensors are prone to false readings, unmodelled currents can affect agent motion and it is unlikely that all underwater object locations are known a priori. This representation still assumes the search space is discretized into a grid.

In the MDP representation used here, the agent uses a reward function distributed over the entire discretized space to decide where to move next, by moving toward locations where it will receive high reward and avoiding locations where rewards are low. Its overall goal is to find a policy that will maximize its reward over time. When no information is provided about the space in advance, the agent is incentivised to search randomly. As the agent explores and senses obstacles, the value of the reward in previously explored regions decreases, and, at sensed obstacle locations, the reward drops significantly. The reward function thus evolves as the agent navigates. Further details on the implementation on the MDP iteration planning applied in this research are available [24].

This formulation provides a human with opportunities to influence the agent's planning by interacting with the reward function. A human operator can set initial goals and keep out areas, high reward and low reward areas, respectively, and can make changes to goal locations and keep-out locations on the fly by changing values of the reward function while the agent navigates. Rather than requiring a human user, especially a non-expert user, to interact with the mathematical values directly, the team developed a graphical user interface to facilitate such interactions, using the Intelligent Multi-UxV Planner for Adaptive Collaborative/Control Technologies (IMPACT) System [25, 26] as a simulation and visualization tool and user interface. The agent's search area, current location, goals and detected obstacles are displayed onscreen in one of two ways: as a greyscale occupancy map (see Figure 2, left) or as a reward function map (see Figure 2, right). The greyscale representation shows the agent's position (agent icon), user-defined agent goals (red circles), detected obstacles (black areas), previously traversed areas free of obstacles (white areas) and unexplored areas (grey areas). The reward function is depicted as a heat map overlaid on the search area, where rewards range from red to blue through a rainbow colour scheme, with red depicting high rewards and blue depicting low rewards. In the heat map representation, the user-defined goals are shown in red, detected obstacles are shown in blue, and other areas primarily green, have intermediate reward values. The reward function is structured such that its value gradually decreases away from a user-defined goal location in all directions.

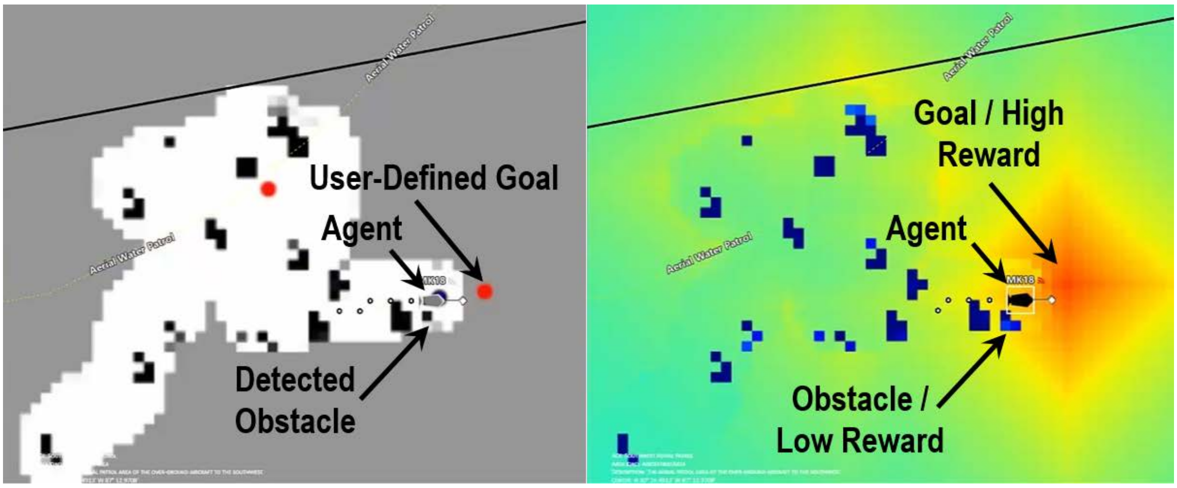

Figure 2: Occupancy map, left and reward function map, right [24], as displayed in IMPACT.

The user is able to interact with the agent's reward function using a custom task incorporated into IMPACT's Task Manager (see Figure 3). The user can toggle between the heat map (reward function map) and the greyscale map using the drop-down menu button. To change the intensity of the reward function, the user clicks on the appropriate button (Increase Intensity or Decrease Intensity) and then clicks on the search area. By changing the reward intensity graphically, the human can influence the agent's behaviour and navigation without understanding the mathematics behind the reward function or the agent's model of its motion, and can do so as the agent executes its motion.

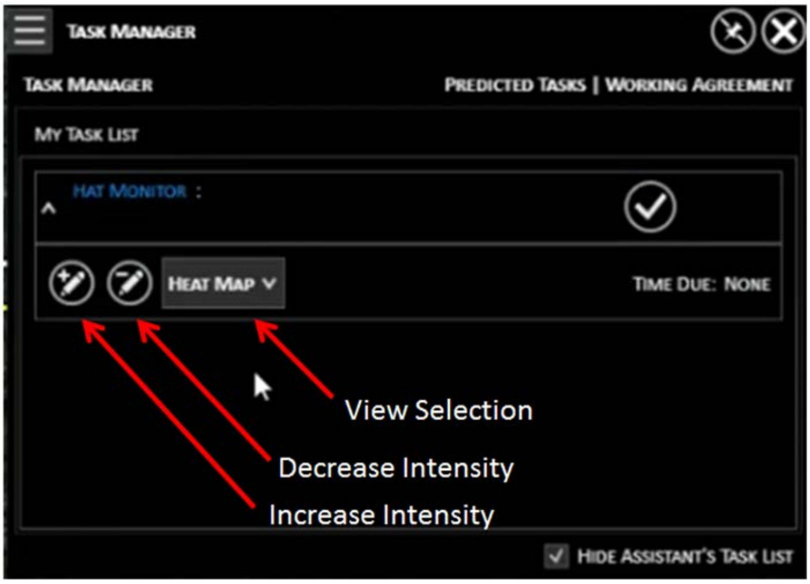

Figure 3: Task Manager controls for human operator, from [24].

3.0 HIERARCHICAL PLANNING WITH MULTIPLE AGENTS

In the single-agent mapping scenario, MDP iteration is used to fuse obstacle data and determine where the agent should go next. As the planning horizon, number and complexity of tasks, and number of agents increase, it becomes impossible to plan the whole mission at the lowest level of granularity (e.g. direct motor control actions). Hierarchical planning allows one to solve a complex task by composing the solution from a nested hierarchy of increasingly abstracted, simplified problems. This approach can be used to guide a team of unmanned systems in mapping an area, persistent monitoring, and tracking moving targets. A concrete example is that a group of agents can monitor an area by first deciding how to split the area, potentially based on agents' different skills or battery charge (algorithm 1) and then letting each agent decide for itself the path to best monitor its assigned area (algorithm 2). At the lowest level, another planner governs the physical actuations required to maintain on course in the presence of unmodelled disturbances or unknown obstacles (algorithm 3).

There are at least two reasons for adopting a hierarchical approach: one to benefit the human and the other to benefit the autonomy. First, people have a limited attention bandwidth available and a person cannot directly control more than one vehicle at a time. Thus, in order for human-autonomy teaming with more than one robot, at least some level of hierarchy is required. Furthermore, it can be challenging for a person to even maintain situational awareness of 8-12 agents each doing their own tasks autonomously [2], let alone to aid by also directly tasking them or providing input on their paths. This leads us to conclude that people are most suited to aiding a team of robots by providing input at the higher levels of hierarchical abstraction.

As one introduces multiple agents into a strategic planning domain, the algorithmic complexity increase dramatically. The challenge comes largely from the coupling between each agents' actions. Because each agent affects the rest of the team as well as the global environment, care must be taken to ensure that all team members coordinate so they do not interfere with one another, not perform the same task twice. However, planning over a long time horizon for all agents at the lowest level of abstraction in a centralized way becomes computationally infeasible due to the combinatorial number of possible joint trajectories that must be evaluated. For this reason, we elect to solve a high level coordination/assignment problem to decide which agent(s) should do which task(s). Once this decision is made, the agents can each perform their own path planning and reactive planning to safely achieve their assigned goals. As tasks are finished or new ones arise, replanning can perform multi-agent assignment again, passing the updated goals to the lower level algorithms. A schematic of our model can be seen in Figure 4.

In addition to depicting the type of planning algorithms appropriate for use at the various levels of the hierarchy, Figure 4 depicts the levels at which various types of human intelligence are most appropriately inserted into the planning process. At the highest level, a human commander will define the overall mission goals, such as priority search locations in a multi-agent search scenario. The commander may also have preferences for which agents perform which tasks, or may specify that agents with particular abilities are suited to particular tasks. This information can be injected into task assignment planning. In addition to task assignment preferences, the human commander may also have additional information about activities or features in the search environment, such as cluttered/unsafe areas or other keep-out zones. Environmental information will be used by each agent as it plans its navigation routes, and such information may change over time, so the human commander requires methods to communicate this information to the appropriate agents. As stated earlier, it is difficult for one human to closely monitor more than 8-12 agents, so aside from occasional tele-operation (assuming the mobile agents have enough autonomous capability to handle their own well-being in most situations), human injection at the lowest levels of planning is not needed.

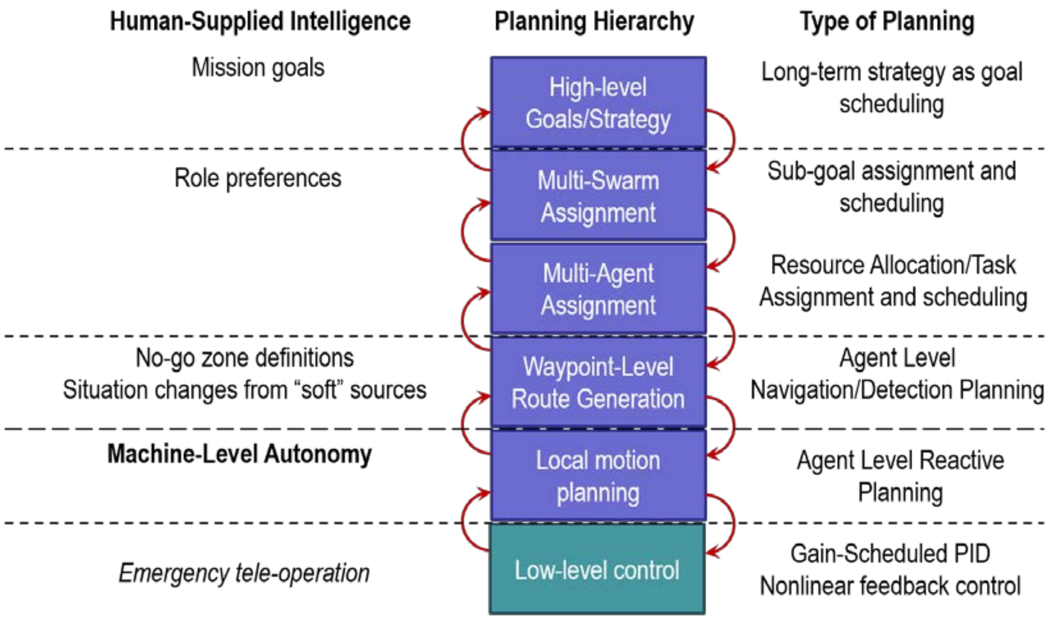

Figure 4: Hierarchical planning model, including areas for injection of human intelligence

3.1 Example: Multi-agent Persistent Monitoring

As an example of a multi-agent hierarchical planning problem, consider a 2-dimensional environment that requires persistent monitoring with multiple mobile machine agents, where the importance of maintaining watch over some locations is of greater importance than others. Following the hierarchy in Figure 4, a human commander defines locations that need to be observed, as well as assigning the level of importance to each location. The hierarchical planning process then starts, omitting the "Multi-Swarm Assignment" level in this case.

Voronoi Partitioning: Since some parts of the environment are more important than others, the highest level of the hierarchy partitions the environment into N continuous regions, one for each agent. The Voronoi partitioning algorithm [27] weights important regions more heavily so some agents may cover large swaths of unimportant areas, whereas other agents cover smaller, more important regions more frequently.

Linear Assignment: Once the optimal partitioning is determined, we use the Hungarian algorithm solution to the Linear Assignment problem [28] to match each agent with one region based on the distance to the areas' centroids. The global matching ensures that sum of distances between agents and their assigned regions is minimized.

Markov Decision Process: Each agent is then provided their region as well as the relative importance of the subareas in that region. The MDP algorithm uses the importance values as the reward for each potential location. When a location is visited, its importance is set to zero. Globally, the importance increases back over time at a rate dependent on how important a person deems that area.

4.0 HUMAN SUBJECT EXPERIMENTAL FRAMEWORK

The prior work described above established methods for non-expert users to interact with individual AI agents performing navigation and search tasks, and the team has introduced and implemented a hierarchical planning structure that can handle multi-agent persistent monitoring as well as multi-agent object search. However, to date the research team has been unable to quantify performance improvements and also has yet to determine how well such methods scale to single-human multi-AI-agent scenarios. Additionally, the team has not gathered any data on the effectiveness of the heat map as a method for displaying an AI agent's goals and belief states to non-expert users. To gather such data, the team has developed a human subject experimental framework, test scenario and protocol, currently under review by the U.S. Navy's Human Protection Review Office.

The human subject study uses a heterogeneous multi-agent search scenario to determine whether the human input enables reduction of computational complexity and time required by the planning agents to generate successful assignments and motion paths that accomplish the search goal. A human subject plays the role of a human commander of a small fleet of 7 autonomous, unmanned surface vehicles (USVs) and 5 unmanned air vehicles (UAVs) in a fictitious, simulated coastal area of the world. The human-UxV fleet has been tasked to find an adversary who is somewhere in the region. Each USV has its own machine artificial intelligence (AI) agent capable of planning and executing its motions, using the MDP iteration method described above. The UAVs are carried by a "carrier ship" that serves as a mobile charging base, which the human participant can instruct to move to different locations. The participant can also instruct the carrier ship to deploy all of the UAVs, which then use their own MDP agents to navigate and perform search tasks while remaining close enough to the carrier ship to recharge their batteries as needed. The UAV deployment is accomplished through Voronoi partitioning. Finally, the participant has an AI assistant that can help assign the USVs to search areas in the region using a combination of Voronoi partitioning and the Hungarian method for linear assignment. Note that the human participant can decide to assign a subset of agents to specific search areas and can have the AI assistant take care of assignments for the remaining agents.

The experiment front-end is a web-based user interface that displays a map of the fictitious region where the experimental scenario occurs, icons of the autonomous agents the user can interact with on the map, as well as various GUI interfaces the user can click on (see Figure 5). This user interface is agnostic to the type of web browser a participant uses, hence one can perform the experiment from any computer on the network where the host server resides. A dedicated computer workstation hosts the front end, and also runs the experiment back-end simulations and AI planning algorithms, and serves as a data recording device.

The web-based user interface (Figure 5) contains a map of the fictitious littoral region in the upper right, a table of simulated unmanned vehicle agents on the left that includes information about each agent's state (e.g. fuel levels, current tasks assigned) and a timeline of events at the bottom. During the experiment, participants receive information about what is happening in the scenario in several ways. Agent locations are displayed on a map in the web interface as they move about the region. Their status, task assignments and power levels are displayed in a table, which dynamically updates as the agents perform tasks. A participant can click on an agent in the map, which then displays a heat-map for that agent overlaid on the main map that displays the agent's reward function over its assigned search region, similar to the heat map shown on the right side of Figure 3. As in the earlier IMPACT-based single-human-single-agent simulation, the human user can add or change the priorities of previously defined search locations by clicking on the appropriate increase or decrease buttons and then clicking on the map. A participant can also assign search areas to specific agents if they so desire, by dragging unassigned tasks to agents (Figure 5, left side), and by creating new tasks and then assigning them to agents. Such methods of interaction are intended to be intuitive, and easy to use with minimal instruction. As part of the pre-experiment brief, participants are provided with a 5-minute tutorial that describes the features of the user interface and provides a walk-through of the interactive functions.

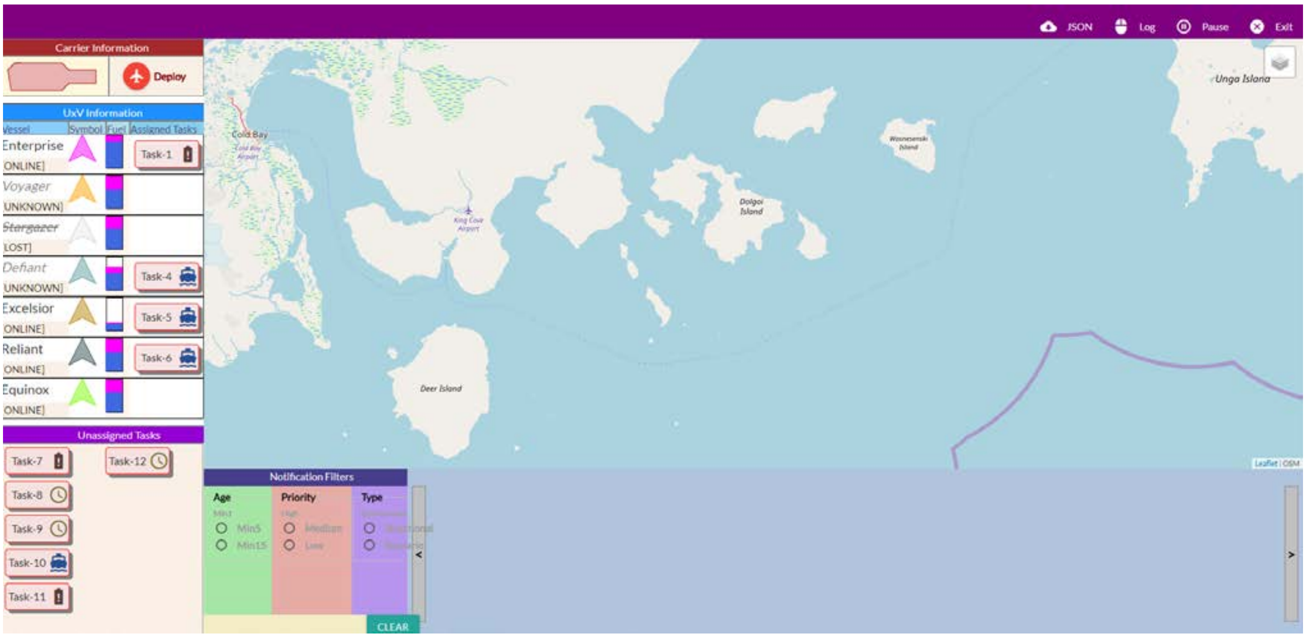

Figure 5: Human subject study web-based user interface

During the experiment, participants will also receive situationally relevant information in the event timeline at the bottom of the web-based interface (see Figure 5). Such information includes camera images from USVs and UAVs as they search the region, as well as messages about events happening in the region that the machine agents cannot understand. In this experiment, the UxVs are assumed to be able to detect objects, but cannot perform the high fidelity classification for distinguishing the adversary from other entities. Participants can adjust search goals and reassign tasks based on this information to help the UxVs find the adversary. Finally, participants can also deploy the UAV fleet to obtain higher-quality images (that will appear in the event timeline at the bottom of Figure 5) if they so choose, by clicking on the "Deploy" button shown in the upper left in Figure 5 under "Carrier Information."

The human subject study records participant performance by tracking performance metrics based on the inputs the participant provides in the web interface, as well as by tracking metrics in the back-end simulation, such as completion time and computational load. Actions taken by a participant to direct the agents (e.g. clicking on the map or other user interface elements in the GUI, change of view, examination of the screen, etc.) will be directly recorded, as will the AI planning agents' computations and actions. Immediately following an interaction with an agent (or agents), the experiment will pause and display a brief "pop-up" survey (multiple choice and short text-based answering) that asks the participant about the action just performed, and the reason the participant had for choosing to interact at the time. Such questions are designed to elicit participants' beliefs about the AI agents' behaviours, as well as to determine whether supplemental information was used to redirect/assist the AI agents. Additionally, pre-and post-experiment surveys are used to collect non-personal identifiable demographic information, such as whether or not a participant plays video games, and perceptions/understanding of the experiment, respectively. Post-experiment questioning will also be used to gauge the effectiveness of the tutorial provided to participants before beginning the experiment. The post-experiment survey also includes the NASA Task Load Index [29, 318], to be used as a supplemental measure of workload as well as to facilitate general comparison to known ranges of workload in other fields, to prior studies, and to other participant groups such as experts compared to novices.

Performance measures include the time to complete mission, the level of computational burden exhibited by the AI agents (e.g. time it takes an AI agent to generate a plan) and instances of agent stagnation during target search. The number of algorithmic course changes detected versus no effect in algorithmic outcome will be measured through accuracy to ground-truth goal representation. Correlation of course changes with instances of human participant interaction, along with participant answers to pop-up survey questions, will be used to determine whether human intervention helped, hindered, or had no effect on the agent's search task. Analysis of a participant's answers on the questionnaire regarding their awareness of the experiment's purpose and understanding of the autonomous planning agents' actions will use the established methods of Endsley and Garland [31].

The following hypotheses about the effects human input have on the performance of AI planning algorithms will be tested using results of this experiment:

- The human-AI teams will successfully interdict the adversary more frequently than the AI alone.
- The human-AI teams will complete the interdiction mission more quickly than the AI alone.
- Those subjects with prior experience playing video games are expected to complete the interdiction task more frequently than those without prior gaming experience.
- Those subjects with prior experience playing video games are expected to complete the interdiction task more quickly than those without prior gaming experience.

Human subject tests are anticipated to start immediately upon protocol approval.

5.0 SUMMARY

In this paper we've described our approach to collaborative human-AI agent planning that can enable plan generation in highly dynamic, uncertain environments with few constraints, by injecting human intelligence into the AI agent's planning process. Our prior work focused on single-human-single-agent collaboration, and we previously demonstrated a method and developed a user interface that permits a non-expert user an intuitive means of interacting with the single agent both before and during plan execution. In this paper, we've presented our extension of the single-human-single agent collaborative planning for search tasks to single-human-multi-agent search and persistent monitoring scenarios. By constructing a hierarchical planning process using several AI agents, we decompose a highly complex scenario into smaller planning problems, where the human supplies situationally relevant information at the higher planning levels, while the individual agents take care of their own lower-level execution plans. By interpreting situationally relevant information that machine agents are incapable of understanding, the human collaborator can help guide the AI planners towards plan generation, speeding up the planning process. Human experience also serves to fill gaps in the AI agents' domain models.

We've developed methods for human interaction with AI planning agents that do not require the user to have expertise in the field of AI planning, or extensive training in order to engage and collaborate with the AI agents. Use of intuitive representations, such as color-coded occupancy and heat maps, pictorial bar charts, and dynamic displays, are intended to make the AI agents' goals, intentions and beliefs about their world states easily interpretable by non-expert users. Use of simple interaction mechanisms, such as adding or subtracting value from map locations by pointing and clicking, as well as dragging tasks into tables, are expected to be easy for non-experts to use to interact with the AI agents. We anticipate human test results will provide quantitative proof that human-influenced planning will result in better performance than if the AI agents perform the task with no additional assistance. We also anticipate qualitative results from survey questions will yield suggested improvements that can be incorporated into future user interfaces as well as suggestions for easier methods of interaction.

In addition to performing the human subject testing described here, future extensions of this work include addressing the problem of collaborative planning when AI agents do not have continuous communication with the human supervisor as well as each other. Additional extensions include expanding the hierarchical planning structure to include scenarios requiring task scheduling for long-duration missions, as well as scenarios requiring multiple human operators collaborating with multiple AI agents. The multi-human-multi-agent scenarios introduce opportunities to incorporate methods for sharing unmanned assets already in development. Finally, this research has been conducted solely with simulated AI agents, so a natural extension is implementation of the AI planning agents on actual UxVs and ground station for real-world testing, experimentation and demonstration.

ACKNOWLDEGMENTS

The authors thank Luiz Martinez, Eric Gustafson and James Jen, the software development team instrumental in developing and building the human subject test framework, as well as integrating the single-human-single agent interaction mechanism into the IMPACT System Task Manager. The research presented here has been funded by SPAWAR Systems Center Pacific's In-House S&T Innovation Program. This paper is approved for public release.

REFERENCES

[1] Work, B. 2015. "The Third U.S. Offset Strategy and its Implications for Partners and Allies," January 28, 2015. Willard Hotel, Washington, DC.

[2] Wang, H., M. Lewis, P. Velagapudi, P. Scerri, and K. Sycara. 2009. "How search and its subtasks scale in N robots." In *Proceedings of the 4th ACM/IEEE international conference on Human robot interaction* (pp. 141-148). ACM. 9-13 March. La Jolla, California, United States.

[3] Russell, S., and P. Norvig. 2010. *Artificial Intelligence: A Modern Approach*. 3rd ed. Prentice Hall, Upper Saddle River, NJ.

[4] Long, D., and M. Fox. 2002. "Progress in AI Planning Research and Applications," *UPGRADE: The European Journal for the Informatics Professional 3(5)*:10–25.

[5] Zilberstein, S. 2015. "Building Strong Semi-Autonomous Systems." In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*, (pp. 4088-4092). 29-30 January, Austin, Texas, United States.

[6] Allen, J. F., C. I. Guinn, and E. Horvtz. 1999. "Mixed-initiative interaction." *IEEE Intelligent Systems and their Applications 14(5)*: 14-23.

[7] Ferguson, G., J. F. Allen and B. W. Miller. 1996. "TRAINS-95: Towards a Mixed-Initiative Planning Assistant." In *Proceedings of the Third International Conference on Artificial Intelligence Planning Systems (AIPS 1996)*, (pp. 70-77). AAAI, 29-31 May, Edinburgh, Scotland.

[8] Bresina, J. L., A. K. Jónsson, P. H. Morris and K. Rajan. 2005. "Mixed Initiative Planning in MAPGEN: Capabilities and Shortcomings," In *Workshop on Mixed-Initiative Planning and Scheduling at the International Conference on Automated Planning and Scheduling (ICAPS 2005)*, (pp. 54-61). 5-10 June, Monterey, California, United States.

[9] *Google Optimization Tools*. 10 July 2018. Google. https://developers.google.com/optimization/. Accessed 14 August 2018.

[10] Nau, D., T.-C. Au, O. Ilghami, U. Kuter, J. W. Murdock, D. Wu, and F. Yaman. 2003. "SHOP2: An

HTN Planning System," *Journal of Artificial Intelligence Research (JAIR) 20*:379–404.

[11] Molineaux, M., M. Klenk, and D. Aha. 2010. "Planning in Dynamic Environments: Extending HTNs with Nonlinear Continuous Effects." In *Proceeding of the Twenty-Fourth AAAI Conference on Artificial Intelligence* (pp. 1115–1120). July 11–15, Atlanta, Georgia, United States.

[12] Sohrabi, S., J. A. Baier, and S. A. McIlraith. 2009. "HTN Planning with Preferences." In *Proceedings of the Twenty-First International Joint Conference of Artificial Intelligence (IJCAI 2009)* (pp. 1790–1797). July 11–17, Pasadena, California, United States.

[13] Borrajo, D., A. Roubíčková, and I. Serina. 2015. "Progress in Case-Based Planning," *ACM Computing Surveys (CSUR) 47(2)*: Article No. 35.

[14] Boutilier, C., T. Dean, and S. Hanks. 1995. "Planning Under Uncertainty: Structural Assumptions and Computational Leverage." In *Proceedings of the Second European Workshop on Monte Carlo Treatment Planning (September)* (pp.157–171). October 19–21, Cardiff, United Kingdom.

[15] Kaelbling, L. P., M. L. Littman, and A. R. Cassandra. 1998. "Planning and Acting in Partially Observable Stochastic Domains." *Artificial Intelligence 101(1)*: 99–134.

[16] Meuleau, N., K. E. Kim, L. P. Kaelbling, and A. R. Cassandra. 1999. "Solving POMDPs by Searching the Space of Finite Policies." In *Proceedings of the Fifteenth conference on Uncertainty in Artificial Intelligence* (pp. 417–426). July 30–31, Stockholm, Sweden.

[17] Campbell, T., and J. P. How. 2015. "Bayesian Nonparametric Set Construction for Robust Optimization." In *Proceedings of the 2015 American Control Conference (ACC)* (pp. 4216–4221). July 1–3, Chicago, Illinois, United States. IEEE.

[18] Brown, N., and T. Sandholm. 2017. "Superhuman AI for heads-up no-limit poker: Libratus beats top professionals." *Science 359*: 418-424.

[19] Mnih, V., et al. 2015. "Human-level control through deep reinforcement learning." *Nature 518*: 529-533.

[20] Schulman, J., F. Wolski, P. Dhariwal, A. Radford, and O. Klimov. 2017. "Proximal policy optimization algorithms." *arXiv preprint arXiv:1707.06347*.

[21] *OpenAI Five*. 25 June 2018. OpenAI. https://blog.openai.com/openai-five/. Accessed 20 August 2018.

[22] Ghallab, M., D. Nau and P. Traverso. 2004. *Automated Planning: theory and practice*. Morgan Kaufmann Publishers Inc., San Francisco, CA.

[23] Thrun, S., W. Burgard and D. Fox. 2005. *Probabilistic robotics*. MIT Press, Cambridge, Massachusetts.

[24] Kelley, L., M. Ouimet, B. Croft, E. Gustafson and L. Martinez. 2017. "An Approach to Integrating Human Knowledge into Agent-Based Planning." In *Workshop on Impedance Matching in Cognitive Partnerships* at the *2017 International Joint Conference on Artificial Intelligence (IJCAI-17)*. 21 August, Melbourne, Victoria, Australia.

[25] Behymer, K., C. Rothwell, H. Ruff, M. Patzek, G. Calhoun, M. Draper, S. Douglass, D. Kingston, and D. Lange. 2017. "Initial evaluation of the intelligent multi-uxv planner with adaptive

collaborative/control technologies (IMPACT).” Infoscitex Corp. Beavercreek, OH.

[26] Lange, D. and J. Reeder. 2017. “IMPACT machine learning efforts.” In *Proceedings of SPIE Unmanned Systems Technology XIX*. *10195*. 11 April, Anaheim, California, United States.

[27] Okabe, A., B. Boots, K. Sugihara and S. N. Chiu. 2009. *Spatial tessellations: concepts and applications of Voronoi diagrams*. Vol. 501. John Wiley & Sons, Ltd. Chichester, England.

[28] Kuhn, H.W. 1955. "The Hungarian method for the assignment problem." *Naval research logistics quarterly 2.1-2*: 83-97.

[29] Hart, S. G., and L. E. Staveland. 1988. “Development of NASA-TLX (Task Load Index): Results of empirical and theoretical research.” *Advances in Psychology, 52*: 139–183.

[30] Hart, S. G. 2006. “NASA-task load index (NASA-TLX); 20 years later.” In *Proceedings of the Human Factors and Ergonomics Society Annual Meeting, 50*: 904–908.

[31] Endsley, M. R., and D. J. Garland (Eds.). 2000. *Situation awareness analysis and measurement*. CRC Press. Boca Raton, FL.