Adjusting Autonomy and Reliance on Agents in Human Supervised UCAV

Henry Hexmoor and Srinivas Battula
Computer Science & Computer Engineering, University of Arkansas
Fayetteville, AR 72701

ABSTRACT

In this paper we offer case studies of empowering agents with adjustment of cognitive notions of autonomy and trust that enable them to have a more socially adept interaction with a human supervisor. The application domain is control of unmanned aerial vehicles. Agents learn to change autonomies as they observe they learn the relationship between their assumed autonomy and performance. Agents also learn to change their reliance on human supervision as it takes different lengths of time.

1. INTRODUCTION

Agent-Human interaction is an emerging area of research in agent-based systems. In general, agent-based computing has been beneficial in four areas and in all of these areas agent-human interaction can be found. First, agents are used in automation of dirty, dull, and dangerous as well as tedious, boring, and routine tasks to relieve humans of such duties [6]. Examples in this area can be found in agents embodied in desktop assistants [7] or intelligent in service of humans. Human supervisors benefit greatly from delegating tasks to such agents [5]. Secondly, agents are used to produce an improved human sense of "presence" for humans collaborating in physically disparate locations. Examples in this area are found in agents in knowledge management tasks like war-rooms and human users benefit from agents who proxy for their human counterparts. Third, agents can be used in democratization of computing, services, and support. Examples in this area are agents in municipal functions such as the department of motor vehicles or public libraries and virtual museums. Here, the public enjoys the benefits of agent services. Fourth, agents are used in reduction of redundancy and overlap due to competition. Research in this area can increase collaboration between agent collectives such as in institutions, organizations, and teams. Examples in this area are found in agents that facilitate tracking and sharing power or telecommunication services. In human-agent interaction, agents might be cognitive assistants capable of discovering human preferences, personality, and emotions. With this, agents will gain human trust along with permission to assure increased autonomy while providing greater human control. Agents will also form social networks that will facilitate their greater ability to work together and to collaborate. We envision agents that will be socially adept. This contributes to robustness and adaptability of collaborative enterprises. Theories and models of human-agent interaction are needed as part of collaborative enterprises to provide foundations for constructing systems able to work with each other and with the people using them.

In complex tasks where humans and agents share control and both make decisions, human-agent interaction must accommodate mixed initiatives. Agents that allow human intervention in their actions are said to have adjustable autonomy [1, 2, 4]. Here an agent's autonomy can be varied dynamically. Adjustable autonomy holds the promise of lowering human controller's burden of continuously controlling the agent and alleviates the agent from dependence on the human controller. The delays involved in making a decision on behalf of the agent and conveying it to the agent may decrease the agents' performance. Time-critical systems cannot afford this delay and in some cases a medium quality decision made by an agent in time will be superior to a high quality decision by a human-controller which may not be timely. However, there are situations where not all decision-making autonomy can be given to an agent based on the assumption that an agent makes good decisions.

In the rest of this paper we will begin by describing our implemented testbed that we used to experiment with adjustable autonomy and human-agent interaction [8]. We will then present a few empirical results and end with concluding remarks.

2. TEST-BED

Our simulator consists of a mountainous terrain with SAM's and a number of UCAV's that fly over them [3]. All the UCAV's are partially autonomous agents whose autonomy can be adjusted by the human controller dynamically. UCAV's starting from the base, fly over the mountainous terrain to reach their destination. UCAV's and SAM's have a visible region within which they can attack one another. *Hit probability* is the probability of the UCAV being hit by the SAM's. If the hit probability of a UCAV crosses a certain limit the UCAV's are considered to be shot-down by the SAM's and they disappear from the simulator. In addition, if two UCAV's are *in coalition* the hit-probability of both the UCAV's decreases by a factor due to the confusion of the SAM(s). A SAM tries to hit the plane as soon as it enters its visible region. When a UCAV comes across a SAM in its course to reach the destination it may *Start Avoiding* the SAM by itself or may ask help from other agents to attack the SAM. The UCAV initially tries to avoid the SAM until its hit probability crosses a certain limit. When the hit probability crosses a limit, it requests for help from other agents. Hit probability of an UCAV is proportional to the number of SAM's that have UCAV in their Visible region. Each Agent gets its turn sequentially. A *cycle* is completed when all the agents get their turn once. The *cycle* continues until all the agents reach their destination [9].

The following are the 10 states that govern the behavior of an agent in the simulator

1) *Fly to Target*: This is a default state. In this state the agent's goal is just to reach the destination.

2) *See SAM*: The agent enters this state as soon as it sites a SAM in its visible region. In this state the agent reasons whether to avoid SAM by itself or to seek help
3) *Start Avoiding*: The agent enters this state when it is in the visible region of a SAM and can avoid the SAM by itself.
4) *Waiting for Help*: The agent enters this state if the hit-probability crosses a certain limit and its better to seek help from other agents than avoid by itself. In this state the agent waits for help from other agent.
5) *Offering Help*: The agent enters this state when any other agent of the system is waiting for help. In this state the agent offers help, which may be accepted or rejected by the help-needing agent.
6) *Being Helped*: The agent enters this state when one of the agents has agreed to offer help and it is willing to accept it.
7) *Helping*: The agent enters this state when another agent accepted help offered by this agent and it is on its way to help the agent.

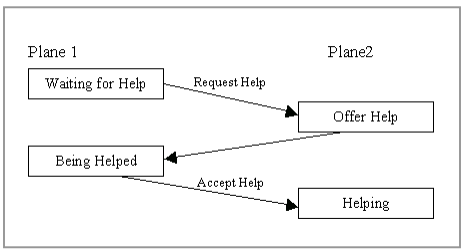

Figure 1 Help Scenario

8) *Helping and See SAM*: The agent enters this state when it is helping another agent and on its way sites a SAM itself.
9) *Helping and Avoiding SAM*: This state is a result state to the previous one.
10) *In Coalition*: When an agent helps another agent the helping agent and the helped agent form a Coalition. They break the Coalition only after both of them are out of any of the SAMs visible region. The hit-probabilities of the planes decrease considerably with the two agents in coalition.

The following are the set of permissions that an agent requires to operate completely autonomous in the system.

1) *Attack SAM*: The agent needs to have this permission set to attack a SAM as soon as it sites it.
2) *Avoid SAM*: The agent needs to have this permission set to avoid SAM by itself. An agent that doesn't have permission to avoid enters *Waiting for Help* state
3) *Get Help*: The agent needs this permission set to accept help offered by other agent. An agent that doesn't have this permission has no choice of selecting the helping agent.
4) *Offer Help*: The agent needs this permission set to offer help to other help-needing agents.
5) *Help*: The agent needs to have this permission set to help other agents when another agent accepts it.

Human controller sets the permissions of an agent initially when an agent is created. The more permission's the agent has the more autonomous is the agent. For example, if an agent wants to attack a SAM and doesn't have the permission to perform the action it has to get permission from the human controller. Human controller can give the agent permission to attack, or deny permission. In addition, if an agent is given all the permissions and later if the human agent wants to change it he can do so by changing the agent's autonomy dynamically.

When an agent asks for a permission from the human controller to act in a particular situation the human controller has to make quick and wise decisions that improve the system performance. The human controller can be in two states *Busy* (i.e., responding to another agent) or *Idle*. The agent has to wait until the human controller makes a decision and conveys the decision to it. In our simulator the overall system performance increases with the decrease in average hit-probability of the agents. When an agent enters the Visible region of a SAM the hit-probability increases with time until the agent gets out of the visible region by avoiding SAM or another agent comes to rescue. The agent's hit-probability at each cycle is recorded. The hit-probability of an agent can be as low as 0.0 when it is not in visible region of any of the SAM's or as high as 0.8, which we set as a higher limit on hit-probability. Agent's which have hit-probability of greater than 0.8 are considered shot down by the SAM's.

Cumulative Hit-Probability (CHP) of an agent is the sum of hit-probabilities of the agent in each cycle through out the simulation run (i.e. from the time the agents took off from the base until they reach their destination) divided by the number of cycles the agent has hit-probability greater than 0.0. *Average Hit-Probability* (AHP) is the average of the cumulative hit-probabilities of each agent.

$$\text{CHP} = \frac{\sum_{i=1}^{l} p_i}{n} \qquad \text{AHP} = \frac{\sum_{j=1}^{m} chp_j}{m}$$

p_i – Hit-Probability of agent at cycle i
n – Number of cycles in which hit-probability of the agent is greater than 0.0
l – Total number of cycles in a simulation run
chp_j – Cumulative Hit-Probability of agent j
m – number of agents

Agents that require permission from the human controller add permission message to a queue from which the controller first selects it and gives the permission or denies the permission. There is a delay involved from the time the agent adds the permission message to the queue and gets a response from the human controller. The factors that affect the delay are

1) The number of permissions asked by other agents
2) The human computer interaction system
3) The efficiency of the human controller (we ignored this in our simulator)

Here some important questions arise.

1) How long should the agent wait for the human permission
2) Should the agent take over control and make an autonomous decision.

The permissions given by the human controller are recorded in the agent's history together with the hit-probability of the agent when the permission is given and cycle number. Other agents use these permissions when they are in a similar situation. Two agents are considered to be in a similar situation if they require the same kind of permission, the permission

given is not more than 30 cycles old and the hit-probabilities are close. Permissions given to an agent that are 30 cycles old become invalid.

The delay in communication between the human controller and the agent deters the performance of the system. With the increase in number of agents the number of decision's to be given by the human controller increases and this degrades the performance of the system further. So we have set an upper limit on the waiting time beyond which further waiting of the agent has degrading effects.

To make a decision autonomously the agent needs to know

- when should it Start Avoiding,
- when should it Wait for help,
- when should it offer help,
- when should it accept help,
- when should it help

The predefined rules set to the above actions are

- An agent starts Avoiding when it sites a SAM and its hit-probability <= 0.2 .
- An agent waits for help when it is avoiding a SAM and its hit-probability > 0.2 .
- An agent offers help when it is in *Fly to Target* state and some other agent needs help.
- An agent, which is waiting for help, accepts help if another agent offers it.
- An agent helps another agent that accepted its help offer.

After waiting for a maximum time limit the agent chooses to follow one of these rules. The agent's decision may not be convincing in all situations. In those cases the human controller can interrupt the agent and gives his decision to the agent.

3. EXPERIMENTS AND RESULTS

We will discuss results for adjustable autonomy as well as trust. We begin with four adjustable autonomy scenarios discussed.

3.1 Adjustable Autonomy

The x-axis in Figures 2 to 5 represents the number of agents taking part in a simulation run and the y-axis represents the hit-probability averaged over of all agents in the scenario. Figure 2 shows the average hit-probabilities of 2, 3 and 4 agents when all the decisions have been autonomous, i.e., without human control. Permissions required to make autonomous decisions by an agent are given to each agent. From the figure we can observe that the average hit-probability remains almost constant. All agents follow a predefined rule set in making a decision.

Figure 3 shows the average hit-probabilities of 2, 3 and 4 agents when the human controller makes all the decisions. i.e., the agents are completely controlled by the Human. We observe that the average hit-probability increases with the increase in number of agents. With increased number of agents, the human controller is flooded with more requests from agents for permissions. Therefore, the delay in making a decision increases the hit-probability for the waiting agents. The increase in hit-probability is more between 3 and 4 agents than between 2 and 3 agents. Figures 4 and 5 shows changes in average hit-probability when the agent's autonomy can be adjusted dynamically. We have considered two cases in which the autonomy of the agent will be varied dynamically. In the first scenario, an agent doesn't wait for the human controller's decision and makes an autonomous decision based on the rule set and continues with it. However, if the human controller feels that the agent did not make a wise decision he can override it with his decision and ask the agent to proceed according to the new decision.

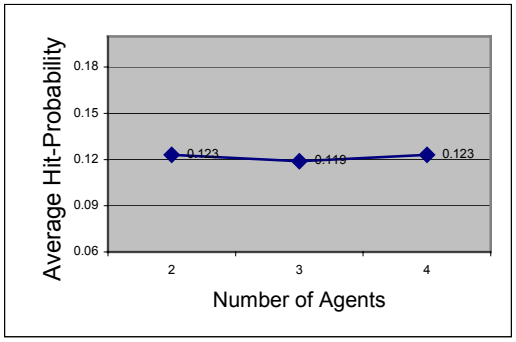

Figure 2. Agent Controlled

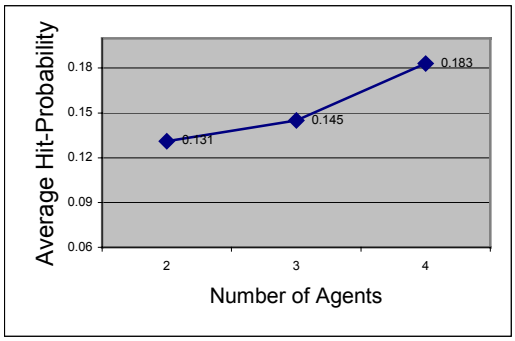

Figure 3. Human Controlled

Figure 4 shows the average hit-Probabilities of 2, 3 and 4 agents.

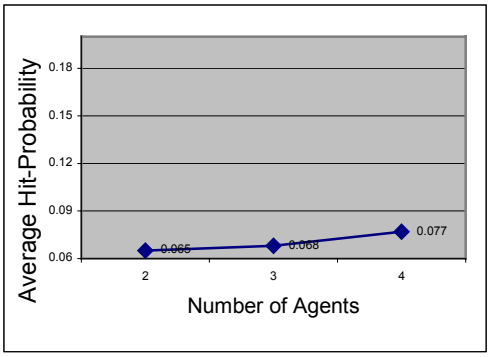

Figure 4. Autonomous Decision made by Agent, which can be reverted by human controller later.

The second scenario is where the state of an agent can't be changed or a decision made once can't be reverted. Here we have set an upper limit on the number of waiting cycles the agent waits before making an autonomous decision. If the human controller feels that the agents can be given more autonomy he can decrease or increase the waiting cycles. In figure 5 we observe that the hit-probability increases with increase in number of agents but this increase is considerably less than the increase in figure 3 in which the human controller makes all the decisions.

Reasoning about *reliance* as a form of trust between human and agent is another method to manage adjustable autonomy where the human controller has the most control over the agents. This is presented next.

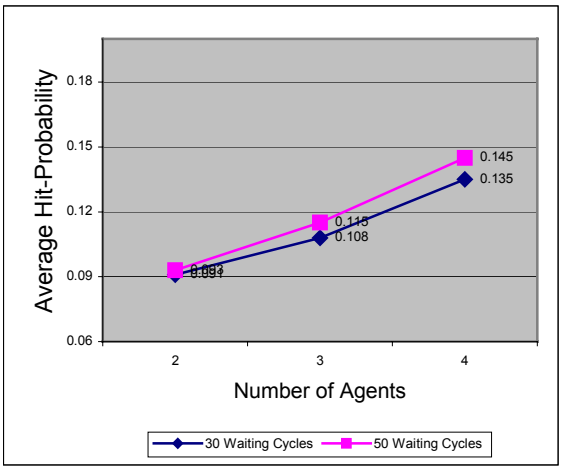

Figure 5. Autonomous Decision made by agent after waiting for certain time limit, which can't be reverted by human controller.

3.2 Reliance

Let's consider that the human controller doesn't have control over the agents but all the agents presume that a human controller's decision is better than a decision made by them. To reiterate, the delay involved in giving a decision increases with the number of requests, agents cannot wait for the human's decision beyond a certain point, which may increase their hit-probabilities. Agents conceive of a Global Human Reliance Value (GHRV) for human decisions with a maximum value of 5[10]. The GHRV gives a measure of the degree of trust agents has over the human controller [11]. The reliance increases if the human controller responds to a request and decreases if the human controller fails to respond to a request in time (say 40 waiting cycles). Agents wait for human controller's decision for a certain amount of waiting cycles based on the reliance value. If the reliance value is low, agents wait for less time before making an autonomous decision. In our simulator agent's waiting time is governed by the following equation:

$$\text{Maximum\_Waiting\_Cycles} = 15 + \text{GHRV} * 5$$

Therefore, each agent waits for at least 15 waiting cycles before making an autonomous decision even if the reliance value is 0. When the human controller gives a decision the value is incremented by 1. So in cases when a human controller is flooded with requests he fails to respond to some of the requests, which decreases the GHRV. With less GHRV agents that require permission wait less before making an autonomous decision. This considerably decreases the number of waiting cycles agents wait for a permission and unsuccessful in getting the human controller's response.

As we have already mentioned human controller's decision is stored in the agents' history. Two agents in a similar situation can use the same response of the human controller. In addition agents also have Agent-Agent reliance values among them. Each agent's reliance on other agents is maintained in an Agent-Agent Reliance Value (AARV) array. Agent-Agent reliance value also varies between 0 and 5. An agent first interacts with other agent it relies on most to check if that agent has received a response from the human controller to perform the same action. The other agent responds to the agent's request as follows

1) returns 0 if the human controller has not given permission to perform the action
2) returns 1 if the human controller has given permission to perform the action
3) returns 2 if it couldn't find it in its history

Reliance value increases by 1 in the first two cases where the agent returns 0 or 1. It decreases by 1 if the agent returns 2 i.e. agents rely more on an agent that provides them with information that is useful in making an autonomous decision. The following figures illustrate the number of interactions between agents and the number of human-agent interactions and how average hit-Probability is affected when Agent-Agent and Human-Agent reliance is considered. In figure 6 we can observe that the average hit-probability increases with increase in the number of agents. The average hit-probabilities in this case are very much similar to the average hit-probabilities of figure 5. Figures 7 and 8 give the number of interactions between human-agent and agent-agent. With increase in the number of agents the interactions between agents increase rapidly, however there is not much increase in the human-agent interactions.

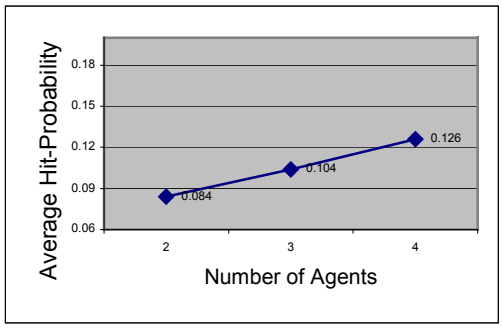

Figure 6 Average Hit-Probabilities of agents when agent-agent and human-agent trust is considered

4. CONCLUSION

The aim of human-agent interaction is to design interfaces and cognitive approaches that increase access of human and agent over one another's decision making process. In this paper we have experimented with adjustable autonomy of agents and shown a few tradeoffs than be made to increase control of human while preserving agent performance. Dynamic adjustment of agent wait cycles for a human decision as well as experience an agent gains from waiting are two specific methods we have explored. The results are promising and show the way to similar methods.

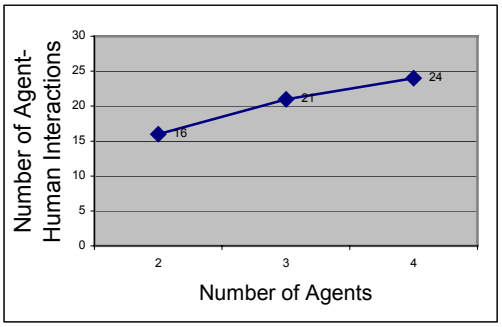

Figure 7 Number of interactions between human controller and agent.

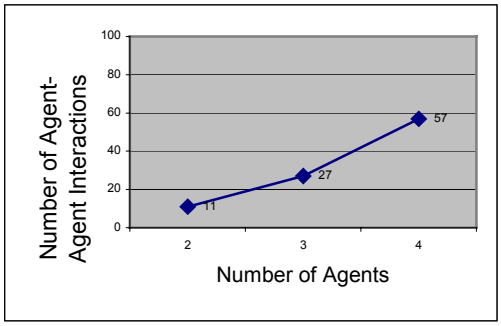

Figure 8 Number of agent-agent interactions

5. ACKNOWLEDGEMENT

This work is supported by AFOSR grant F49620-00-1-0302.

6. REFERENCES

[1] G.A. Dorais, R.P Bonasso, D. Kortenkamp, B. Pell, and D. Schreckenghost, 1998. "Adjustable Autonomy for Human-Centered Autonomous Systems on Mars", **In Proceedings of the First International Conference of the Mars Society**.

[2] M. A. Goodrich, D. R. Olsen jr., J. W. Crandall, and T. J. Palmer, 2001. "Experiments in Adjustable Autonomy". **In Proceedings of the IJCAI-01 Workshop on Autonomy, Delegation, and Control: Interacting with Autonomous Agents**.

[3] H. Hexmoor, and X. Zhang, 2002. "Socially Intelligent Air Combat Simulator", **In Proceedings of The Seventh Pacific Rim International Conference on Artificial Intelligence, (PRICAI-02), Tokyo, Japan**.

[4] P. Scerri, D. Pynadath, and M. Tambe, 2002. "Towards adjustable autonomy for the real-world". **Journal of AI Research** (JAIR), Volume 17, Pages 171-228.

[5] D. Shapiro, 2001. "Value-driven agents", Ph.D. thesis, Stanford University, Department of Management Science and Engineering.

[6] P. Maes, 1994. "Agents that Reduce Work and Information Overload". Commun. **ACM** 37,7, 31-40

[7] M. Huhns and M. Singh, 1998. "Personal Assistants," **In IEEE Internet Computing**, Vol. 2, No. 5: Sept-Oct 1998, pp. 90-92, IEEE press.

[8] P. Scerri and N. Reed, 2001. "Designing Agents for Systems with Adjustable Autonomy". **The IJCAI-01 Workshop on Autonomy, Delegation, and Control: Interacting with Autonomous Agents**.

[9] H. Hexmoor and S. Battula ,2003. "Towards Collaboration between Human and Social Agents that mind Human Social Personality". **International Symposium on Collaborative Technologies and Systems**. (to appear)

[10] S. Marsh, 1992. "Trust and reliance in multi-agent systems: a preliminary report**". In Proceedings of the 4th European Workshop on Modeling Autonomous Agents in a Multi-Agent World**, Rome.

[11] Y.T. Tang, P. Winoto & X. Niu, 2002. "Investigating Trust between Users and Agents in a Multi agent Portfolio Management System: a Preliminary Report". **In the Workshop on business agents and the semantic web, Fifteenth Canadian Conference on Artificial Intelligence**.