Socially Intelligent Combat Air Simulator

Henry Hexmoor *♦, Xin Zhang *

* Department of Computer Science & Computer Engineering
University of Arkansas
Fayetteville, AR 72701

♦Center for Multisource Information Fusion
Department of Industrial Engineering
State University of New York at Buffalo
Buffalo, NY 14260
{hexmoor, xxz03}@uark.edu

Abstract. We are exploring teaming in complex and dynamic environments that requires reasoning about roles and social influences. Towards that end we describe our implemented testbed and experiments that empirically show effects of such reasoning in a team setting.

1 Introduction

We have developed a testbed to explore teamwork and explicit reasoning about roles. We are seeking methods whereby agents reason about teamwork issues such as forming, joining, and working with teammates. Beyond the model of teamwork such as Cohen and Levesque's now famous Joint Intentions theory [2], there has been some work on monitoring team effectiveness and tools for composing teams; Karma-Teamcore is an example [7]. Tambe's work describes a methodology he calls Team Oriented Programming whereby a human designer interacts with the system to determine roles and individuals and assign roles and plans to individuals. Our developments are similar but the intent is for the agents to autonomously determine when and how they work with a team, and there is no notion of an external system designer. In this paper we present our testbed and discuss preliminary results of contrasting strategies for agents who might work in a team. Elsewhere, we considered fundamental issues about the nature of teams of agents and the mental states of the agents required for the formation of a team [1]. We put forward some criteria for the existence of a team as a guide for the development of systems of artificial social agents. We posited that a "*team*" is an aggregation of agents with a common mental state that includes having a common intention, awareness of being a member of the team and the team's abilities, and a cooperative attitude. Furthermore team members must have autonomy in order to form intentions. Later we extended that work to

measure team effectiveness [3]. This paper offers empirical results for effectiveness in a class of teams.
In our implemented testbed three or more fighter aircraft agents have the mission to deliver a bomb over a remote designated site. There is a one to one relationship between agents and planes. Artificial agents control all the planes except one, which is controlled by a human operator. The human operator controls its plane in the field along with the other planes, and will have similar visual and auditory sensing as well as similar flight maneuvering capabilities. The system is implemented in Java.

We simulate five Surface to Air Missile sites (SAMs), which are randomly generated each time when the program starts running. Figure 1 shows the main simulator screen with two agent-controlled planes flying close together followed by a human-controlled plane. A user friendly GUI allows the user to see the base runway (lower left in Figure 1), mountains, terrain, target (beyond the screen in the upper right), and SAMs. Figure 1 shows one of the planes firing just short of a SAM site. As the agent interacts, a message window pops up to show the aircraft agent information, shown in Figure 2. The autonomous agents are allowed 10 atomic roles based on the situation. There is also several buttons in the main GUI (Figure 1) allowing users to stop or run the application at any time. Running is indicated by "play" in Figure 1.

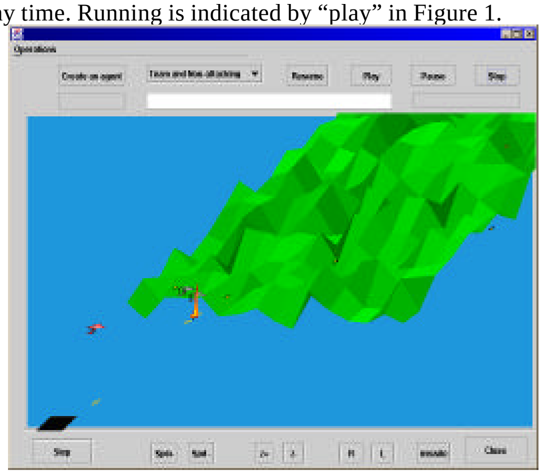

Figure 1 Testbed and its GUI

Sensors onboard planes report locations of threats from SAMs. SAMs can be either averted or neutralized. The planes flying over the terrain with SAMs have limited sensing ability. Therefore, judicious sharing of sensed data as well as sharing fire power are important. SAM sites move small random distances. SAMs have limited capabilities for shooting the aircraft in their range modeled with probabilities, e.g. 5% when a SAM has full power. SAMs are distracted and have more difficulty when there are multiple planes, or when the agent has already been informed of the SAM position by its teammates. We model the SAM's ability to shoot by calculating the dynamic probabilities of downing planes. The agent's overall decision to act comes from consideration of the conversational and physical states, as well as the status of its social network. The agents are by and large autonomous in order to react to the environment. For examp le, after taking off from the base station, agents can fly toward targets while avoiding SAMs. *Roles* are used here to describe the agents' performances at different stages.

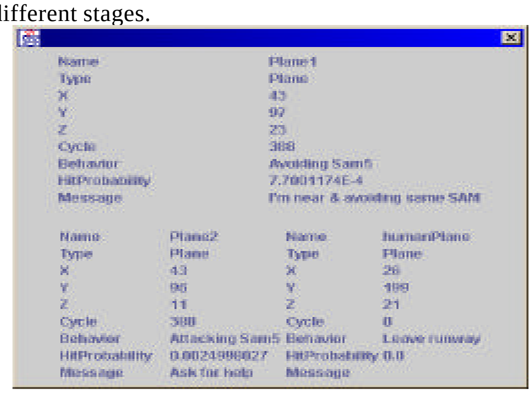

Figure 2 Agent information pop up window

The agent can adopt different roles to perform a variety of functionalities according to the environment. Agents and roles don't need to follow one to one assignment. Different agents can adopt a given role simultaneously, but one agent can only adopt one role at a time.

In the remainder of this paper, we will present the team-oriented ideas in the testbed, and discuss how agents handle roles in team or non-team formation. We will then outline an experiment where we measure the time to complete the mission versus the probability of being shot down under a few strategies, which we call top level roles. We then review related work and offer concluding remarks.

2 Team Roles

2.1 Role Hierarchy

We have developed a role hierarchy based on the testbed, shown in Figure 3. The following ten atomic roles are at the lowest level: Take-off, Fly-toward-target, Avoid-sam, Avoid-airplane, Attack-sam, Go-around, Send-message, Receive-message, Respond-message and Land. The agent will assume only one of these ten atomic roles at a time. Take-off is the role that the agent adopts when it has just departed from the base station. The agent will take Fly-toward-target role at later time in order to fly over the terrain to the target. Agents take the Avoid-airplane role when two agents are too close to each other in order to avoid collision. Avoid-sam is the role that agents assume to avoid the danger posed by a SAM site after detecting the SAM. Other types of atomic roles will be explained later.

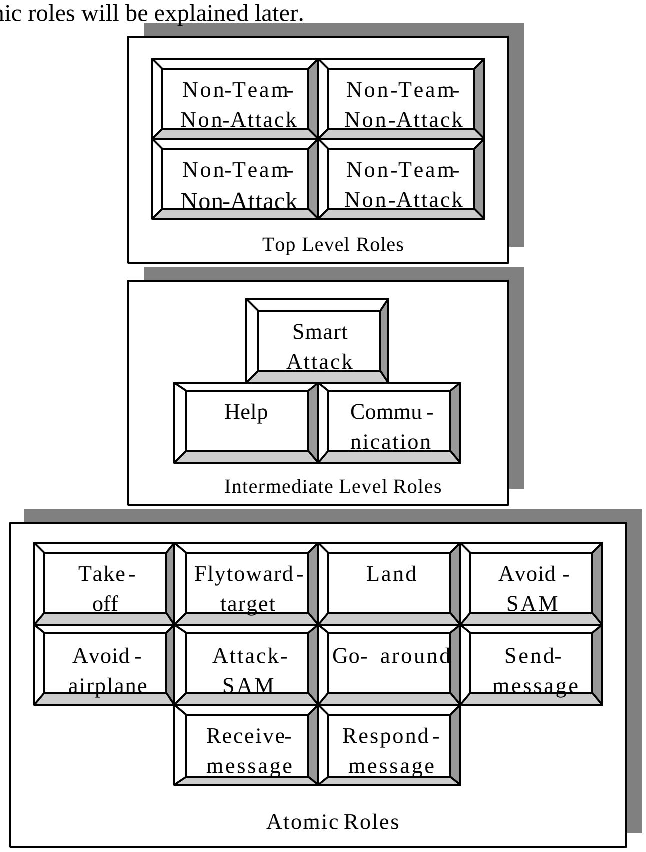

Figure 3 Role Hierarchy

More complex roles are composed of atomic roles for more sophisticated capabilities. Help, Communication, and Smart-Attack are three intermediate roles. Help is a social action that agents in a team use to fly together and to attack a SAM site. If one agent needs help, other agents can assume Help roles to help the agent confronting a SAM in its attack on the SAM. Communicate is a speech act for agents to inform one another about their locations as well as status of help. The Communication role is composed of three atomic roles of Send-message, Receive-message, and Respond-messages. When an agent is a part of a team, Communication role is assumed by a team member to exchange information with other agents. An agent takes Smart-Attack role when the agent sees a SAM and has committed to attack. Smart-Attack Role is composed of Attack-sam role and Go-around role, which means the agent attacks the SAM while maintaining a safe distance and moving around the SAM toward the target. The Attack-sam role alone does not take self-protection and moving toward target into account.

Combining atomic and intermediate roles, we defined four complex roles at the topmost level for the testbed: Non-Team-Non-Attack, Non-Team-Attack, Team-Non-Attack, and Team-Attack as the main roles. When adopting Non-Team-Non-Attack role, each agent performs independently, and the agent does not attack the SAMs. After the agent takes off from the base, it will fly toward the target. When it sees SAMs, it will avoid the SAMs until land. Non-Team-Attack role is the same as Non-Team-Non-Attack role, except that the agents assume Smart-Attack role. When a SAM is detected within their visual range they could attack but since the distances are far, their attack will be inaccurate. In this role, agents wait until they reach closer to the SAM for more accurate attack.

When the agents take Non-Team roles (both in Attack and in Non-Attack roles), they do not communicate or consider one another in their decision making process. Under Non-Team roles, the agent is independent, and will fly without taking account of other agents.

In Team roles (both in Team-Non-Attack and Team-Attack roles), all the agents will be in one team, and are acting as team members while performing their own task. Each agent will consider the performance of other agents via Communication, or Help roles. In Team-Non-Attack role, agents will communicate with one another to stay informed of such as the location of the SAMs they have detected. Therefore, even if an agent doesn't see any SAMs on its way to the target, it might already know where SAMs are by getting the messages from other team members. Communication provides agents with broader viewing areas (since the agent can "see" what the others "see"), thus have higher chance of avoiding SAMs. Since the agents in Team-Non-Attack role doesn't have attack features, they don't need to fly near to the agent to help each other.

In Team-Attack role, agents will form a team to aid one another in attacking a SAM site within certain distance. When agents form a team, each agent can communicate with other agents, such as the agent can inform the team member of seeing a SAM, or when the agent is in danger, it can ask for help from other agents. When an agent receives a "need help" message from other agents, and is willing to go to help, it will assume the Help role. After adopting the Help role, the agent will change its flight direction from going toward the target to going near the agent who needs help, and attack the SAM with that agent. The Help role only appears in the Team-Attack role to help other agents attack the SAMs.

2.2 Role Transitions and Exchanges

An agent can adopt only a single atomic role at one time. The initial role for each agent is Take-off. Under certain situations, an agent relinquishes its role for another. If the role change is irreversible we will call this role change a *transit*. For example, consider an agent who starts with the role of Take-off from the base station. After some time, the agent will change to the role of Fly-toward-target. After the agent decides to adopt Fly-toward-target role, it cannot go back to the Take off role. If the agent can return to the current role (i.e., reversible) we will call the role change an *exchange*. For an example, consider that the agent is under the Attack-sam role. After the agent successfully avoids the SAM or destroys the SAM, the agent can change its role back to Fly-oward-target once again.

Figure 4 shows role transition and role exchange in our testbed. For Take-off role, the agent takes off from the base station. After some time cycles, the agent will transit its role to Fly-to-target. If the agent doesn't see anything in its visual field, it will keep this role until it adopts the Land role for landing on the destination. If the agent encounters a SAM before reaching the target, it can decide to avoid it or attack it. To avoid a SAM, the agent exchanges its role to Avoid-sam role. Avoid-sam role will determine which direction the agent should turn and when it is safe enough for the agent to exchange back to Ffly-to-target role. To attack a SAM, the agent exchanges its role to Attack-sam role. This role will determine how to attack the SAM, and also this role will be combined with Go-around role to find the best point to make the attack. When two agents fly too near to each other, they will adopt Avoid-airplane role, which is similar to the Avoid-sam role. When the agent is within the landing field, it will transit its role to the Land role and land in the destination.

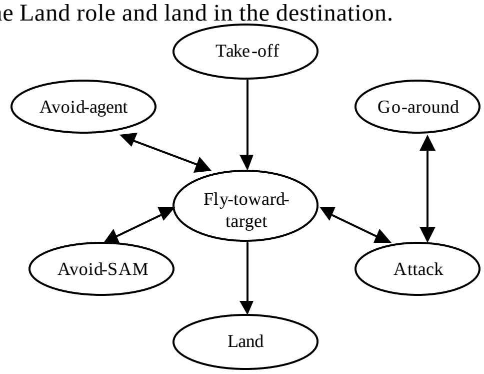

Figure 4 Role transitions and exchanges

3 Experimental Setup and Analysis

In the testbed experiment, we are interested in measuring the effectiveness of each of the four top level roles. In the testbed, at any time agents are in danger of being shut down by the SAMs. The most dangerous situation is when an agent encounters a SAM for the first time. Each time the agents pass by the SAMs, they will risk being shot down by the SAMs. We can use the probability of the agent being shot down (Hit Probability) as the main value of measuring the effectiveness of the top level roles. If the probability of being shot down is very low, we can say that effectiveness of the role is high; otherwise, the effectiveness is low. In order to measure the probabilities of being destroyed, we set up several parameters to help analyze the testbed.

3.1 Experiment Parameters

We have developed a probabilistic model of planes being shot down that is based on a model of the SAM's power and accuracy rate. We do not consider partial damage, and each hit is considered fatal. Initially, each SAM's power is 100%, and its accuracy to shoot down an agent-controlled plane is 5%. When an agent attacks a SAM, the SAMs' power is decreased 2%. After the initial encounter with each subsequent encounter, the SAM's accuracy is halved in team-attack or team-non-attack roles. This is due to the fact that the agent who has met the SAM first will inform the other agents about the SAM's position by communication. We model each agent's probability of being shot down when passing by each SAM as equation (1):

$$\textit{Hit Probability} = \textit{(SAM's power)} * \textit{(SAM's accuracy)} \qquad (1)$$

The maximum probability for each agent in a fresh encounter is 5%. This probability is lower in all other cases determined by whether the agent is attacking, and the number of times the plane has encountered the SAM. In the case of Team-Attack and Team-Non-Attack roles, since team members inform one another about SAMs they encountered, fewer fresh encounters occur. The Hit Probability of each agent is accumulated when it encountered more SAMs until it reaches the target.

In our testbed we set up a timer, called *cycle*, to measure the running time for each agent from taking off from the base until reaching the target. It takes a longer time for the agent to attack SAMs than to avoid SAMs. The extra time is for communications, or helping other agents.

All of the agents in the testbed in each run must choose only one of those 4 roles and maintain it until completion. We measured the number of steps for planes to fly from beginning to end of each mission with simulation of cycles of agent's top level loop. As the agents approach SAM sites they experience probabilities of being shot down. We summed and averaged these probabilities per mission, per agent, per SAM. This average probability is termed "Average Hit Probability". Since for each run the position of SAMs are not fixed, the number of SAMs that the agents will encounter in each run is not fixed.

3.2 Result Analysis

In Figure 5, each position represents one run of the testbed, where the horizontal axis shows the time in cycles for this run, and the vertical axis shows the Average Hit Probability of the agents in a single run. No plane is considered shot down in the run and every plane gets to the other side. However, each plane gathers cumulative measures of being destroyed as they fly near SAMs. We ran the simulation for 20 runs for each of the four top level role types for a total of 80 runs corresponding to 80 data points in Figure 5. From the figure we can see that by and large agents with Team roles have lower Hit Probability than agents with Non-Team roles. Team-Attack is the best role among the four roles for the lowest threat of being shutdown. Team-Non-Attack is the second, Non-Team-Attack is the third, and Non-Team-Non-Attack is the least effective role. In the Team roles, the Average Hit Probability is lowered roughly 5.6 times by attacking from Non-Attack (about 0.025) to Attack (about 0.0045). If the objective is to minimize Hit Probability, attack is clearly preferred.

When considering the cycles to complete each mission, on the average Non-Team-Non-Attack takes the shortest time, Team-Non-Attack is the second, Non-Team-Attack is the third, and Team-Attack takes the longest time. By not attacking, agents take a shorter time to complete the mission. If the objective is to minimize time, not attack roles are preferred. The Average Hit probability is halved with not attack from Non-Team (about 0.048) to Team roles (about 0.024). But if we had taken the possibility of communication interceptions into account, the advantage might not be as good.

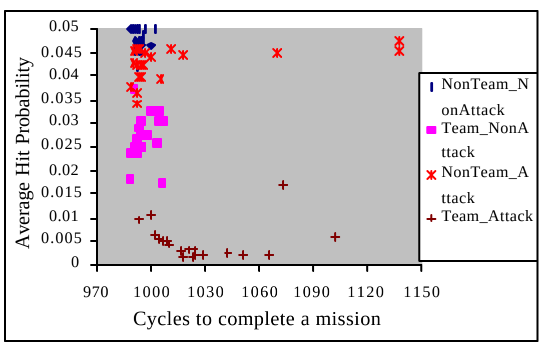

Figure 5 Mission Cycles versus Average Hit Probability

4 Related Work

Sycara's work ([4] and [6]) is closely related to our approach to teamwork, as is [5]. However, Sycara intends to provide a general level of cohesion for heterogeneous agents and a toolkit for assembling teams. She is considering large open agent groups on the Internet. Much like public services of a town, Sycara's middle agents mediate transactions between agents.

Recent developments from Milind tambe’s group include a monitor for distributed teams that minimizes the amount of required communication through having the monitor infer the progress of team plans by observing the actions of member agents [7]. Tambe’s Overseer is an example of an agent assistant. Inferring agent states from team actions suffers from uncertainty and from growth of computational complexity that is exponential in the number of agents being monitored. By modeling the group as a whole, rather than modeling the individual agents constituting the group, Overseer is able to effectively predict team responses during normal and failed plan execution in linear time. The gain is accomplished through restricting attention to coherent monitoring hypotheses. The trade off is that the team is not accurately modeled in some failure modes because an inaccurate, but efficient, temporal model is used. Knowledge bases containing the plan hierarchy, and containing the group hierarchy enable *socially attentive* monitoring. Tambe’s approach blends reasoning about teamwork and reasoning about uncertainty.

5 Conclusion and Future Work

This paper presented an empirical investigation of roles and social influences in a testbed we have implemented. Our preliminary results demonstrate that responding to team members in need of help improves the team’s overall effectiveness. This extends our earlier treatment of teams and measurement of team effectiveness along the dimensions of cohesion. We plan to use our testbed for further quantifications of team effectiveness along many other dimensions.

A particular focus of our ongoing work is the human operator and interactions of agents with the human operator. The human interpretation of social action “help” is quite varied. Human usage of “help” and reactions to it are not as deterministic as actions of agents in the simulator. Since there are also differences between people and even between two different runs we will automatically record these interactions for further analysis and richer models of “help” within a team.

Acknowledgement

We appreciate contributions of Babar Asif, Muhammad Arrabi, Ciprian Caleau, Srinivas Battula, and Dr. Gordon Beavers. This work is supported by AFOSR grant F49620-00-1-0302 and a subcontract from State University of New York at Buffalo’ Center for Multisource Information Fusion.

References

1. G. Beavers and H. Hexmoor, 2001. Teams of Agents, In the Proceedings of IEEE SMC, Arizona.
2. P. Cohen, H. Levesque, I. Smith, 1997. On Team Formation, In J. Hintika and R. Tuomela, *Contemporary Action Theory*, Synthese.
3. H. Hexmoor, and G. Beavers, 2002. Measuring Team Effectiveness, In proceedings of 20th IASTED International Multoiconfernce: Applied Informatics (AI 2002), Innsbruck, Austria.
4. T. Lenox, R. Payne, S. Hahn, M. Lewis, K. Sycara, 1999. MokSAF, How should we support teamwork in human-agent teams? *CMU-RI-TR-99-31, The Robotics Institute, Carnegie Mellon Universit.*.
5. E. Salas, C. Bowers, and E. Edens (Eds), *Improving Teamwork in Organizations* Lawrence Earlbaum Pubs, 2001.
6. K. Sycara, Brokering and Matchmaking for Co ordination of Agent Societies: A Survey. In *Coordination of Internet Agents*, A. Omicini et al. (eds.), Springer., 2001.
7. M. Tambe, D. Pynadeth, and N. Chauvat, 2000. Building Dynamic Agent Organizations in Cyberspace, *IEEE Internet Computing*, March-April issue.