Team-Oriented Agent Coordination in the RETSINA Multi-Agent System

Joseph A. Giampapa
The Robotics Institute
Carnegie Mellon University
5000 Forbes Avenue
Pittsburgh, PA 15213-3890
garof+@cs.cmu.edu

Katia P. Sycara
The Robotics Institute
Carnegie Mellon University
5000 Forbes Avenue
Pittsburgh, PA 15213-3890
katia+@cs.cmu.edu

ABSTRACT
Complex tasks are often solved by teams because no one individual has the collective expertise, information, or resources required for the effective completion or performance of a task. This paper describes a prototype, implemented in the RETSINA multi-agent infrastructure, in which agents interact with each other via capability-based and team-oriented coordination. We propose a model of team-oriented agent coordination that is based on the joint intentions theory, so that agents can communicate their intended commitments to each other. Team-oriented agents communicate partial descriptions of the context in which a mission must be executed and the resources to do so via data structures that are analogous to the SharedPlans *recipe*. The agents then proceed, in a process reminiscent of SharedPlans partial plan refinement, to refine and revise their understanding of the mission context, via both team-oriented and capability-based coordination with other RETSINA agents, while executing their mission. The partial plan refinement behavior is made possible through the RETSINA Agent Architecture, which interleaves HTN planning and process execution. We enhance the above models of teamwork by adding our own characterizations of checkpoints, role and subgoal relations in software agent teamwork, and show how the software agents can acquire this information from their operating environment during plan execution time. Such enhancements create a scalable team-oriented multi-agent system architecture, in which team coordination strategies can be implemented in a general and domain-independent way.

Categories and Subject Descriptors
I.2.11 [**Computing Methodologies**]: Artificial Intelligence Distributed Artificial Intelligence [Intelligent agents, Multiagent systems, Coherence and coordination]

General Terms
Algorithms, Design, Performance, Theory

Keywords
Teamwork, Coordinating multi-agent systems and activities

AGENTS'02, July 15-19, 2002, Bologna, Italy.

1. INTRODUCTION
Complex tasks are often solved by teams because no one individual has the collective expertise, information, or resources required for the effective completion or performance of a task. Team problem solving involves a multitude of activities such as gathering, interpreting and exchanging information, creating and identifying alternative courses of action, choosing among alternatives by becoming aware of differing and often conflicting preferences of action by team members, and ultimately implementing a choice, determining how incremental progress will be measured, and monitoring its evolution. There have been many attempts and views in the agent community as to how intelligent software agents should organize themselves into teams. The work of this paper was inspired by the *joint intentions* theory of Cohen and Levesque [3, 14], the *SharedPlans* theory proposed by Barbara Grosz [10, 11], and Milind Tambe's research based on the *TEAMCORE* [18, 27] multi-agent software system.

In the joint intentions theory, a team is composed of agents that jointly commit to the achievement of a *joint persistent goal*, or JPG. Agents desiring to be part of a team must communicate their intention to each other that they intend to commit to that goal. A team is formed once every agent has committed to a goal and has received a communication that all the other agents have committed to it, as well. The team remains a team as long as: (a) no agent has a reason to believe that the goal is unachievable; (b) no agent decides on its own to de-commit from the team goal; (c) the goal is not yet achieved and all agents remain convinced that it is still achievable; and (d) no agent perceives that another agent has de-committed from the team goal. Team goals are formed by an individual agent nominating a task as a proposed team goal, and communicating that intention until consensus is formed that the nominee is worth pursuing as a team goal. The joint intentions theory is significant because it is a formal model of what motivates agent communications about teamwork. Further, it has enjoyed broad recognition within the agent community as making pertinent claims and observations about team-oriented behaviors. But the theory does not address: (1) the problems of how agents acquire a team goal; (2) how agents identify the roles that contribute to the fulfillment of a team goal, and identify those roles for themselves; (3) how agents relate the roles of their individual goals to the overall goal of the team; and (4) how the agents know when to break commitments to their individual goals while still maintaining the team goal.

The SharedPlan theory [10, 11] emphasizes the need for a common high-level team model that allows agents to understand all requirements for plans that might achieve a team goal, even if the individuals may not know the specific details of the plans or how the requirements will be met. This allows team members to map their capabilities to a plan to achieve a team goal, assign roles to themselves, and measure their progress at achieving their overall team objective. Like joint intentions theory, SharedPlan theory is based on observations of human forms of teamwork.

TEAMCORE is an agent architecture that implements many of the basic principles of joint intention theory. TEAMCORE application scenarios are usually situated in the military and robotic soccer domains, where the team-oriented agents are homogeneous and their roles typically represent either authority relations, such as military rank, or high-level capability descriptions, such as *transport* or *escort* helicopters. But once individual TEAMCORE agents commit to being part of a team, they cannot dynamically add or subtract members to or from their team so as to adapt to a new situation while executing their plan. Further, TEAMCORE models the monitoring of team activity for performance tracking and for facilitating collaboration among the teammates as a process that is apart from the actual team-oriented communications and that can experience orders-of-magnitude increases in communications overhead to perform such monitoring [12], if sophisticated monitoring techniques are not used.

Within human-machine teams, intelligent software agents can play a variety of roles that help reduce some of the overhead of teamwork, as well as help solve team problems. By means of their autonomy, agents can: get information requested by a human; self-activate and present unsolicited important information to a human user; suggest solutions to a problem; actively monitor the environment and cache relevant information so as to provide quick updates to "situation knowledge" if required; and recombine, as needed, with other agents to adapt to the particular task requirements over time.

To effectively coordinate agents into teams in dynamic environments, our research thrust has been in line with the following principles: (1) we make open world assumptions about the nature of tasks and the strategies for solving them — a multi-agent solution to a problem will most likely involve agents of different architectures and abilities, available at different times; (2) we subscribe to the belief that there are meta rules that describe the nature of teamwork, that are independent of the specific task being performed [6, 17, 20], and that it is possible to reuse this knowledge in different application domains; and (3) that individual roles and objectives of agents within the team may need to change in order to maintain and achieve the full team goal.

In this paper we describe a prototype, implemented in our RETSINA[1] multi-agent infrastructure, in which agents interact with each other via capability-based [22] and team-oriented coordination to form and execute a team plan for the purposes of executing a team mission in a simulated world. Capability-based coordination is the condition in which agents only interoperate with other agents when they seek the use of another agent's information or services (capabilities). The contextual information that determines the plan requirements is initially acquired from humans and other intelligent software agents by a conversational case-based planner [9]. This initial plan resembles a SharedPlans *recipe*, a high-level description of the mission, and is shared among the team-oriented agents. The plan is continually updated and revised, again via capability-based and team-oriented agent coordination, in a process that is reminiscent of the SharedPlans partial plan refinement process, while the agents are determining and negotiating team roles. Once the team commits to and commences the execution of the process, the team agents continue to evaluate and refine their plans and roles as the simulated world and team agent subgoals change, throughout the execution of the mission. The continuous partial plan refinement behavior is made possible through the RETSINA Individual Agent Architecture [23, 4], which interleaves HTN planning and process execution. We enhance the above models of teamwork by adding our own characterizations of checkpoints, role and subgoal relations in software agent teamwork, and show how the software agents can acquire this information from their operating environment during execution time. Such enhancements create a scalable team-oriented multi-agent system architecture, in which team coordination strategies can be implemented in a general and domain-independent way.

In the sections that follow, we present the RETSINA model of teamwork and a command and control scenario that we used to evaluate the effectiveness of our team-oriented coordination hypotheses. After a brief review of related work, we conclude with some ideas for future work in this area.

2. RETSINA MODEL OF TEAMWORK

The RETSINA model of teamwork begins with the assumption that all agents will begin with their own copy of a partially instantiated SharedPlan model of the task. Each agent then maps their capabilities to the task fulfillment requirements, and produces a set of candidate *roles* from which it will select one at a time to propose to its teammates until all required capabilities of the task are covered. Once the agents have reached a consensus that the goal is achievable, has not been achieved, and that there are no role conflicts, then the agents commit to executing the team plan.

Team-oriented coordination in a dynamic and open environment imposes certain design requirements on individual agents and teams of agents. One of those requirements is that the information that must be communicated from one individual to another impose little overhead in terms of communications bandwidth, frequency and reliability of commu-

[1] **RETSINA** is an acronym for: **R**eusable **E**nvironment for **T**ask-**S**tructured **I**ntelligent **N**etwork **A**gents.

nication, and in the processing required for generating and understanding the messages received. We introduce the concept of a *checkpoint* as a compact data structure for effecting such efficient and possibly non-verbal communications.

2.1 Definition of Role

The RETSINA model of teamwork defines a role as a commitment by an individual, in the context of achieving an overall team goal, to: (a) the accomplishment of a task, such as meeting a teammate at a certain location and time; or (b) to the persistence of an activity until the reasons for performing that activity no longer hold, or until the individual is asked to decommit from that activity. This latter commitment is typically manifested as behaviors such as providing information to a requester at regular intervals, as would an agent that reports weather daily, or committing to provide support to a teammate should they come under attack. Roles are initiated and terminated by time, conditions, and events, both in the descriptive observation of an individual's activity, as in "supplying weather information" or "clearing an area of a minefield," and in the expectation that a teammate sets in its teammates when it commits to a role, such as promises to provide information in a timely manner, or promises to provide support should it be requested.

In our model of team coordination, a role must be based on the reasoned recognition of an individual's capability to accomplish a certain task within the time frame for which that capability is requested. For example, antipersonnel mine clearing agents must recognize that they cannot clear antitank mines, nor commit to a clearing operation if they are already engaged in one, cannot arrive to the minefield in time, or would be unable to complete the task in time given the resources at their disposition. A nice implementation consequence of an agent being able to reason about its capability is that the same advertising mechanisms can be used both for the explicit declaration of the agent's capability (e.g. "clear antipersonnel mines") for purposes of its candidate role recognition in team-oriented coordination, as for advertising its capabilities in a capability-based coordination model.

A role also must be proposed and accepted by teammates in the context of knowledge about an agent's authority and responsibility to assume a role, and with knowledge of the social structure to respect that authority. In the military, for example, a private soldier must respect the authority of a commanding officer, such as a general, and the general relies on the respect of the private soldier to obey and execute his commands. In a civilian context, however, the general no longer has authority over the soldier.

Figure 1 describes the role of a team member, m, more formally. Each g_i is a group, subgroup, or team, of which individual m is a member. Each a_i is a set of authority relations that exist among members of the corresponding group, g_i. Each o_i is an objective that m hopes to achieve by committing to the role. Note that to be consistent with joint intention theory, the overall team goal to which all members are committed, is a subset of O[2]. S represents the status of the role, and indicates: if the role's applicability to a team

[2]An overall team goal can also be a set of goals.

$$
\begin{aligned}
Role_m &= \{\ G, A, O, S, T\ \} && (1)\\
G &= \{\ g_1, \ldots, g_p\ \} && (2)\\
A &= \{\ a_1, \ldots, a_p\ \} && (3)\\
O &= \{\ o_1, \ldots, o_q\ \} && (4)\\
S &= \{\ \mathit{unevaluated} \mid \mathit{proposed} \mid && (5)\\
&\qquad \mathit{committed} \mid \mathit{decommitted}\ \} &&\\
T &= \mathit{temporal\ expression} && (6)
\end{aligned}
$$

Figure 1: The description of team member m's *role*.

plan is still *unevaluated*; if the role has been *proposed* to other individuals; if m *committed* to it upon its acceptance by the other team members; or if m *decommitted* from it, for whatever reason. T indicates the time frame in which m will assume the role or decommit from it. Additionally, there are other features that allow agents to synchronize their concept of time and coordination precision.

2.2 Requirements for Individuals

A team-oriented individual must possess three types of knowledge: his capabilities, the team plan requirements, and social parameters for role assessment, such as knowledge of his authority to address team plan requirements, and knowledge of social structure, such as superior, peer, and subordinate relationships. To be able to act on that knowledge, a team-oriented individual should also know how to perform certain types of assessments, such as: how to match his individual capabilities to plan requirements, how to evaluate if his authority allows him to apply his capabilities to the plan requirements, how to assess the impact of his and his teammates' roles on achieving the overall team goal so that he may offer more appropriate role proposals when situations change, how to monitor progress when executing a plan, and how to map social structure to plan requirements such as knowing to report to an immediate superior or that only particular team members have the authority to assign certain tasks.

The acquisition of situation-specific knowledge serves to "update" an individual's beliefs about the three types of team knowledge, mentioned above. Individuals should attempt to become aware of as much situation-specific knowledge as is necessary in the stages of forming a team, committing to roles in the team plan, and while executing the team plan. Some situation-specific knowledge may fill gaps in the partial shared plan and transform it into a full shared plan. Other situation-specific knowledge may modify the individual's knowledge of his capability, social status, or of his authority. Still, another form of situation-specific knowledge is that which is communicated by teammates at consensually-determined *checkpoints* so as to indicate individual progress in relation to the overall team plan.

2.3 Checkpoints

When humans coordinate their activities with each other, they frequently decide upon and accept a set of checkpoints as a way of verifying the progress and continuing commitment to their team goal. Often, even though the decision is not explicitly made, it is possible for a participant to reason about what may constitute an appropriate checkpoint given their model of the team's tasks, when it does become necessary to have a confirmation of the team's global commitment to the team goal. Checkpoints are often simple, observable or results-oriented phenomena, such as a phone call that someone arrived home safely, or a reply to a question in a short period of time. In the RETSINA model of teamwork, part of the team maintenance activity is explicitly dedicated to the expectation, communication, and evaluation of checkpoint information. Expectations are created when an agent communicates its commitment to a *future* role to a listening agent, at which point, the listening agent has the expectation of being able to verify by observation that the committed agent has begun its role, is in the process of executing it, or has just completed it. When a checkpoint is not communicated at the expected moment, say at the start or completion of a role, a teammate has cause for re-evaluating the participation of the missed agent. An example of this can be seen in the evaluation scenario, when an unreciprocated communication violates an agent's expectation for a reply, and is thus interpreted as a missed checkpoint. RETSINA checkpoints are modeled explicitly, as in the case of the explicit locations along a route, or implicitly, typically through timed-out replies to queries. The checkpoints are lightweight and do not model the cause of the failure. Often, understanding the cause of a failure is not warranted, and if the agent has another workaround strategy for regaining contact with its teammate, it is often more efficient to employ the workaround rather than to attempt to understand the cause of the failure.

2.4 Role Acquisition and Maintenance

As requirements of the task change, subgoals are achieved, or as individuals change their capabilities, individuals must communicate these changes to the appropriate teammates. These communications have been found to be critical in human high performance teams [6, 17, 20]. If an individual discovers that he is no longer capable of performing a role, then he must communicate this knowledge to his teammates and superiors (if appropriate) and break with his subgoal commitment. Alternatively, the individual may opt to stay with the subgoal commitment if it does not impede progress to achieving the overall team goal and there is no reason or request to assume another role.

An individual determines candidate roles for himself by matching his individual capabilities to the requirements of the overall team goal within the constraints of his authority and other social parameters.[3] If there are no candidate roles, then the individual has the following options: (1) to attempt to further refine the requirements of the overall team plan; (2) to attempt to acquire those capabilities that match the plan requirements; (3) to attempt to acquire the authority for applying the capability to the plan requirements; and (4) if it is not possible to either specify the requirements, acquire the capability or the authority to generate a candidate role in the team plan, then the individual should not commit to the team plan. If the individual were successful at generating candidate roles, however, then the individual should select the candidate roles that he feels comfortable with committing to — by whatever evaluation metric at his disposition (e.g. most commitments, least commitments, those that are the best for a certain metric, etc.) — and communicate them to his teammates.

[3]Since individuals may be committed to roles for the entire duration of the full team plan, or for less time, we often call roles, *subgoals*, as well, and use the two terms interchangeably.

An individual with candidate roles must communicate them to the other team members as proposals for his role in the team plan. Similarly, the individual must receive the proposals for roles of the other team members, and evaluate if all plan requirements are covered by all the proposals that were generated or received by the individual. If all role proposals cover all plan requirements without conflicts, then the individual may commit to the team plan and to his roles. If there are no role proposal conflicts but not all plan requirements are met, then the individual must evaluate if the requirements must be met as a precondition to executing the plan. If they are, then the individual should reconsider if he has the capability for addressing the non-assigned plan requirement, since he might have withheld proposing the role for cost reasons. If he does have the capability, and the benefit of achieving the team goal outweighs the cost of committing to that role, then the individual should propose it. If he does not have a capability to respond to the requirement, then he must wait until all other teammates have attempted to bid on it. If the requirement is eventually covered, then the team can commit to the SharedPlan. If the requirement is not covered, then the team members cannot commit to the SharedPlan, but they may actively recruit new team members that could cover the requirement.

If there are any conflicts, then only those agents with conflicting role proposals must renegotiate their role proposals in a generate-and-repropose cycle. If the conflicting parties cannot resolve their differences, they should enlarge the circle of participants to include non-conflicting individuals, in the hope that new members of the conflict resolution group may have capabilities that can permit a reassigning of proposed roles so as to avoid the conflict. Once conflicts have been resolved by the conflicted individuals, the proposed roles must be recommunicated to all team members to form a shared mental model of the team plan, so that all members can commit to it.

3. EVALUATION BY PROTOTYPE

The effectiveness of the RETSINA model of teamwork was tested by implementing a prototypical team-oriented agent system, using the RETSINA multi-agent system infrastructure, and by applying it to a simulated command and control scenario that challenged three team agents with unreliable, incomplete, and dynamically changing world knowledge, while the three team agents attempt to perform a scouting mission in a military simulator. RETSINA, the agents of the prototype, and the scenario are briefly described in the sections that follow.

3.1 The RETSINA MAS

As a multi-agent system, RETSINA agents can be described in terms of the *RETSINA Functional Architecture* [26, 23], which categorizes agents as belonging to any of four agent types:

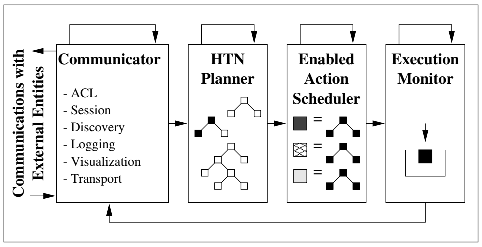

Figure 2: Schematic diagram of the RETSINA Agent Architecture. The boxes represent concurrent threads and the arrows represent control and data flow. The "external entities" may be agent or non-agent software components.

Interface agents present agent results to the user, or solicit input from the user. In addition, they could learn from user actions [2]. Interface agent behaviors can also be associated with *task agents*.

Task agents encapsulate task-specific knowledge and use that knowledge as the criterion for requesting or performing services for other agents or humans. In this respect, they are the typical agent coordinators of a multi-agent system.

Middle agents [28, 8] provide infrastructure for other agents. A typical instance of a middle agent is the *Matchmaker* [24, 25], or *Yellow Pages* agent. Requesting agents submit a *capability* request to the Matchmaker, which will then locate the appropriate service-providing agents based upon their published *capability descriptions*, known as *advertisements*.

Information agents model the information world to the agent society, and can monitor any data- or event-producing source for user-supplied conditions.

By classifying agents functionally, we believe that it is possible to uniformly define agent behaviors [4] that are consistent with their functional description.

The *RETSINA Individual Agent Architecture* [23, 4] is illustrated by Figure 2. This agent architecture implements Hierarchical Task Network (HTN) Planning [7, 15] in three parallel execution threads. A fourth thread, the *Communicator* [19], provides the means by which the agent communicates with the networked world. The coordination among the three planning modules is done in such a way that high-priority actions can interrupt those being executed by the Execution Monitor, if those being executed are of a lower priority. This coordination mechanism, which interleaves planning and execution, is the the means by which RETSINA team-oriented agents can refine their plans while executing an action.

3.2 The Agents

The following is a group of agents that had to organize themselves so as to perform in a mission of the *ModSAF* simulation environment [5][4]. There were other agents in that community, but the group that was relevant to the mission was composed of:

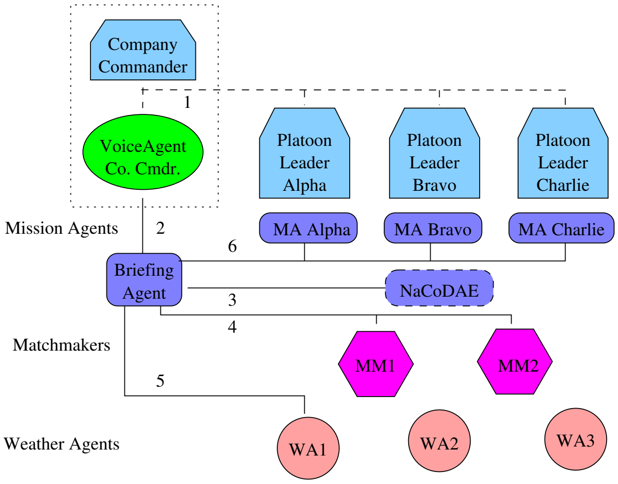

Figure 3: The agent subteam that acquires the initial shared plan from the Company Commander's briefing, and from capability-based interactions with other agents.

BriefingAgent a task agent that, together with **NaCoDAE** [1] maintains the domain-specific knowledge of the full and partial shared plans for the MissionAgents. They are explained in detail in [9].

MissionAgents team-oriented task agents that must plan their joint mission with each other. Each one monitors and commands a platoon on behalf of the human platoon commanders, in the ModSAF simulation environment.

VisualReconnaissance an information agent that scans the ModSAF map and notifies its subscribers of the location of a Threat Platoon when it finds one.

VoiceAgent a voice recognition interface agent that is based on the Sphinx [13] speech recognizer.

Figure 3 illustrates some communications that were effected among agents in order to create and to partially refine the initial team-oriented plans.

3.3 The Scenario

The scenario takes place in an area that is represented by the map of Figure 4. A human Company Commander briefs three human platoon leaders on the nature of the mission.

Threat forces are retreating southward. The tank platoons must scout the area to ensure that Threat forces have retreated. The platoon commanders may engage the enemy if necessary. There may be anti-tank mines and anti-tank ditches, but the platoons may request the services of mine clearing robots if necessary. The platoon leaders can also subscribe to reconnaissance satellite agent reports, as well.

[4]ModSAF is an acronym for **Mod**ular **S**emi-**A**utomated **F**orces.

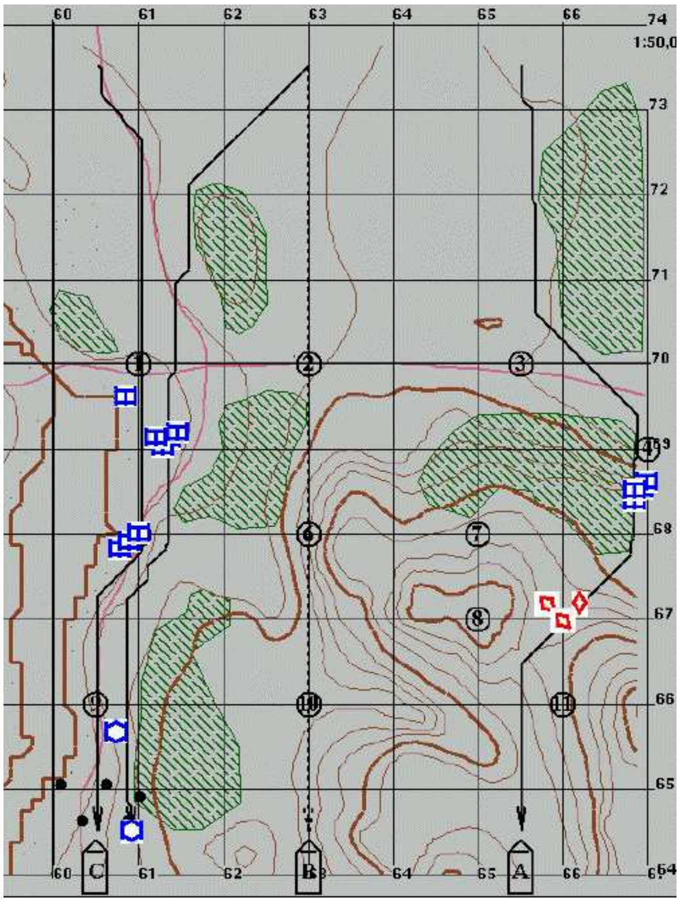

Figure 4: The Mission Scenario. Here we see Platoon Bravo providing reinforcements to Platoon Charlie on the left, instead of proceeding down its originally-planned route of passing through checkpoints 2, 6, and 10. The MissionAgents just received notice that the Threat Platoon (in red) is at checkpoint 8, which is in the path of Platoon Alpha.

```
BEGIN ACTION briefing_action                          TITLE "Briefing Object"
TEXT "(briefing_object :brief_type captains-orders
  :goal (goal :type (1 2 primary-goal)
    :1 (description :do move :what your-platoons :to horizontal_line_500)
    :2 (description :do force :what Threat_Platoon :to (behind :landmark
                                        horizontal_line_500) :mode if-found)
    :primary-goal both)
  :map-checkpoints
    (map-checkpoints :type (Xcoords-Ohio Ycoords-Ohio width-Ohio
      Xcoords-Texas Ycoords-Texas width-Texas
      Xcoords-Utah Ycoords-Utah width-Utah
      checkpoint-1    [...]   checkpoint-11     map-reference)            [...]
  :plan-requirements (requirements :type (names ohio texas utah distribution)
    :names (ohio texas utah)                  :ohio (maneuverability :rating 3-6)
    :texas (maneuverability :rating 6-8)      :utah (maneuverability :rating 3-6)
    :distribution (description :quantity 1 :assigned negotiate)          )
  :team-capability (team :type (capability-1 capability-2 capability-3)
    :capability-1 (description :maneuverability 5 :firepower 6  [...]        )
  :warning ()     )"
TOOLBOOK                                               END ACTION
```

Figure 5: Part of the *Briefing Object*, encoded as a NaCoDAE case action

As he speaks, the Commander's VoiceAgent translates his statements into text-based messages, and sends those messages to the BriefingAgent. The BriefingAgent extracts a description of the mission from the Commander's briefing, and by interacting with NaCoDAE[5], Matchmakers, and other agents via capability-based coordination, illustrated by Figure 3, forms a partial shared plan of the mission requirements, as shown in Figure 5. At the conclusion of the Commander's briefing, the BriefingAgent finds the MissionAgents by querying the Matchmakers for agents with knowledge of team coordination for the command of platoons in ModSAF. Upon receipt of a list of four MissionAgents, the BriefingAgent selects three and then assigns one platoon to each of them. All three MissionAgents receive the same description of the task that requires teamwork in the form of an HTN plan objective, with the briefing_object as the data segment. The goals are: (a) to scout the terrain up to landmark `horizontal_line_500`; (b) to force `Threat_Platoon`, if encountered, behind the landmark; and (c) the team goals are conjunctive: `:primary-goal both`. Other information that the MissionAgents receive in the partial shared plan of the BriefingObject are: descriptions of three rough map corridors that the agents should scout, each corridor with its own maneuverability ratings; a distribution requirement of one MissionAgent per corridor, as determined by the Commander; a list of eleven map checkpoints that are shown on the map of Figure 4; reminders for the type of information to monitor during the execution of the plan; and capability requirements for the platoons that should perform the mission.

A MissionAgent's capabilities are those of the platoon that it represents. Since a platoon's composition is provided by the Company Commander's briefing, the team members initially learn of their capabilities via the briefing_object, for example, `:team-capability-1 (description :maneuverability 5 :firepower 6 ...)`. During the execution of the mission, if any component of a MissionAgent's platoon suffers damage, or needs to share resources, then the MissionAgent will perceive the change to its own capabilities and communicate that knowledge to the other team members, and to the supervising commanding officer via the BriefingAgent. For example, it frequently happens in the ModSAF simulation that one of MissionAgent Charlie's tanks gets stuck in a trap and breaks a track[6]. When that occurs, Charlie Platoon's maneuverability and firepower ratings are decreased as the platoon loses one of its members.

The plan requirements are represented in the briefing_object as the three corridors of Figure 5, `ohio texas utah`, respectively, the *left*, *center*, and *right* corridors of the map. There is also a `distribution` requirement that one platoon should patrol one corridor, but leaves it up to the MissionAgents to `negotiate` their assignment to a corridor, the only role that they can assume at this point of the mission. Each corridor has its own `maneuverability :rating` requirement, which the MissionAgents match to their own `maneuverability :rating` capabilities, to generate candi-

[5]The details of how NaCoDAE forms the partial shared plan by interacting with the BriefingAgent and other agents are described in [9].

[6]This is shown by the tank closest to checkpoint 1 in Figure 4.

date roles. Any one of the MissionAgents could possibly patrol one of the corridors, but only MissionAgent Bravo has the superior maneuverability that matches the slightly higher requirements of the central corridor, so he proposes that as his first candidate role, and has his proposal accepted by the others. MissionAgents Alpha and Charlie at first propose to patrol the same corridor, which they both recognize as a violation of the preferred distribution requirement as soon as they communicate their intentions to each other, because they are both reasoning from the same partial shared plan. Without needing to communicate the discovery of the mismatch, Alpha and Charlie each randomly choose a role to propose from their complete set of best candidate roles, to determine who will take the other corridor that has the same characteristics, until the distribution requirement is met.

At this point, each MissionAgent has a role that seems to satisfy the requirements of the mission. They perceive the mission as doable since there are three MissionAgents to divide the task of scouting the three corridors, there are no role conflicts, and each team member is aware of the commitments of the other team members to achieve the overall team goal. But just as they are about to propose their roles to their respective human commanders, for approval, the MissionAgents receive an intelligence report that a Threat Platoon was sighted near checkpoint 1, right in MissionAgent Charlie's path. MissionAgent Charlie then reconsiders his maneuverability and firepower capabilities: his platoon might not be stronger than the Threat Platoon in the event of a possible engagement, so he requests a reinforcement from the other two MissionAgents. Both MissionAgents Alpha and Bravo examine their capabilities, and the capabilities of each other, in consideration of helping Charlie. MissionAgent Alpha could help, but decides not to offer assistance, based on its recognition of MissionAgent Bravo's superior capabilities, which are part of MissionAgent Alpha's partial shared plan, as well. Indeed, MissionAgent Bravo offers to modify its route to reinforce MissionAgent Charlie's, and its proposal is accepted by all teammates. The mission-specific distribution requirement is relaxed in deference to the more general team-oriented requirement of providing support to requesting teammates. This reassignment of MissionAgent Bravo's role illustrates how an agent can maintain commitment to the overall team goal — scout the area and escort Threat forces out of the area — while de-committing from a temporary or individual subgoal, e.g. patrolling the center corridor. Had MissionAgent Charlie not requested Bravo's assistance, MissionAgent Bravo would have committed in his originally proposed role of scouting the central corridor.

The MissionAgents agree that they will use the checkpoints of their partial shared plan, which is derived from the checkpoints on the map, to monitor each other's progress while executing the mission, and commit to making an announcement every time that they pass one of the map checkpoints. The MissionAgents present their proposed plans to the human commanders via the MokSAF[16] agents. The human commanders approve them, and the MissionAgents commit to the plans and begin executing the mission.

If nothing more occurs, the MissionAgents complete their mission, scouting the area down to the southern landmark. Each MissionAgent announces a checkpoint as it passes one so that the other MissionAgents can update their mental models of its actual geographical location. An anti-tank minefield is discovered by MissionAgent Charlie's scout. This event triggers an expansion of the description of the shared plan capability requirement for MissionAgent Charlie, namely, to be able to clear a path through the field. Since MissionAgent Charlie does not have the capability to clear a minefield, he searches for a robotic mine clearing team[21] via the Matchmakers, finds an unassigned team, enlists their services, and thereby extends his capability through capability-based coordination. If the mine clearing is successful, then Platoon Charlie can pass through the minefield. If the minefield remains uncleared, Platoon Charlie either sits and waits, or reroutes to another destination, as determined by a random decision generator.

Halfway through the ModSAF scenario, if a human user places a Threat Platoon at checkpoint 8 (c.f. Figure 4), its presence is detected and reported to the MissionAgents via the VisualReconnaissance agent. This prompts the MissionAgents to query each other to find out if they should alter their individual roles. MissionAgent Alpha, navigating the right corridor in Figure 4, does not respond to the queries, which is interpreted by default as a missed checkpoint by the other agents. MissionAgents Charlie and Bravo have not encountered the Threat Platoon, as previously anticipated, and the risk that the Threat Platoon might be in ambush of Alpha Platoon, prompts them to consider breaking from their roles. Since MissionAgent Bravo has the superior firepower and maneuverability, he will ask MissionAgent Charlie if he can break from his support-providing role in order to assist MissionAgent Alpha. MissionAgent Charlie agrees, and MissionAgent Bravo replans his route from where he is to checkpoint 8, and begins executing that new role. Again, this illustrates how an individual can remain committed to the overall goal of a team while changing their own role in the team.

4. CONCLUSIONS

In the work described by this paper we have tested and evaluated joint intention theory, tested and evaluated SharedPlans, and shown how the two can be combined and implemented by a multi-agent system. We also contributed some extensions that have not been directly addressed by the team-oriented agent coordination literature. Namely, we have: (1) characterized the notion of role to include notions of social order and authority, that permit an agent to propose a role for itself and to recognize the validity of a role proposed by its teammates; (2) explicitly recognized checkpoints as a vital and lightweight mechanism by which teammates can communicate and verify commitment to a goal, and propose them as a necessary activity of team-oriented agent coordination; (3) described a model of teamwork that allows team-oriented agents to acquire and determine subgoals for themselves; (4) described a model of teamwork that permits agents to decommit from subgoals and roles, yet still maintain the overall mission goal that motivates the team. From the perspective of team-oriented agent research, we have provided principles for developing and supporting agent teams, and tested them by applying these principles to a scenario that involved software agent teams operating in a simulated environment. Through such tests, we contribute to the understanding of agent roles and human-agent interactions in teams composed of humans and intelligent software agents.

5. ACKNOWLEDGMENTS

The authors would like to acknowledge the contributions of past and present members of the Intelligent Software Agents Laboratory at CMU. Without their ideas and enthusiasm, this research would not have been possible. This research has been sponsored in part by the Office of Naval Research Grant N-00014-96-16-1-1222, DARPA Grant F-30602-98-2-0138, and by AFOSR Grant F49620-01-1-0542.

6. REFERENCES

[1] D. W. Aha, L. A. Breslow, and T. Maney. Supporting conversational case-based reasoning in an integrated reasoning framework. *Case-Based Reasoning Integrations: Papers from the 1998 Workshop*, 1998.

[2] L. Chen and K. Sycara. Webmate: A personal agent for browsing and searching. In *Proceedings of Agents 1998*, May 1998.

[3] P. R. Cohen and H. J. Levesque. Teamwork. *Noûs*, 25(4):487–512, 1991.

[4] K. Decker, A. Pannu, K. Sycara, and M. Williamson. Designing behaviors for information agents. In *Proceedings of Agents 1997*, February 1997.

[5] DMSTTIAC Defense Modeling, Simulation, and Tactical Technology Information Analysis Center, Orlando, Florida 32826. *ModSAF Kit C*, ModSAF 5.0 edition. http://www.modsaf.org.

[6] D. J. Dwyer, J. E. Fowlkes, R. L. Oser, E. Salas, and N. E. Lane. *Team Performance Assessment and Measurement: Theory, Methods and Applications.* Erlbaum, 1997.

[7] K. Erol, J. Hendler, and D. S. Nau. HTN planning: Complexity and expressivity. In *Proceedings of AAAI-94*, 1994.

[8] J. A. Giampapa, M. Paolucci, and K. Sycara. Agent interoperation across multagent system boundaries. In *Proceedings of Agents 2000*, June 2000.

[9] J. A. Giampapa and K. Sycara. Conversational case-based planning for agent team coordination. In D. W. Aha and I. Watson, editors, *Proceedings of the Fourth International Conference on Case-Based Reasoning, ICCBR 2001*, volume 2080, pages 189–203, 30 July – 2 August 2001.

[10] B. Grosz and S. Kraus. Collaborative plans for complex group action. *Artificial Intelligence*, 86(2):269–357, 1996.

[11] B. Grosz and S. Kraus. The evolution of sharedplans. In *Foundations of Rational Agencies*, pages 227–262. Kluwer Academic Publishers, March 1999.

[12] G. A. Kaminka, D. V. Pynadath, and M. Tambe. Monitoring deployed agent teams. In *Proceedings of Agents 2001*, pages 308–315, 28 May – 1 June 2001.

[13] K.-F. Lee, H.-W. Hon, and R. Reddy. An overview of the Sphinx recognition system. In *IEEE Transactions on Acoustics, Speech and Signal Processing ASSP-38*, volume 1, pages 35–45, January 1990.

[14] H. J. Levesque, P. R. Cohen, and J. H. T. Nunes. On acting together. Technical report, AI Center, SRI International, 1990.

[15] M. Paolucci, O. Shehory, and K. Sycara. Interleaving planning and execution in a multiagent team planning environment. Technical report, CMU, 2000.

[16] T. Payne, K. Sycara, M. Lewis, T. L. Lenox, and S. K. Hahn. Varying the user interaction within multi-agent systems. In *Proceedings of Agents 2000*, pages 412–418, June 2000.

[17] C. Prince, D. P. Baker, L. Shrestha, and E. Salas. Situation awareness in team performance. *Human Factors*, 37:123–136, 1995.

[18] D. V. Pynadath, M. Tambe, N. Chauvat, and L. Cavedon. Toward team-oriented programming. In *Intelligent Agents VI: ATAL, N.R.*, pages 233–247, 1999.

[19] O. Shehory and K. Sycara. The retsina communicator. In *Proceedings of Agents 2000*, 2000.

[20] R. J. Stout and E. Salas. The role of planning in coordination team decision making: Implications for training. In *Proceedings of the Human Factor and Ergonomics Society* 37^{th} *Annual Meeting*, pages 1238–1242, 1993.

[21] G. Sukthankar. Team-aware multirobot strategy for cooperative path clearing. In *AAAI-2000*, July 2000.

[22] K. Sycara. Multiagent systems. *AI Magazine*, 19(2):79–92, Summer 1998.

[23] K. Sycara, K. Decker, A. Pannu, M. Williamson, and D. Zeng. Distributed intelligent agents. *IEEE Expert, Intelligent Systems and their Applications*, 11(6):36–45, 1996.

[24] K. Sycara, K. Decker, and M. Williamson. Middle-agents for the internet. In *Proceedings of IJCAI-97*, 23-29 August 1997.

[25] K. Sycara, M. Klusch, S. Widoff, and J. Lu. Dynamic service matchmaking among agents in open information environments. *Journal ACM SIGMOD Record*, 28(1):47–53, March 1999.

[26] K. Sycara and D. Zeng. Coordination of multiple intelligent software agents. *International Journal of Intelligent and Cooperative Information Systems*, 5(2 and 3):181–211, 1996.

[27] M. Tambe, D. V. Pynadath, N. Chauvat, A. Das, and G. A. Kaminka. Adaptive agent architectures for heterogeneous team members. In *Proceedings of ICMAS-2000*, pages 301–308, 10–12 July 2000.

[28] H.-C. Wong and K. Sycara. A taxonomy of middle-agents for the internet. In *Proceedings of ICMAS 2000*, pages 465–466, 10–12 July 2000.