Executing Team Plans in Dynamic, Multi-agent Domains

Milind Tambe
Information Sciences Institute
University of Southern California
4676 Admiralty Way, Marina del Rey, CA 90292
tambe@isi.edu
http://www.isi.edu/soar/tambe

Abstract

This paper focuses on flexible teamwork in dynamic and real-world multi-agent domains. Such teamwork is not simply a union of agents' simultaneous execution of individual plans, even if such execution pre-coordinated. Indeed, uncertainties in complex, dynamic domains often obstruct pre-planned coordination, with a resultant breakdown in teamwork. The central hypothesis in this paper is that for durable teamwork, agents should be provided explicit *team plans*, which directly express a team's joint activities. When agents execute such team plans, they abide by certain "commonsense" conventions of teamwork. Essentially, such conventions provide a deeper model of teamwork, facilitating flexible reasoning about coordination activities. Such a framework also frees the planner or the knowledge engineer from specifying very detailed low-level coordination plans. This framework has been implemented in the context of a real-world synthetic environment for helicopter-combat simulation.[1]

1 Introduction

Many AI researchers are today striving to build agents for complex, dynamic multi-agent domains. Such domains include virtual theatre(Hayes-Roth, Brownston, & Gen 1995), realistic virtual training environments (e.g., for emergency drill(Pimentel & Teixeira 1994) or combat(Tambe *et al.* 1995)), virtual interactive fiction(Bates, Loyall, & Reilly 1992) and RoboCup robotic and virtual soccer(Kitano *et al.* 1995).

This paper focuses on plan execution in such dynamic, multi-agent domains. While this topic is well-investigated in the literature, most research has focused on individuals rather than agent teams. In the case of individuals, the difficulty in traditional plan-execution — i.e., execution of a rigid precomputed list of actions — is now well-recognized. Instead, individual agents in dynamic environments are often based on *hierarchical reactive plans*(Firby 1987). For instance, agents built in PRS(Ingrand *et al.* 1992), RAP(Firby 1987), BB1(Hayes-Roth, Brownston, & Gen 1995), Soar(Newell 1990) and other architectures in dynamic domains may be characterized in this fashion. Reactive plans are qualified by preconditions, which help select plans for execution based on the agent's current high-level goals/tasks and beliefs about its environment. Selecting high-level abstract plans for execution leads to subgoals. Hierarchical reactive-plan expansion now ensues, which bottoms out in primitive skills. Activated reactive plans in the plan-hierarchy terminate via terminating conditions; which automatically terminates any active subgoals. In effect, these reactive plans provide a deeper domain model that enables agents to generate appropriately flexible and reactive behaviors. Although this abstract charachterization fails to capture the full richness of the aforementioned architectures — e.g., selection mechanisms for arbitrating among competing plans — it captures a reasonable enough aspect of plan execution to form the basis for the discussions in this paper.

This paper aims to extend the hierarchical reactive plan approach to encompass agent teams. All around in our daily lives, we participate in , interact with or observe team activities, such as, coauthored papers, team sports, plays (theatre), orchestras, political campaigns and military exercises. These team activities are reflected in many of the multi-agent domains discussed above. Such team activities are not merely a union of simultaneous, coordinated individual activities in service of their individual goals(Grosz & Sidner 1990; Cohen & Levesque 1991). For instance, ordinary automobile traffic is not considered teamwork, despite the simultaneous activity, coordinated by traffic signs(Cohen & Levesque 1991). On the contrary, driving in a convoy, although sometimes uncoordinated, would be considered teamwork. Such teamwork cannot necessarily be decomposed as coordinated individual activities. Consider the example of two children collaboratively building a single tower of blocks — they are not coordinating to build two separate towers of blocks with gaps in just the right places(Grosz & Sidner 1990). In soccer, two teammates together can execute a *wall pass* to dodge an opponent; however, no individual can execute a wall pass in isolation.

While such examples are illustrative of teamwork, its precise nature is very much a topic of active debate(Cohen & Levesque 1991; Grosz & Sidner 1990; Searle 1990; Kinny *et al.* 1992; Jennings 1995). Indeed, in much work

---

[1] This research was supported as part of contract N66001-95-C-6013 from ARPA/ISO. Domain expertise was provided by Dave Sullivan of BMH Inc. Thanks to Wei-Min Shen for valuable comments on an earlier draft of this paper.

on dynamic, multi-agent domains, a strong distinction between teamwork and ordinary coordinated activity does not necessarily exist. Individual agents are often provided individual plans to achieve individual goals, with detailed precomputed plans for coordination and communication; team activities are either not represented explicitly, or rely on shallow rules(Jennings 1995). However, an analogy appears to hold: just as detailed precomputed plans were deemed inadequate for an individual in dynamic domains, rigid precomputed coordination actions appear inadequate to enable flexible teamwork in dynamic environments. In particular, in real-world dynamic environment (such as our domain described in Section 2), unanticipated events often disrupt preplanned coordination, diminishing a team's effectiveness in jointly accomplishing its goals.

This paper focuses on enabling a team to jointly achieve its goal, and do so effectively (effectiveness maybe measured in terms of time, risk, rewards etc.). The central hypothesis in this paper is that for durable and effective teamwork in complex, dynamic environments, individual team members should be provided explicit *team plans*, that directly express a team's joint activities. Thus, an individual agent directly executes *reactive team plans*, which may hierarchically expand out into reactive plans for its role in the team. These team plans are based on the *joint intentions* framework(Cohen & Levesque 1991), which forms the basis of certain *commonsense* conventions for teamwork — although these conventions have been modified to accommodate the constraints that appear typical in some real-world dynamic domains.[2] When agents execute such team plans, they abide by these teamwork conventions. Essentially, these conventions provide agents with a deeper model of teamwork, so they can flexibly reason about coordination and communication actions. This flexible reasoning is especially important to bind the team together when unanticipated events occur.

In the remainder of this paper, we first introduce the domain of our work — we believe that given its real-world nature, it reflects many of the key characteristics of other real-world, dynamic multi-agent environments. We also describe our initial experiences in designing agent teams for this domain. These initial experiences are illuminating, as they motivate the need for team plans. The joint intentions framework and modifications are introduced next; followed by a detailed description of our implementation of this framework using the Soar architecture(Newell 1990; Rosenbloom *et al.* 1991). We assume some familiarity with Soar's problem-solving model, which involves applying an operator hierarchy to states to reach a desired state. As mentioned above, such an operator hierarchy can be seen as a hierarchical reactive plan.

2 Domain and Initial Experiences

The domain of our work is a real-world battlefield simulator, commercially developed for the military for training(Tambe *et al.* 1995). We are building intelligent pilot agents for synthetic aircraft in this environment. These pilot agents have participated in large scale combat exercises, some involving expert human pilots. This paper will focus on pilot agents for a company of (up to eight) attack helicopters, which execute their mission in a synthetic 3D terrain, complete with hills, valleys, roads and ridges(Tambe, Schwamb, & Rosenbloom 1995).

As shown in Figure 1, in a typical attack mission, the company may fly 15-20 kilometers or more in various formations, to halt at a holding point. One or two helicopters in the company are designated as scouts, and they fly forward to check the battle position, i.e., the location from where the company will attack enemy forces. Once the battle position is scouted, other members of the company move into the battle position, each hovering in its own designated subarea of the battle position. Here, an individual pilot agent hides/masks its helicopter. To attack, the pilot has his helicopter "popup" (rise high), to shoot missiles at enemy targets. The helicopter then quickly masks, as protection again return fire. It may then shift its location (so its next popup location is a surprise). When the mission completes, the helicopters regroup and return to base.

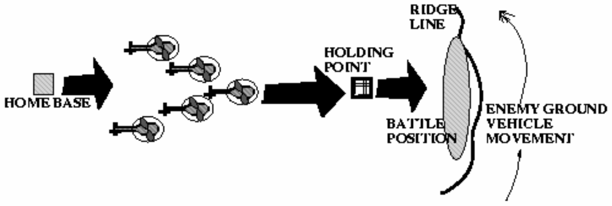

Figure 1: A simulated combat scenario involving a company(team) of helicopters. The ridge line provides an ideal masking/hiding location.

This domain possesses many of the characteristics outlined in the call for papers for this symposium, given that it is a real-world synthetic domain, with multiple collaborating and competing agents:

- *Complexity:* Pilot agents do not possess perfect information about the world, e.g., they are unaware of types and locations of enemy vehicles.
- *Dynamism:* Multiple friendly or enemy agents can independently execute actions in this world, e.g., enemy vehicles may move in and out of sensors and weapons range.
- *Uncertainty:* A real-time simulation with realistic vehicles ensures that even identical start states lead to very different scenario configurations and outcomes. For instance, helicopters may unexpectedly crash or get shot down.
- *Interruptibility:* Some actions last appreciable durations, e.g., flying from and to the battle position may require 30-40 minutes. These operators can be interrupted by events (such as appearance of enemy vehicles).

---

[2] Commonsense is to be understood here as the "common knowledge about the (social) world that is possessed by every schoolchild and the methods for making obvious inferences from this knowledge"(Davis 1990).

- *Goal variability:* While operators such as flying in formation are mainly maintenance goals, operators such as one that identify enemy vehicles are achievement goals.
- *Changing objectives:* Agents objectives changes depending on their location, their sensor information, and the state of their vehicle.

Our first implementation of the pilot agents for helicopters focused on enabling individuals to cope with the complexities of this dynamic domain(Tambe, Schwamb, & Rosenbloom 1995). Figure 2 illustrates a portion of the operator hierarchy for an individual (at any one time, only one path in this hierarchy from the root to a leaf node is active). Each operator consists of (i) precondition rules, to help select the operator; (ii) application rules to apply the operator once selected (a complex operator may subgoal, rather than being directly implemented); (ii) termination rules, to terminate an operator if achieved or unachievable.

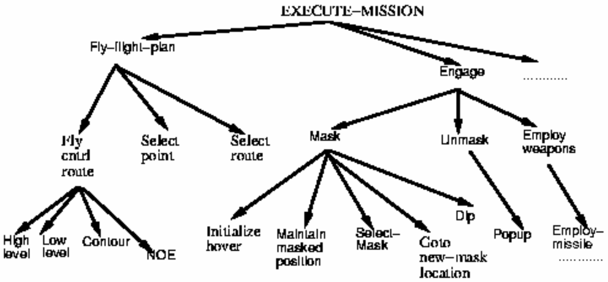

Figure 2: A portion of the operator hierarchy for an individual helicopter pilot agent.

Given the success of individual helicopter pilots, the next step was to enable a company of helicopters to jointly execute a mission. To this end, three techniques were used. First, each agent was provided a list of agents participating in the mission, e.g., pilot agents cheetah41, cheetah42, cheetah43, cheetah44 and cheetah45 are all participants in the mission. Second, specific coordination plans were added. For instance, after flying forward from the holding area, the scout executes a plan to fly back to the holding area and informs those waiting at the holding area about the battle position being scouted; individuals in the company then move to the battle position.

Third, to enable agents to fly closely in formation, each agent had listed with it a "partner" agent. When flying in formation, unless the agent was the leader of the formation, it executed an operator to follow the actions of its partner agent (e.g., trail the partner at a distance of 100 meters). Large formations were built in this fashion via appropriate choice of "partner" agents (partner relationships are changable and asymmetric). When not flying in formation, an agent did not follow the actions of its partner. Eventually, all coordination within a group could be accomplished by each agent only coordinating with its partner.

Thus, this was a reasonable and serious effort to enable a company of pilot agents to coordinate their mission execution — it is not much different from previous such efforts, at least in the synthetic battlefield domain(Tambe *et al.* 1995). The overall system contained more than 1000 rules, and it was tested in October 1995 in a large-scale exercise lasting three days (with a maximum of 400 agents in the synthetic battlefield).[3] The helicopter company missions were predesigned, and tested prior to the exercise; although given the very large number of behavioral variations and agent interactions in the synthetic battlefield, the company faced many unanticipated events. While in general the helicopter company performed adequately, the tests and the exercise revealed some key weaknesses in teamwork. A few illustrative examples of such failures are described in Figure 3.

1. Upon reaching the holding area, the company waited, while the scout started flying forward. Unfortunately, during the exercise the scout unexpectedly crashed into the terrain. Hence, the rest of the company just waited indefinitely at the holding area, waiting to receive a message from the (crashed) scout that the battle position was scouted.
2. Upon recognizing that the mission was completed, one company member (the commander) returned to home base, abandoning others in its company at the battle position. The commander's "partner" agent was unexpectedly shot down, and hence it failed to coordinated with other members in its company.
3. In attacking the targets from the battle position, only one member of the company could see targets, and the others could not. Thus, only one member engaged the targets; the others returned without firing a single shot.
4. Some company members failed to recognize that they had reached a waypoint — the agent leading the formation had reached the waypoint, but those trailing in formation concluded they had not individually done so (despite tolerance ranges in measuring distances).

Figure 3: Illustrative examples of a breakdown in teamwork.

The planner or the knowledge engineer could of course add specific coordination plans to address problem such as the ones listed here. However, it is difficult to anticipate such failures and build specific recovery actions; and such a strategy will be unlikely to scale up well with increasingly complex and unrehearsed scenarios. Instead, the approach pursued in this work is to focus on what appears to be the root of such teamwork breakdown — non-explicit teamwork. Agents do not recognize that they are participating in team activities, i.e., jointly executing team plans. Thus, they are unable to reason about their own or other agents' roles and responsibilities in the team plan, or the underlying justifications for communicative acts. As a result, their coordination breaks down when unanticipated events occur — as in the scout's crashing on the way to the battle position (Item 1, Figure 3). Yet, these failures are not in the agents' domain-level plans (i.e., helicopter tactics). Even as non-experts in this domain, we can recognize these teamwork failures, and if in that situation, would take at least some corrective actions. This is possible, it seems, due to our

---

[3] This demonstration was jointly done with Paul Rosenblooom and Karl Schwamb.

commonsense understanding of team activities, which the agents lack.

3 Explicit Teamwork

At the core of our model of teamwork is the *joint intentions* framework(Cohen & Levesque 1991; Levesque, Cohen, & Nunes 1990). There are two main aspects to this framework. The first aspect defines the explicitly joint nature of team activity — a team jointly intends a team activity if it is jointly committed to completing that activity, mutually believing that there were about to do it.[4] Joint commitment implies that (at least initially) team members have a mutual belief that the team activity is not achieved, and a mutual belief that they have each adopted the team activity as their goal. This aspect of the framework enables an individual to treat its team as a single entity that commits to a joint or team activity. As a simple example, when a company of helicopters flies to a waypoint, it is a team jointly committed to a team activity — each individual is not flying on its own to that waypoint, while merely coordinating with others. Thus, each individual need not individually reach the waypoint; it is sufficient if the team reaches the waypoint[5]. In effect, there is a change in the level of expressiveness of plans, alleviating concerns such as the fourth item raised in Figure 3.

The second aspect of the framework is a convention to ensure that team members cannot freely disengage from a joint commitment at will. Specifically, the establishment of a joint intention imposes a certain responsibility for a team member when it privately comes to believe that the team's jointly intended activity is either (i) achieved; (ii) unachievable or (iii) irrelevant. In such cases, the team member must have this private belief become mutual belief(Cohen & Levesque 1991); and the joint venture is now dissolved. Establishing such mutual beliefs often requires an agent to create a communicative goal.

This communication convention is essentially a piece of "commonsense", providing a *deeper justification* for inter-agent communication within a team and enabling flexible reasoning about agents' communicative goals (as seen below). Agents abiding by this convention enjoy several benefits. First, by informing others that the current task is achieved, unachievable or irrelevant, an individual member ensures that the others will not waste their time, or expose themselves to unnecessary risks. This addresses difficulties such the second example in Figure 3, where an individual disengaged from the joint commitment without informing other members of the team, and exposed them to unnecessary risks. Second, if team members adhere to this convention, each agent can execute its part of the team plan, assured that it will not be abandoned.

The following subsections motivate and present several modifications to this base framework to accommodate the constraints of complex, dynamic and uncertain domains.

3.1 Communication Responsibilities

In many environments, such as synthetic battlefields or soccer fields, communication can be costly, risky or otherwise problematic, making it difficult to abide by the communication convention in the joint intentions framework. For instance, in battlefield simulations, communication may break radio silence, exposing a team to unnecessary risks. Even if communication costs are not very high, continuous message broadcasts can be highly redundant and burdensome. There are three possible approaches to address this difficulty. First, an agent must not immediately act on a communicative goal. Instead, *it must first balance the costs and benefits of communication*. If unfortunately the costs do outweigh benefits, the agent must attempt to reduce communication costs. For instance, if communication over the radio is high risk, the agent may travel to personally deliver the message. Or, the agent may wait until a reduction in communication costs to convey the information.

Second, an agent may rely on its teammates' visual (or other) sensors, rather than communication, to update the team's mutual beliefs. For instance, if a company of helicopters reaches a pre-specified waypoint, individuals can be trusted to recognize it, and it is not necessary to broadcast messages back and forth. Analogously, an agent may rely on its teammates' *agent tracking*(Tambe & Rosenbloom 1995; Tambe 1996) capability — inference of higher level goals and behaviors based on observations of actions — to reduce its communicative burden. In particular, an agent can assume that teammates are able to interpret its actions, and thus the need for communication may be obviated.

3.2 Complex Teams, Roles and Failures

In the joint intentions framework, establishment of a joint intention leads individuals or subteams in the team to intend to do their share of a team activity (subject to the joint intention remaining valid). Kinny et. al.(Kinny *et al.* 1992) elaborate the issue of a share in a team activity, via their notion of a *role*. A role defines an activity that a subteam, possibly just an individual, undertakes in service of the team operator. A role may be in turn based on an individual's capability. In the case of a company of helicopters, a specific individual may be the commander (capability depends on the chain of command), a scout (capability depends on training), or the leader of a formation (everyone in the company possess this capability).

In simpler, short-term team activities, involving two-three individuals (e.g., lifting a heavy table), an individual's success or failure in its role is directly linked to the success or failure of the joint activity. However, as we scale up to complex teams, with at least five-eight individuals, and to team activities that last hours, the relationship between an individual's own role performance and the team's joint activity becomes more complex as well. Individuals may succeed or fail in their role, or even withdraw from the team's joint intention, but that may not automatically

---

[4] These joint commitments have a common escape clause, which we will not address here.

[5] This may mean that the leader or some pre-specified percentage of vehicles reach close to the waypoint

dissolve the team's joint venture. For instance, if an individual helicopter is shot down during an engagement, the helicopter company should not automatically dissolve its joint intention to engage the enemy.

These considerations lead to the following modifications in the teamwork conventions. Agents should communicate their role success or failures to others in the team (should other individuals be banking on this role performance). Since agents may be unable to communicate role non-performance (e.g., if a helicopter simply crashes or is shot down), team members must monitor other agents' role performance (at least to the extent it is computationally feasible). Based on such monitoring, team members should determine the impact of others' role (non-)performance on the team's joint intention or on their own role. Two heuristics may be used:

1. *Critical expertise heuristic*: If the success of the team's joint intention is solely dependent on the role of an individual agent, then the agent's role non-performance (failure) implies that the team's joint intention is unachievable; while successful role performance implies that the team's joint intention is achieved.
2. *Dependency heuristic*: If an agent's own role performance is dependent on the role of the non-performing agent, then the agent's own role performance is unachievable.

4 Implementing the Modified Joint Intentions Framework

To implement the modified joint intentions framework the concept of *team operators* has been defined. For the team Θ, a team operator OP will henceforth be denoted as $\boxed{\text{OP}}_{\Theta}$. As with the usual individual operators (see Figure 2), team operators also consist of: (i) precondition rules for selection; (ii) application rules for application (complex team operators will lead to subgoals); and (iii) termination rules. However, unlike individual operators, team operators encode the expressiveness and associated conventions of joint intentions, as discussed below.

4.1 Team Operators: Expressiveness

Team operators express a team's joint activity rather than an individual's own activity. Thus, while individual operators apply to an agent's own state, a team operator applies to a "team state". The team state is an agent's (abstract) model of the team's mutual beliefs about the world, which include identities of members in the team, specification of the team's overarching goal etc. For a helicopter company, such shared beliefs may additionally include the detailed team mission information, such as the routes to fly to the battle position. The altered operator hierarchy of helicopter pilot agents with team operators is shown in Figure 4, where operators such as $\boxed{\text{Engage}}_{\Theta}$ are team operators. The children of such operators can be individual operators or team operators. The children of $\boxed{\text{Engage}}_{\Theta}$ are all individual operators to *mask* the helicopter, *unmask* it, and *employ-weapons*. The team operators are not tied to any specific number of agents within a team.

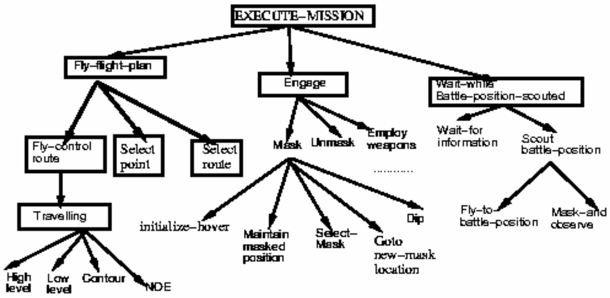

Figure 4: A portion of the modified operator hierarchy, executed by individual helicopter pilot agents. Team operators are shown in boxes; the rest are individual operators.

It is important to note, though, that team state and team operators are only an individual's model of the on-going team activities, team members do not physically share memory. In fact, in (Tambe 1996), we have used these team operators for tracking other team's activities, i.e., inferring the joint goals and intentions of those teams, and tracking their progress. As in that work, when an individual agent applies a team operator to a team state, it tracks or marks the progress of the team's activities.

4.2 Team operator: Conventions

Once a team member enters a joint commitment, by previously introduced convention, it must not decommit without updating the team state (mutual beliefs). This is accomodated in that a team operator may only be terminated by updating the team state to satisfy the team operator's termination rules.

This convention provides an example of how communication is not pre-planned, but arises in this framework from deeper reasoning. Thus, if an agent's private state contains a belief that makes a team operator achieved, unachievable or irrelevant, and such a belief is absent in the team state, then it automatically creates a communicative goal, i.e., a communication operator. When implemented, this operator leads the agent to broadcast the information to the team. For instance, if a (commander) pilot agent's own state contains a belief that would cause the $\boxed{\text{Engage}}_{\Theta}$ operator to be achieved, and such a belief is absent in the team state, then a communicative goal is generated to inform team members (the commander cannot just head back to home base alone). Here, based on the modifications discussed in Section 3.1, the agent first weighs the cost and benefits of communication. If communication by one specific method is prohibited, an agent tries to reduce communication costs by attempting another method. Finally, should the agent satisfy the communicative goal, the sender and the receivers then update their team state.[6] This then causes the team operator to be

---

[6] At present, once information is communicated, it is assumed

terminated (either because it is achieved, unachievable or irrelevant).

In some specific circumstances, agents also track their team or particular teammates rather than waiting for communication from those agents. Agent tracking is based on the RESC technique(Tambe & Rosenbloom 1995), which uses the existing operator hierarchy for tracking others' higher level goals (it executes those operators, and matches operators' predictions with actual observations). However, since many agents act in parallel, it is difficult to track all agents simultaneously. Agents are thus tracked selectively; currently such selectivity is based on hand-coded heuristics.

4.3 Roles and role failures

In the team operator framework, roles are instantiated via suboperators in the operator hierarchy. For the team Θ, a team operator $\boxed{\text{OP}}$ with $\mathcal{R}$ roles is denoted as $\boxed{\text{OP}}_{\Theta}< \gamma_1, ..., \gamma_R >$. Θ may have R sub-teams, $\sigma_1 ... \sigma_R$, which must then undertake each of these roles. Many team operators, however, can be defined via multiple role combinations. For instance, $\boxed{\text{Engage}}_{\Theta}$ performed by anywhere between two to eight agents, some of them attack helicopters and some scouts. Furthermore, the type of roles allowed may change depending on the current situation.

If a separate version of $\boxed{\text{OP}}_{\Theta}< \gamma_1, ..., \gamma_R >$ is defined for each role combination, or for each situation, a large number team operators would need to be represented. Furthermore, for each new role combination, the suboperator combination would need to be defined anew. To alleviate this concern, constraints are specified to implicitly define role combinations. Constraints may specify the composition of subteams and their allowable roles. For instance, for $\boxed{\text{Engage}}_{\Theta}$, the constraint may only specify that the allowable role-performing subteams be individual agents in the team, i.e., the role performing subteam $\sigma_i = \text{I}$ where $\text{I} \in \Theta$; but there can be as many of them as possible. In other cases, allowable subteam may be a subteam of two agents, rather than individuals. Each agent only instantiates the constraint relevant to itself, i.e., it knows if it is expected to act alone or as part of a team or subteam. The actual role an agent undertakes then is based on this allowable subunit, and any static specification of the role that that unit/subunit plays in that situation (where the static specification may be, for instance, that the agent is a scout).

Currently, members of a team track an agent's role performance in one of three ways:

1. Communication from that agent itself;
2. Inference based on the agent's being engaged in some other role (e.g., if a scout is busy scouting, it cannot be participate in any company activity at the holding point);
3. Inference based on the agent tracking.

The heuristics in Section 3.2 are used to determine if the agent's role (non-)performance obstructs the progress of a team's lower-level team operators. If so, agents attempt to repair the team operator. In particular, if another agent is capable of substituting for the role non-performer, it takes over the role and sends out a *role-change* message to the team. For an illustration, consider the following example, addressing the problem in the first example in Figure 3. Here, a company of five helicopters, Cheetah41 through Cheetah45, has the role and capabilities as shown:

Current roles:
Cheetah41 <— Commander, Scout
Cheetah42, Cheetah43,Cheetah44, Cheetah45 <— Attack
Current capabilities:
Cheetah41,Cheetah43 <— Scout
Cheetah42, Cheetah43,Cheetah44, Cheetah45 <— Attack
Chain of command: Cheetah41->Cheetah42->Cheetah43->Cheetah44->Cheetah45

Suppose, the team is currently executing $\boxed{\text{wait-while-bp-scouted}}_{\Theta}$. In service of this team operator, the scout (Cheetah41) is moving forward to scout the battle position, while the rest of the company is waiting at the holding area. Now if the scout crashes (as in Item 1 in Figure 3), $\boxed{\text{wait-while-bp-scouted}}_{\Theta}$ is deemed unachievable (critical expertise heuristic). Two changes will then take place. First, Cheetah43 will take over the critical role of the scout — it has the capability of becoming a scout. This enables the $\boxed{\text{wait-while-bp-scouted}}_{\Theta}$ operator to be re-established for execution. Next, Cheetah42, the next in command, will replace Cheetah41 as the commander.

5 Related Work

Few other research efforts have implemented theories of joint or collaborative action. We are aware of only two other efforts to operationalize the joint intentions framework — Jennings's work on an industrial multi-agent setting(Jennings 1995), Huber and Durfee's effort in a testbed domain(Huber & Durfee 1995). Both suggest modifications to the framework in the course of its operationalization. However, there are several key differences. First, in both these efforts, agents' collaborative activity involves a two level hierarchy of a joint goal and a joint plan, with individuals engaged in specific roles in the plan. When the joint goal is accomplished, the collaborative activity is terminated. In contrast, the work here involves a team jointly executing a long-term, well-practiced team activity, which often involves the execution of a complex, dynamically changing team operator hierarchy. A high-level mission leads to the execution of a whole variety team operators. It thus becomes essential to maintain and track explicit team and subteam states, and manipulate them via team operators — else agents will lose track of the next team action. Second, the above efforts typically involve two-three agents in the joint intention. The shift from two-three agent teams to five-eight agent teams (as in our work) creates new possibilities. More specifically, even if a single agent is incapacitated, at least the highest-level joint intention does not completely fall apart. However, lower-level team operators may be affected, and individuals reason about both the team operator unachievability and repair. Repair is important, else the en-

---

to reach the other agents securely.

tire team effort will go to waste. Finally, in (Jennings 1995) issues of communication risk and cost are not considered.

Our recent work on team tracking(Tambe 1996) — which involves inferring other team's joint goals and intentions based on observations of their actions — is the predecessor to the work reported here. However, given its focus on tracking other teams, it only inadequately addressed issues such as communication, and repair of team or role plans.

6 Summary and Discussion

A variety of dynamic multi-agent environments are currently being developed, some where intelligent agents directly interact with humans(Hayes-Roth, Brownston, & Gen 1995; Tambe *et al.* 1995; Bates, Loyall, & Reilly 1992; Kitano *et al.* 1995). This paper focused on flexible teamwork in such environments. Given the dynamism, complexity and particularly the uncertainty in such environments, it is difficult to anticipate all of the agent interactions. The planner (or the knowledge engineer) in such situations is burdened with the difficult task of designing a large number of specific coordination plans — at least, this has been our experience in the real-world synthetic domain under investigation.

The approach presented in this paper is an attempt to alleviate this difficulty. The key hypothesis is that in dynamic multi-agent domains, there are really two areas of expertise: one is clearly the domain under investigation (e.g., military helicopter tactics), but over and above that is the second, that of commonsense knowledge of teamwork. If agents are equipped with such core knowledge of teamwork, the planner or the knowledge engineer can specify higher-level *team plans*, and let the individual agents reason about the coordination activities and recovery, based on their deeper, commonsense knowledge-base. Agents need not be provided with detailed, low-level coordination plans. Other potential advantages of such a framework include: (i) Improved self-explanations, e.g., an agent can better explain the reasons underlying its communication; and (ii) Guidance for knowledge acquisition, e.g., the framework helps to avoid omissions(Jennings 1995). For a further understanding of the issues involved in more complex teams, we are currently implementing this framework for agents in the RoboCup virtual soccer tournament(Kitano *et al.* 1995).

References

Bates, J.; Loyall, A. B.; and Reilly, W. S. 1992. Integrating reactivity, goals and emotions in a broad agent. Technical Report CMU-CS-92-142, School of Computer Science, Carnegie Mellon University.

Cohen, P. R., and Levesque, H. J. 1991. Teamwork. *Nous* 35.

Davis, E. 1990. *Representations of commonsense knowledge.* Palo Alto, CA: Morgan Kaufmann.

Firby, J. 1987. An investigation into reactive planning in complex domains. In *Proceedings of the National Conference on Artificial Intelligence (AAAI)*.

Grosz, B. J., and Sidner, C. L. 1990. Plans for discourse. In *Intentions in Communication.* Cambridge, MA: MIT Press. 417–445.

Hayes-Roth, B.; Brownston, L.; and Gen, R. V. 1995. Multiagent collaobration in directed improvisation. In *Proceedings of the International Conference on Multi-Agent Systems (ICMAS-95)*.

Huber, M., and Durfee, E. 1995. On acting together: Without communication. In *Proceedings of the AAAI Spring Symposium on Reasoning about Mental states.*

Ingrand, F. F.; Georgeff, M. P.; ; and Rao, A. S. 1992. An architecture for real-time reasoning and system control. *IEEE EXPERT* 7(6).

Jennings, N. 1995. Controlling cooperative problem solving in industrial multi-agent systems using joint intentions. *Artificial Intelligence* 75.

Kinny, D.; Ljungberg, M.; Rao, A.; Sonenberg, E.; Tidhard, G.; and Werner, E. 1992. Planned team activity. In Castelfranchi, C., and Werner, E., eds., *Artificial Social Systems, Lecture notes in AI 830.* Springer Verlag, New York.

Kitano, H.; Asada, M.; Kuniyoshi, Y.; Noda, I.; and Osawa, E. 1995. Robocup: The robot world cup initiative. In *Proceedings of IJCAI-95 Workshop on Entertainment and AI/Alife.*

Levesque, H. J.; Cohen, P. R.; and Nunes, J. 1990. On acting together. In *Proceedings of the National Conference on Artificial Intelligence.* Menlo Park, Calif.: AAAI press.

Newell, A. 1990. *Unified Theories of Cognition.* Cambridge, Mass.: Harvard Univ. Press.

Pimentel, K., and Teixeira, K. 1994. *Virtual reality: Through the new looking glass.* Blue Ridge Summit, PA: Windcrest/McGraw-Hill.

Rosenbloom, P. S.; Laird, J. E.; Newell, A.; ; and McCarl, R. 1991. A preliminary analysis of the soar architecture as a basis for general intelligence. *Artificial Intelligence* 47(1-3):289–325.

Searle, J. R. 1990. *Collective intention and action.* Cambridge, MA: MIT Press. 401–415.

Tambe, M., and Rosenbloom, P. S. 1995. RESC: An approach for real-time, dynamic agent tracking. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*.

Tambe, M.; Johnson, W. L.; Jones, R.; Koss, F.; Laird, J. E.; Rosenbloom, P. S.; and Schwamb, K. 1995. Intelligent agents for interactive simulation environments. *AI Magazine* 16(1).

Tambe, M.; Schwamb, K.; and Rosenbloom, P. S. 1995. Building intelligent pilots for simulated rotary wing aircraft. In *Proceedings of the Fifth Conference on Computer Generated Forces and Behavioral Representation.*

Tambe, M. 1996. Tracking dynamic team activity. In *Proceedings of the National Conference on Artificial Intelligence (AAAI)*.