An Initial Assessment of Plan-Recognition-Based Coordination for Multi-Agent Teams

Marcus J. Huber
Intelligent Agents Group
Orincon Corporation
9363 Towne Centre Drive
San Diego, California 92121
marcush@orincon.com

Edmund H. Durfee
Distributed Intelligent Agents Group
The University of Michigan
1101 Beal Avenue
Ann Arbor, Michigan 48109
durfee@engin.umich.edu

Abstract

Plan recognition remains a largely unexplored paradigm for facilitating coordination. In this paper, we begin to explore domain, task, and agent characteristics which impact upon the utility of using plan recognition for coordinating multiple agents and, in particular, collections of agents organized into competing teams. Agents in our research are supplied plan-recognition capabilities in the form of specially instantiated belief networks called Plan Recognition Networks (PRNs). Our initial experiments test several hypotheses concerning coordination performance as a factor of plan-recognition model expressiveness compared to coordination based upon using communication protocols of varying expressiveness. The initial results demonstrate that plan-recognition capabilities permit agents to realize better coordination performance in many situations due to increased knowledge provided through observation and inference compared to that supplied by communication protocols.

Introduction

An agent that considers the concurrent activities of other agents when deciding on its own activities is usually better able to choose actions that lead to outcomes that it favors. Such *coordination* between an agent's actions and those it expects of others has been an ongoing concern in the multi-agent systems (MAS) community, whether the agents are trying to cooperate, compete, or merely co-exist.

Effective coordination thus requires knowledge about what others are, and will be, doing. A variety of strategies for possessing this kind of knowledge have been developed, including: (1) static specifications of agents' roles in an organization (Corkill & Lesser 1983; So & Durfee 1992) or social laws that agents adhere to indefinitely (Shoham & Tennenholtz 1992); (2) collective, multi-agent plans that agents construct before pursuing their current goals, to ensure the agents act in concert when the plan is executed (Corkill 1979; Georgeff 1983; Ephrati & Rosenschein 1993; Durfee & Montgomery 1991); (3) partial, tentative collective plans formulated during the process of execution through the ongoing exchange of evolving plans (Durfee & Lesser 1991); and (4) immediate, reactive responses to interactions generated on-the-fly without considering long-term repercussions at all (Tambe & Rosenbloom 1995).

A principle concern in the dynamic, anticipatory approaches (2 and 3 in the list above) has been in figuring out how agents should decide what information about their plans and goals they should share, with whom to share it, and when. This has led, for example, to approaches that allow an iterative revelation of the relevant information (Durfee & Montgomery 1991; Ephrati & Rosenschein 1993) or a reliance of "meta-level organizations" that guide such communicative decisions (Durfee & Lesser 1991). More reactive approaches sidestep such issues by assuming that it is instead up to the agent who *needs* the information to acquire it by watching out for it, instead of assuming that others will volunteer the right information at the right time.

In our work, we want to take advantage of the abilities of agents to watch out for their own interests, but we still want agents to be able to anticipate what other agents *will do*, rather than simply reacting to what they *are doing*. We do this by providing agents with the ability to use observations of others not only to recognize current actions, but also to recognize ongoing plans and goals based on the sequences of actions that others take. Once they recognize the plans and goals of others, our agents can employ any of a number of techniques (see (O'Hare & Jennings 1996), especially Chapter 6) for coordinating their planned actions with those of others.

Plan recognition through observation offers several advantages over explicit communication about plans, including lower communication overhead, higher reliability and speed when channels are faulty or congested, greater expressiveness (the communication language might be less rich than what can be recognized[1]), and it is robust in the face of agents who are not forthcoming about their plans or who might be deceitful.[2] Of course, it also has disadvantages, including the fact that agents need to (at least assume they) know how other agents act in pursuit of their goals, that actions might not be accurately observable or might support multiple possible plans, and that inferring the plans of others can be more time-consuming than being told explicitly by them.

In this paper, our objective is to demonstrate that plan-recognition-based coordination can achieve many of the advantages listed above, without suffering overly from the disadvantages. Specifically, while for now we assume that agents have reasonably good models of how others accomplish their goals, we do not assume that agents take actions that are unambiguous. We therefore explicitly deal with the inherent uncertainty involved in plan recognition, and we empirically demonstrate that in domains such as our test domain, uncertainty can be managed and plan-recognition inferences can be accomplished without excessive overhead.

In the next section, we describe our approach for explicitly handling plan-recognition uncertainty by mapping the procedures agents follow into a belief network representation that facilitates managing uncertainty efficiently and soundly. We then turn to application domains where plan-recognition-based coordination makes sense, and in particular describe the experimental domain that we use for evaluating the impact of plan-recognition-based coordination. The subsequent section posits a series of hypotheses about the effects of using goal-based agents with plan recognition in the experimental domain, and reveals the strengths and limitations of our approach as those hypotheses are tested. Finally, we conclude the paper with our assessment of the role of plan-recognition-based coordination in multi-agent systems.

Probabilistic Plan Recognition

We argue that native plan representations - those which autonomous agents execute - are not conducive to agents wishing to use them to perform plan recognition. Plan recognition requires the ability to reason "backward" from actions to determine the active goal(s) that motivated the action. Plan representations were designed with other issues in mind, however, such as execution efficiency, representational clarity, and conciseness. Similarly, current plan-recognition representations were designed to address issues other than execution. To bridge this gap, we have conceived of and implemented ASPRN (Automated Synthesis of Plan Recognition Networks) (Huber, Durfee, & Wellman 1994), a computational system that automatically constructs a plan-recognition representation directly from an agent's executable plans.

In brief, our approach to plan-recognition-based multi-agent coordination starts with a plan library of an observed agent. From this plan library, ASPRN then creates a probabilistic model which we call a Plan Recognition Network[3] (or PRN for short). PRNs provide a well-founded model (based upon probability theory) with which to perform plan recognition, and also deal naturally with the uncertainty inherent in both the inferencing process and domain (e.g., observational uncertainty such as inaccuracy and perception system errors (Huber & Durfee 1993)). Agents can then use the PRNs to incorporate evidence in the form of observations of other agents' actions and, based upon this evidence, determine through inference the most likely goals and plans[4] that the observed agent is executing.

To give an idea of how PRNs work, the inferred information is represented within a PRN as a posterior probability distribution over possible states in each node. The state space of each node depends upon the type of plan component that it represents. For instance, nodes representing goals have a state space of {*Inactive, Active, Achieved*}, while nodes representing primitive actions have a state space of {*Notperformed, Performed*}. As observations of an agent that is working on achieving a particular goal are made, the posterior distribution for the PRN's associated goal node skews toward the *Achieved* state. Posteriors for alternative goals, those that are not being achieved, skew toward the *Inactive* state. The observing agent can query its PRNs for information regarding various nodes and can then reason about possible conflicts or synergies with its own goals and modify its own behavior accordingly.

Agent plans

Our research assumes that the coordinating agents are "deliberative". That is, the agents do not simply react

[1] As in the saying, "A picture is worth a thousand words."

[2] And, "Actions speak louder than words."

[3] This is not the same as Charniak and Goldman's plan recognition network of (Charniak & Goldman 1991). While they serve the same function, our PRNs were designed explicitly to deal with executable plan representations and issues. Charniak and Goldman's PRNs were designed to deal with natural language issues such as syntactics, semantics and pragmatics.

[4] Information concerning the beliefs, internal state, and even the most likely next primitive action of the observed agent can also be determined with PRNs.

to situations, but reason about a structured response, perhaps in the form of goals and plans, for accomplishing those goals. One of the primary issues in coordination based upon plan recognition quite naturally is the modeling of other agent's goals, plans, and actions. We assume these agents' plans can be modeled with a certain set of constructs, such as sequences of actions, subgoaling, iteration, context (i.e. preconditions), state (i.e., beliefs), and conditional execution of alternative execution paths (i.e., branches). While the plans of the agents being observed and coordinated through plan recognition may not explicitly contain such constructs as those that we model them as having, we are interested in agents having the ability to model the behavior resulting from execution of their plans *as if their plans did* have such constructs. The accuracy of such a model becomes apparent in the success with which such a characterization supports recognition of the plan(s) being executed and the ability to generate accurate hypotheses regarding the future actions of the agent.

Our initial efforts have utilized one particular plan representation, which we believe represents the common plan constructs in a relatively generic manner. The representation is that of UM-PRS (Huber *et al.* 1993), a C++ implementation of the Procedural Reasoning System (PRS) of Ingrand, Georgeff, Rao, et. al. (Ingrand, Georgeff, & Rao 1992). Please refer to these references for representation details. Of primary concern at this point is that UM-PRS supports action sequences, conditional branching, contextualized goals and subgoals, and iteration.

Coordination of Teams of Agents

The form of team coordination that we are exploring is that which is facilitated by each agent's own observations and plan-recognition inferences. In this paradigm, each team member continually performs dynamic and opportunistic reasoning about coordinating with others on the team based upon its own goals and the relative merits of each other team member's goals. When a team member recognizes a situation where the benefits of assisting another team member outweigh the benefits of its own current tasks, the first agent will assume the tasks or roles most advantageous to the agent that it has decided to aid. Team coordination, then, emerges as a result of each agent's decision-making with regards to coordinating with other agents of the same team or against agents of the opposing team.

The experiments that we have performed so far can be expressed in terms of the following informal hypotheses:

- Hypothesis 1: An agent that has an explicit representation of goals and procedures for accomplishing them is at least as competent as an agent where all of this is implicit, but which might incur more overhead.
- Hypothesis 2: An agent that has the explicit representation is more flexible when circumstances change. Specifically, when a communication blackout occurs, the explicit representation (along with ASPRN) allows plan-recognition-based agents to perform much better.
- Hypothesis 3: Plan recognition with rich models will be superior to communication when communication is rooted in a limited language/protocol.
- Hypothesis 4: Performance of plan recognition will degrade when the models of other agents are limited (e.g., incomplete).
- Hypothesis 5: Even poor plan-recognition models can be better than relying solely on communication when communication fails!

The domains that we are particularly interested in exploring are dynamic, uncertain environments where agents execute complex plans and where there are time pressures to correctly coordinate with or against other agents. The research described in this paper is not limited in applicability to just these domains, however. The truths revealed by testing the above hypotheses are applicable to any coordination domain where observation of other agent's actions is possible. For example, our approach should work well in dynamic air combat domains such as described by Tambe in (Tambe & Rosenbloom 1995) and certainly in less time-critical applications such as intelligent user interfaces. The hypotheses above are, of course, very general, and we are just beginning to test some of the issues that arise. However, as will be shown by our initial experimental results, the above hypotheses do appear to be valid, and should be of some usefulness to others exploring coordination paradigms for multi-agent systems.

Experiments

The world in which we have placed our plan-recognizing agents is a dynamic, real-time, fine-grained, multi-agent simulator originally constructed for the Internet network game called Netrek ((Ahn 1996) is a good source for more information). In Netrek, agents act in a grid world where the agents are organized into two teams of up to 8 agents on each team. The environment has distributed throughout it forty labeled, immobile landmarks (called planets) each with a dynamically changing, limited number of resources (called "armies"), with each team initially "owning" ten of the planets. The world updates rapidly (10 updates/second); agents can perform various movement, combat, navigation, and tactical actions up to five times a second. Individual agents engage in highly dynamic "dog-fights" with individual agents of the other team, which involves arcade-like reactivity and real-time management of ship speed, course, shields, weapons, and other functions. Those agents that are victorious are then capable of picking up resources (armies) from the team's own planets (which produce armies) and transporting the armies to a planet owned by the opposing team. If more armies are transported to an opponent's planet than that planet has opponent's armies, the planet changes ownership to the team transporting armies. The global objective of each team in the simulation is to capture the other team's planets. The termination conditions for the experiments is when the losing team owns 7 or fewer planets.

The Netrek environment domain is a challenging, real-time environment with a number of realistic characteristics. The environment changes dramatically and quickly, with agents moving about quite rapidly, acting upon each other and upon the environment itself. Sensing of other agents' activity is uncertain, depending upon their distance and other factors. Each agent must reason about a large number of dynamic objects (the other agents and their weapons) and low-level actions as well as managing high-level roles (as discussed above). Human players find the environment extremely challenging to maintain cognitive awareness of both the tactical (short range) display and the galactic (long range) display due to the fast pace and large amount of information to process and the number of decisions to make.

Domain Tasks and Plans

The agents that we implemented were originally programmed in the *C* language by Ted Hadley, currently a research scientist at the University of California, Irvine. In order to model these agents using ASPRN-constructed PRNs, this original code was in part replaced by UM-PRS Knowledge Areas (plans) and UM-PRS primitive functions. The original agent design partitioned the top-level task of winning the game into a number of roles, or states. We followed the same scheme in the UM-PRS plans. The possible roles (goals) that the agents can take include: dog-fighting with the closest opponent (ENGAGE), bombing or taking a planet (ASSAULT), help a team agent when it tries to take an opponent's planet (ESCORT), suicide attack an opponent (OGG), defend a team-owned planet from an assault (PROTECT), repair and/or refuel (RECHARGE), and move to a team-owned planet that has armies and pick them up (GETARMIES). The UM-PRS agents maintained the exact same functionality as the original agents in all respects, including the capability of communication with other agents.[5]

The resulting UM-PRS plan library as used by agents is too large to list here unfortunately, but consisted of 33 plans in total. This plan library was given to ASPRN, which produced a PRN of 105 nodes and 150 arcs that modeled the plan library.

The agents' basic decision-making architecture was left as it was when obtained from Hadley in order to clearly attribute performance differences. The decision-making architecture is a set of order-dependent heuristics ordered by role according to the relative importance of the role in winning the game. Each agent determined its role (goal) based upon its view of the current situation (each agent performed its own perception) and maintained some persistence with respect to its role once a role was undertaken. During decision-making, each plan-recognizing agent queried posterior distributions (beliefs) of random variables within it PRNs for relevant information[6] approximately once every 0.7 seconds.[7]

The plan-recognizing agent would use the information extracted from the PRNs to determine what other agents were doing and, if it determined that one of the other agents was doing (or was going to be doing) something that it should help or hinder, it would switch to the appropriate role. For example, if an agent moving to bomb a planet determined that another agent was already going to bomb that planet, it would either select another planet to bomb or take on some other role, such as defending armies on its own planets or defending against an enemy carrying armies. As another example, if an agent noticed that a team/opponent agent was going to take a planet and it was in a position to coordinate, it would escort or defend the planet as appropriate.

As was mentioned earlier, one of the major issues associated with coordination via plan recognition is the uncertainty and computation overhead incurred by needing to infer information that might otherwise (in a perfect world) simply be communicated. The Ne-

[5]Although communication was never actually utilized by the UM-PRS agents in the experiments described in this paper.

[6]Each agent maintained one PRN for each other agent that it was coordinating with or against

[7]This was an arbitrary value that seemed to work well and was never changed - this would be an interesting parameter to change in future experiments.

trek environment introduces uncertainty (sensed locations of other agents can have reduced precision or be slightly randomized), as does the agent's observation functions, which are non-trivial and are not infallible.

Experiment Statistics

During each experiment, we accumulated statistics which are typically accumulated during human Netrek matches. These statistics are analyzed after human games to determine relative performance and can provide insight into autonomous agent performance as well. These statistic consisted of:

- tpt - total planets taken.
- tpd - total planets destroyed (i.e., reduced to 0 armies).
- tek - total dog-fights won.
- def - the number of times team agents died.
- tab - total armies bombed.
- tac - total armies carried.
- cak - carried armies killed.
- eao - enemy armies ogged.

Some additional information was also captured, such as total experiment time (referred to as "time" in experiment results shown later) and bombing latency (the accumulated time between when a planet grew armies and when the planet was bombed, and referred to as "latency" in the results).

Explicit vs. Implicit Representation

We briefly describe our first experiments' results as they simply represent a baseline comparison of the UM-PRS agents (without plan-recognition or communication capabilities) when they competed against the equivalent *C* agents ("Stdbots"). We hypothesized that the UM-PRS agents with their explicit representation of goals and plans could only do as well, but not better, than the Stdbots because of the inherent overhead of UM-PRS.

Out of forty games, the Stdbots won 25, or 62.5 percent, of them. In these games, the Stdbots performed slightly better than the UM-PRS agents almost across the board, indicating a slight performance decrease due to the overhead associated with the use of UM-PRS and its explicit goal and plan representation and execution scheme. Clearly, the computational overhead associated with UM-PRS's more general and flexible architecture outweighed its benefits in this simple, direct translation.

Plan recognizing Agents vs. Non-communicating *C* Agents

Our next experiments were conducted to test how flexible the agents that coordinate based upon PRNs would be against the original agents, but in this case in a world where communication channels are disabled for some reason (e.g., being jammed). The results of experiments of the plan-recognizing agents ("PRbots") competing against non-communicating Stdbots is shown in Table 1.

In a series of forty games, the PRbots won 36. Almost all of the statistics in the table point to dominance by the PRbots. The PRbots captured and destroyed more planets, won more dog-fights, carried more armies, and killed more enemy-carried armies. Bombing latency reflects that the PRbots coordinated much better in their bombing and distributed themselves better when multiple planets needed bombing (more on this later). The higher "tab" statistic for the Stdbots seems at first glance to indicate that the Stdbots did a better job bombing overall, but this is mitigated by the fact that, as the experiments progressed, the Stdbots had fewer planets producing armies so the PRbots had less opportunity to bomb them.

In this set of games, the PRbots clearly outmatched the Stdbots. One important result of these runs was to establish that the PRbots were apparently able to recognize the goals/plans of the other agents early enough to give one or more of the PRbots an opportunity to be in the right place and time to help or hinder, as the situation required. In contrast, had the same observations been made just before the observed agent completed the critical portion of its task (e.g., dropping armies on a planet) the the PRbots would have been unable coordinate with the observed agent. This introduces the concept of "observation distribution" - a measure of where observable actions occur during execution of a plan (e.g., early or late in the plan). This is an extremely important issue in plan recognition and one that deserves much more attention.

Another important aspect was the negligible overhead associated with performing plan-recognition observations and inferencing (typically 0.02 real-time seconds). Clearly this is an ideal situation, and in future work we will simulate more complex and higher cost perceptual processing and inferencing in order to gain a better understanding of where plan recognition becomes too much of a burden to be of utility.

Our observations showed that the PRbots bombed in a much more coordinated fashion, using their plan-recognition capabilities to determine that some other teammate was already bombing (or going to bomb) an enemy planet and choosing another planet to bomb

Team	Stat.	total	avg.	std.dev
PRbots	tpt	122	3.05	5.86
	tpd	143	3.58	5.80
	tek	954	23.85	52.85
	def	752	18.8	49.04
	tab	8057	201.43	460.60
	tac	1134	28.35	44.31
	cak	37	0.93	6.33
	eao	80	2	8.60
	time		16469	
	latency		30649	
Stdbots	tpt	47	1.18	5.53
	tpd	46	1.15	5.20
	tek	738	18.45	48.22
	def	974	24.35	52.72
	tab	8136	203.4	476.03
	tac	484	12.1	50.79
	cak	80	2	8.60
	eao	36	0.9	6.34
	time		15338	
	latency		47219	

Table 1: Experiment statistics of full-model PRbots vs. non-communicating Stdbots.

Team	Stat.	Total	Average	Std.Dev.
PRbots	tpt	129	3.23	7.38
	tpd	134	3.35	6.80
	tek	937	23.43	59.82
	def	766	19.15	44.24
	tab	8005	200.13	521.73
	tac	1178	29.45	60.41
	cak	65	1.55	12.53
	eao	59	1.48	12.21
	time		15148	
	latency		17743	
Stdbots	tpt	41	1.03	7.46
	tpd	56	1.4	8.47
	tek	776	19.4	53.62
	def	981	24.53	69.76
	tab	8167	204.18	620.64
	tac	511	12.78	71.54
	cak	64	1.6	12.21
	eao	65	1.63	14.71
	time		12478	
	latency		26234	

Table 2: Experiment statistics of full-model PRbots vs. Stdbots with limited communication.

or, if there were no more planets to bomb, switching to some other role. The Stdbots, on the other hand, quite often bombed planets *en masse* and oftentimes moved and bombed in a cluster. The PRbots, with their better bombing efficiency, had more opportunity to perform other roles, such as protecting armies and defending planets from enemies carrying armies.

Plan recognizing Agents vs. Communicating C Agents

Hypothesis 3 claims that the use of rich plan-recognition models can yield better coordination performance than when limited communication capabilities are used. In these experiments, a Stdbot utilized a restricted communication protocol to inform teammates of its intent to attempt to capture a planet[8]; the receiving C agents would use this information in determining their own courses of action and would assist the planet-capturing agent if they could.

The results of forty experiments of the PRbots competing against Stdbots with limited communications capabilities is shown in Table 2. Of the forty experiments, the PRbots won 35 while the Stdbots won just five.

The statistics in Table 2 again show dominance by the plan-recognizing agents (PRbots). Bombing latency again shows the PRbots' improved bombing coordination, with total bombing being nearly equivalent. Again, this increased efficiency permitted the PRbots to be more flexible, permitting them to switch more dynamically to other, more useful, roles. One significant item to note in the results of these experiments is that the "eao"/"cak" ratio was reduced from a nearly 2:1 ratio in the non-communicating experiments to a 1:1 ratio in the communicating experiments. This indicates that the Stdbots were much more successful in protecting (coordinating with) teammates when the teammates were taking planets.

Coordination through utilization of a limited communication protocol by the Stdbots led to increased Stdbot success in safely delivering armies to the PRbot planets. The Stdbots' communication language and protocol, as defined for the C agents, are not powerful enough, however, to overcome the flexibility provided the PRbots by PRNs. The relatively rich modeling of the complete task structure maintained by the PRbots provided them with a broader scope of coordination information than that provided by the Stdbots' communicated information, even though the information provided to the PRbots by the PRNs was uncertain. Of note here is that the overhead associated with the Stdbots' communication was negligible, posing virtually no computational load and having virtually instantaneous transmission. Issues and tradeoffs related to non-negligible communications with regard to relative performance against plan recognition is yet another re-

[8]This was the default C agent configuration and communication protocol supplied by Hadley. Future work will examine relative performance as the expressiveness of the protocol increases.

Team	Stat	total	avg	std.dev
PRbots	tpt	72	1.8	6.78
	tpd	87	2.18	8.80
	tek	904	22.6	59.24
	def	919	22.98	42.16
	tab	9104	227.6	562.97
	tac	737	18.43	67.67
	cak	41	1.03	4.65
	eao	72	1.8	11.68
	time		18545	
	latency		23310	
Stdbots	tpt	86	2.15	5.9287
	tpd	102	2.55	6.3126
	tek	919	22.98	40.604
	def	959	23.98	61.953
	tab	9105	227.63	539.29
	tac	836	20.9	46.931
	cak	81	2.03	13.808
	eao	41	1.03	4.6542
	time		15382	
	latency		22475	

Table 3: Experiment statistics of restricted-model PRbots vs. communicating Stdbots.

Team	Stat	total	avg	std.dev
PRbots	tpt	79	1.98	9.01
	tpd	100	2.5	12.37
	tek	914	22.85	69.78
	def	878	21.95	48.32
	tab	9116	227.9	637.72
	tac	861	21.53	92.11
	cak	57	1.43	14.07
	eao	129	3.23	14.43
	time		19392	
	latency		22819	
Stdbots	tpt	72	1.8	9.3380
	tpd	95	2.375	10.274
	tek	887	22.175	53.309
	def	958	23.95	75.309
	tab	9674	241.85	648.36
	tac	883	22.075	63.795
	cak	134	3.35	14.575
	eao	57	1.425	14.071
	time		15282	
	latency		24532	

Table 4: Experiment statistics of restricted-model PRbots vs. non-communicating Stdbots.

search area to be explored.

Restricted Plan-Recognition Model Agents vs. Communicating Agents

In a test of Hypothesis 4, the PRbots were limited to using a restricted plan-recognition model, one corresponding to only the information included in the inter-agent communication protocol (i.e., only the planet-taking portion). In this set of games, the Stdbots were able to use their limited-protocol communication. We expected to see a reduction in the PRbot team's performance due to the limitations placed upon the plan recognition models used. However, it was unclear to us how the PRbots' ability to recognize planet taking of agents on both teams compared to the Stdbots' ability to communicate without uncertainty.

The result of forty experiments showed the PRbots finally coming out on the losing end, winning only eighteen of the forty games. In these games, the PRbots fell into the mass bombing pattern demonstrated by the Stdbot and were subsequently less flexible in their ability to change roles in a timely manner. The "head-to-head" competition of communication and a functionally similar plan recognition model provides some evidence that the additional overhead involved with plan recognition processing and uncertainty sometimes outweighs the benefits gained by plan recognition. An interesting area of research would involve exploring team performance as communication and plan recognition costs varied.

Restricted Plan-Recognition Model Agents vs. Non-communicating Agents

To test Hypothesis 5, we again gave the PRbots an incomplete model of the Stdbot's plan library. But, in comparison to the previous experiments, the Stdbots' ability to communicate was disabled, simulating an event such as equipment failure or communication jamming. Of forty experiments, the PRbots won 22 of them, in comparison to 18 victories where the Stdbots could communicate.

Game statistics shows that the two teams were fairly evenly matched except for a tremendous difference in carried armies killed and enemy armies killed. The PRbots demonstrated that they were able to effectively coordinate against the opposing team (and with their own team) even with their incomplete plan model. The results of these experiments demonstrate that the Stdbots' reliance upon communication, which is advantageous if communication is available, leaves them vulnerable when the situation is such that communication cannot be used.

Conclusions

Our experiments verified our hypotheses that the flexible UM-PRS architecture would be slightly detrimental compared to functionally equivalent *C* agents, and that the PR-based coordination of UM-PRS robots would be superior to the *C* agents if communication was impossible for both types of agents. We were surprised, however, with our results on a more fair comparison, where the C agents use communication and UM-PRS agents use plan recognition. We expected that the teams would be fairly evenly matched, while in fact the plan-recognition-based agents were significantly better. Our analysis revealed that the strength of plan-recognition-based coordination is that it allows agents to infer as much as they can about others based on observations, instead of being restricted to only knowing as much about others as can be expressed in the communication language. Of course, the tables can be turned in cases where observability is even more limited than language expressibility. But a clear outcome of our studies so far has been to establish the benefits of PR-based coordination when the "picture" is indeed worth a thousand (or more) words!

The research discussed here only touches lightly upon a very rich and deep set of research issues. We have noted some of these issues in the text of the paper and currently are exploring aspects of several issues, including the affect of observation distribution and observational uncertainty upon coordination performance.

Acknowledgements

This research was sponsored in part by the NSF under grant IRI-9158473, and by DARPA under contracts DAAE-07-92-C-R012 and N66001-93-D-0058. We would like to thank Milind Tambe and the reviewers for very helpful comments.

References

Ahn, D. 1996. International Netrek League WWW Home Page. World Wide Web. http://indigo2.medeng.bgsm.edu/ ahn/INL/index.html.

Charniak, E., and Goldman, R. 1991. A probabilistic model of plan recognition. In *Proceedings Ninth National Conference on Artificial Intelligence*, 160–165. Anaheim, CA: American Association for Artificial Intelligence.

Corkill, D. D., and Lesser, V. R. 1983. The use of meta-level control for coordination in a distributed problem solving network. In *Proceedings of the Eighth International Joint Conference on Artificial Intelligence*, 748–756.

Corkill, D. D. 1979. Hierarchical planning in a distributed environment. In *Proceedings of the Sixth International Joint Conference on Artificial Intelligence*, 168–175.

Durfee, E. H., and Lesser, V. R. 1991. Partial global planning: A coordination framework for distributed hypothesis formation. *IEEE Transactions on Systems, Man, and Cybernetics* 21(5):1167–1183. (Special Issue on Distributed Sensor Networks).

Durfee, E. H., and Montgomery, T. A. 1991. Coordination as distributed search in a hierarchical behavior space. *IEEE Transactions on Systems, Man, and Cybernetics* 21(6):1363–1378. (Special Issue on Distributed AI).

Ephrati, E., and Rosenschein, J. S. 1993. Multi-agent planning as a dynamic search for social consensus. In *Proceedings of the Thirteenth International Joint Conference on Artificial Intelligence*, 423–429.

Georgeff, M. 1983. Communication and interaction in multi-agent planning. In *Proceedings of the National Conference on Artificial Intelligence*, 125–129. (Also published in *Readings in Distributed Artificial Intelligence*, Alan H. Bond and Les Gasser, editors, pages 200–204, Morgan Kaufmann, 1988.).

Huber, M. J., and Durfee, E. H. 1993. Observational uncertainty in plan recognition among interacting robots. In *International Joint Conference on Artificial Intelligence Workshop on Dynamically Interacting Robots*, 68–75.

Huber, M. J.; Lee, J.; Kenny, P.; and Durfee, E. H. 1993. *UM-PRS Programmer and User Guide*. The University of Michigan, Ann Arbor MI 48109.

Huber, M. J.; Durfee, E. H.; and Wellman, M. P. 1994. The automated mapping of plans for plan recognition. In *Proceedings of the 1994 Distributed AI Workshop*, 137–152.

Ingrand, F.; Georgeff, M.; and Rao, A. 1992. An architecture for real-time reasoning and system control. *IEEE Expert* 7(6):34–44.

O'Hare, G. M. P., and Jennings, N. R. 1996. *Foundations of Distributed Artificial Intelligence*. Wiley Interscience.

Shoham, Y., and Tennenholtz, M. 1992. On the synthesis of useful social laws for artificial agents societies (preliminary report). In *Proceedings of the National Conference on Artificial Intelligence*, 276–281.

So, Y., and Durfee, E. H. 1992. A distributed problem-solving infrastructure for computer network management. *International Journal of Intelligent and Cooperative Information Systems* 1(2):363–392.

Tambe, M., and Rosenbloom, P. S. 1995. RESC: An approach for real-time, dynamic agent tracking. In *Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence*, 103–110.