A Hierarchical Protocol for Coordinating Multiagent Behaviors

Edmund H. Durfee and Thomas A. Montgomery
Department of Electrical Engineering and Computer Science
University of Michigan
Ann Arbor, Michigan 48109
durfee@caen.engin.umich.edu, monty@caen.engin.umich.edu

Abstract

We describe how a behavior hierarchy can be used in a protocol that allows AI agents to discover and resolve interactions flexibly. Agents that initially do not know with whom they might interact use this hierarchy to exchange abstractions of their anticipated behaviors. By comparing behaviors, agents iteratively investigate interactions through more focused exchanges of successively detailed information. They can also modify their behaviors along different dimensions to either avoid conflicts or promote cooperation. We explain why our protocol gives agents a richer language for coordination than they get through exchanging plans or goals, and we use a prototype implementation to illustrate our protocol. We argue that our hierarchical protocol for coordinating behaviors provides a powerful representation for negotiation and can act as a common foundation for integrating theories about plans and organizations.

Introduction

In a world inhabited by numerous active systems (agents), the agents often must engage in cooperative and competitive behavior in order to achieve their goals. To decide how they should behave, they need to know how agents with whom they might interact might behave. Just as important, they need to avoid expending their limited reasoning resources on studying how agents with whom they will not interact might behave. If potential interactions between agents can be predicted because each has some predefined role (due, for example, to commonly known organizational constraints [Corkill and Lesser, 1983; Durfee *et al.*, 1987] or to functional or geographical relationships [Conry *et al.*, 1988]), then channeling detailed information about planned actions is straightforward. On the other hand, if agents are ignorant about with whom they might interact, then they need to narrow down the possibilities before they exchange detailed information, otherwise they might swamp the communication channels and each other. Mobile robots on a warehouse floor, for example, can have an ever-changing group of neighbors, and when plotting collision-free paths for the near future should identify and converse only with their current neighbors.

We introduce a hierarchical protocol for this type of problem. In our protocol, an agent represents its anticipated behavior at numerous levels of abstraction. When it must decide with whom it might interact, the agent blindly broadcasts its most abstract behavioral information rather than details about its planned actions. Then, as an agent receives abstract information from others, it compares anticipated behaviors to decide with whom it might interact, and then exchanges more detailed information with only those agents. In essence, the protocol allows the agents to engage in a dialogue in which they can move between different levels of detail about behavior, feeling each other out to discover who they can safely ignore and how they should change their behaviors based on anticipated interactions with agents that they cannot ignore.

Our protocol advances the field of distributed artificial intelligence (DAI) in two ways. The obvious way is that it adds to DAI's arsenal of coordination techniques, permitting a form of coordination that has been unavailable. By enlarging the set of coordination techniques, we both extend the range of coordination problems that we can solve and learn more about the nature of intelligent coordination. The more subtle but potentially far-reaching contribution of our protocol is that it introduces the use of behaviors into multi-agent coordination. Behaviors subsume the more traditional representations of goals and plans, and even blur the boundaries between plans and organizations. As a result, our protocol has the potential to integrate diverse theories and mechanisms for intelligent planning and organizational design. We return to these possibilities in the concluding section. First, we outline our protocol, relate our protocol to other research, and describe a preliminary implementation of the protocol and experimental results.

[0]This research was sponsored, in part, by the University of Michigan under a Rackham Faculty Research Grant, and by a Bell Northern Research Postgraduate Award.

The Protocol

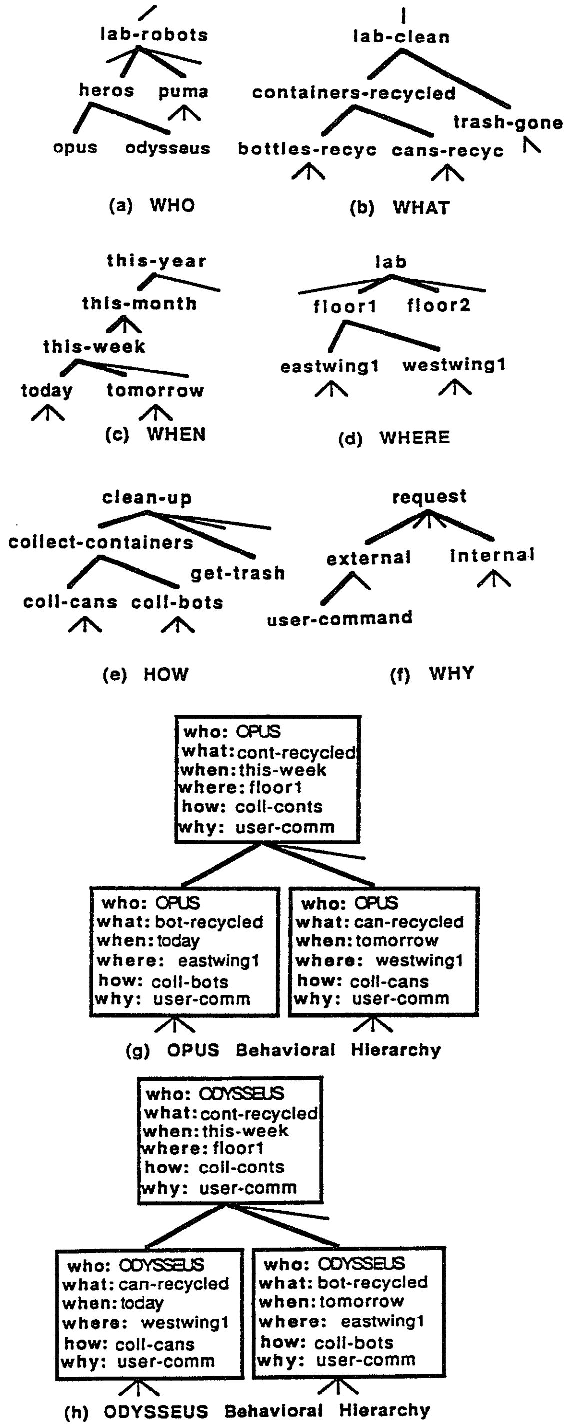
At the heart of our protocol is the concept of a *behavior hierarchy*. To define what this hierarchy is, we first indicate what it is not. It is not a plan hierarchy, where plans are represented in varying levels of detail and detailed subplans are linked to their more abstract counterparts. Similarly, it is not a goal hierarchy, that links goals and their subgoals for a given task. Although plan and goal hierarchies are useful for deciding on actions to achieve results, they only describe limited aspects of an agent's behavior. For example, plan hierarchies emphasize *how* to do things: At the abstract levels are vague, encompassing instructions, while the detailed levels prescribe specific actions. Alternatively, goal hierarchies emphasize *what* to do: At the abstract levels are broad objectives, and at the detailed levels are specific, atomic goals.

A behavior hierarchy subsumes plan and goal hierarchies because it represents the what and how of agents, and also the who, when, where, and why. That is, an entry in the behavior hierarchy represents *who* are behaving in a particular way, *what* they are trying to achieve, *when* they are behaving this way, *where* they are behaving this way, *how* they are acting to accomplish this behavior, and *why* they have initiated this behavior. Each of these 6 aspects represents a different dimension for composition or decomposition.[1] In fact, we might have hierarchies for each dimension, as well as for the overall behavior. Example partial hierarchies are shown in Figure 1. Note that moving between entries in a behavior hierarchy can involve moving along several separate dimensions simultaneously.

Our protocol currently assumes that all agents use the same language to describe the different behavioral dimensions. The agents thus can map a received behavior into their local representations; in essence, the behavioral dimensions describe a 6-dimensional space, and an agent can find the region in this space that a received behavior occupies. From this mapping, an agent can recognize potential behavioral interactions when behaviors' regions overlap or are proximal. The ability to recognize interacting behaviors and to move along dimensions to find alternative interactions is at the core of our protocol.

With this representation in mind, we now present a generalized version of the protocol; a more detailed description for a particular application is given later. An agent with its own tasks (or a group of agents with group tasks) forms a behavior hierarchy representing a decomposition of the tasks along the dimensions that are appropriate given the tasks and the environment. Agents then exchange information at the highest level of abstraction, and map received information into their local behavior hierarchy. By inspecting

[1]Other dimensions might be incorporated into the representation in the future. However, the 6 currently included were an obvious initial choice.



In (a) is a partial personnel hierarchy, (b) is a partial goal hierarchy, (c) is a partial temporal hierarchy, (d) is a partial spatial hierarchy, (e) is a partial plan hierarchy, and (f) is a motivation hierarchy. In (g) and (h) are partial behavior hierarchies for robots OPUS and ODYSSEUS, respectively.

Figure 1: Example Partial Hierarchies

relationships between behaviors along the different dimensions, an agent can recognize potential negative interactions (such as resource conflicts) with another agent. For example, if the robot Opus, with the behavior hierarchy in Figure 1g, receives from Odysseus its most abstract behavior (Figure 1h), then it finds a potential conflict along the when and where dimensions. The robots will be on the same floor in the same time interval, and thus could collide.

When an agent recognizes a potential negative interaction, it has two alternatives. One choice is to modify its own behaviors such that the interaction is avoided. To do this, it searches through alternative values along the behavioral dimensions that lead to acceptable, but no longer interacting, behaviors.[2] For example, Opus might modify its behavior so that it works on the floor next-week instead. The other choice an agent has is to refine its view of the potential interaction, so as to identify more specifically how the behaviors are interacting, or possibly to discover upon closer inspection that there is no interaction. To pursue this choice, the agents move down to the next level of the hierarchy and exchange only the behavioral information that is relevant to the potential interaction. When it receives new behavioral information at less abstract levels, an agent maps this information into its hierarchy and the process repeats. For example, at the more detailed level, Opus learns that he and Odysseus will be working in different wings, and so no collision is possible (Figures 1g and 1h). If they discover that their behaviors do not, in fact, interact, the agents discontinue their communications. Thus, agents that are not interacting discover this with minimal communication of more detailed information.

Agents can also interact positively, and at times might change their behaviors to capitalize on positive interactions. For example, Opus and Odysseus might decide that one of them could achieve the goals of both. Opus might extend its behaviors for today and tomorrow along the "what" dimension (to "containers-recycled") and along the "how" dimension (to "collect-containers"), while Odysseus is now free to pursue other behaviors. In our protocol, an agent has a choice between modifying its behavior at the current level of abstraction to improve interactions, or exchanging information to probe more deeply into positive interactions and establish them at a more detailed level. Detailed exchanges can lead to crisper coordination. For example, if Odysseus wants to meet Opus to cart away the containers collected in the morning, then going to more detailed levels allows them to decide more specifically when and where to meet.

This outline of our protocol leaves many unanswered questions. Some deal with the agents' knowledge: Where does the hierarchical knowledge come from? How can we ensure consistency between agents? How do we represent constraints that moving along one dimension might impose on other dimensions? Other questions deal with effectively using the protocol: When should agents initiate an exchange of abstract behavioral information? How do agents decide which of them should modify their behavior to avoid negative or promote positive interactions? How do agents decide which dimensions of their behavior to modify? What happens when modifying an interaction with some agents introduces new interactions with other agents? Can the protocol guarantee convergence? What are the computational and communication costs of the protocol? Answering these questions is an ongoing effort, and our initial answers to some of them are only a first step. Before discussing these, we relate our protocol to prior research.

[2] Searching hypothetical behaviors is similar to searching hypothetical cases for argumentation [Ashley, 1988].

Relationship to Other Research

Our protocol addresses problems where the spatial and functional relationships between agents change over time. These issues also arise in the air-traffic control problems studied at Rand [Cammarata *et al.*, 1983; Steeb *et al.*, 1988]. Unlike our protocol, the Rand approach involves centralizing authority. Specifically, in their approach a group of potentially interacting aircraft engage in a discussion to choose a group leader. Once chosen, the leader collects detailed information from each group member, designs a group plan to avoid negative interactions (aircraft collisions), and then tells each aircraft what it should do. Although the leader could send processing tasks to group members during the planning phase [Steeb *et al.*, 1986], control is centralized.

The Contract-Net protocol [Davis and Smith, 1983] provides a mechanism by which autonomous agents negotiate to assign tasks among themselves to balance load and connect tasks with the agents that are most suited to pursuing them. The view taken by Contract-Net is that interactions between agents are due to mutually agreed upon task assignments; unlike our protocol, the Contract-Net protocol does not anticipate that agents might independently take actions that could lead to unintended interactions.

The multistage negotiation protocol developed by Conry and her colleagues [Conry *et al.*, 1988] does address the need to identify and recover from resource conflicts. Developed in the context of a distributed network management task, their approach first propagates information about tasks (circuits to complete) through the network, so that each subnetwork controller can enumerate the possible ways it can assign its resources. The controllers then engage in an iterative dialogue where they tentatively choose combinations of assignments, exchange information to detect constraint violations (overburdened channels that connect their subnetworks), and eventually converge on a consistent set of assignments (even if it means neglecting low-priority tasks). Unlike our protocol, multistage negotiation assumes a fixed topography that controllers use to guide communication. Also, their approach assumes that the agents can enumerate *all* of the possible resource assignments, and so the protocol is a constraint labeling process. In our protocol, the space of possible behaviors can be intractable, so the search for acceptable combinations of behaviors involves modifying behaviors on the fly.

In the partial global planning framework [Durfee and Lesser, 1987; Durfee, 1988] local plans are exchanged to recognize larger goals to work toward, and then the actions taken to achieve these goals are reordered to improve group performance. In the context of our new protocol, the information about *what* each problem solver is doing (the results it is trying to develop) is mapped into a hierarchy of goals in order to identify more encompassing results, and then the *how* and *when* of the plans are revised to improve group problem solving. Thus, partial global planning is subsumed by our new protocol. In addition, to avoid the overhead of unconstrained exchanges of the plans, partial global planning insists that the agents have static organizational knowledge, which essentially dictates who each agent should exchange plan information with. Our protocol assumes no such knowledge because agents instead broadly exchange abstract behavioral information and use this to focus more detailed information appropriately. Our approach also differs from recent efforts to more generally classify plan and goal interactions [Decker and Lesser, 1989; von Martial, 1989] because of our extension to agent behaviors, not just plans and goals.

Lansky has developed a formalism for specifying behavioral information about agents, and has investigated its use for multi-agent domains [Lansky, 1985]. Her use of the term "behavioral" and ours differ, in that she considers behavior in terms of the constraints on possible relationships between actions, whereas we see behavior in terms of (possibly underspecified) information about the who, what, when, where, how, and why of activity in the world. These views are not incompatible, and her formalisms could be incorporated into our approach. Davis [Davis, 1981] also has enumerated many of the issues involved in coordinating multiagent behavior. While Davis' treatment is at a conceptual level, we have moved toward an experimental investigation.

Implementation and Experiments

As a preliminary test of our new protocol, we have been investigating the general problem of detecting and resolving resource conflicts. As a representative example of this problem, consider an application where several mobile robots move in common areas and between these areas through shared doorways. The common space, and especially the space in and around the doorways, represents important resources for which the robots might contend.

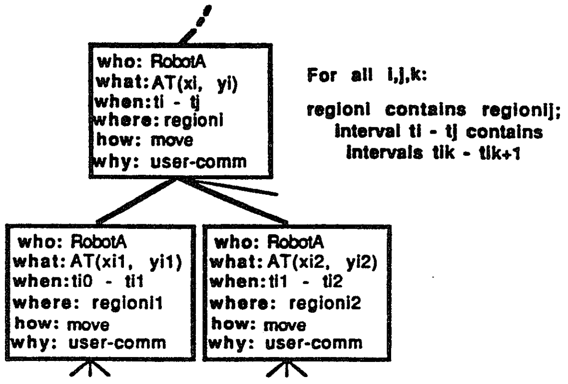


Figure 2: A Robot's Partial Behavior Hierarchy

We have implemented and evaluated our protocol using a simulation of this application. A partial, generalized behavior hierarchy for the application is shown in Figure 2. What sets this type of hierarchy apart from more typical plan/goal decomposition hierarchies is that the representation at one level summarizes the information in the lower levels. For example, in a typical plan/goal decomposition, a desire to move to a goal location might be decomposed into a sequence of intermediate locations to achieve, which might be further decomposed, and so on. In that type of decomposition, an entry at one level only reflects the *expected end result* of the levels below it, and *not* the behavior of the agent in achieving this result. In contrast, the spatial (where) information of an entry in our behavior hierarchy is computed as the smallest region that subsumes the regions below it, just as the temporal (when) interval is the shortest interval that includes all of the intervals below it. As a result, an entry essentially describes where the robot might be (as opposed to where it is going) over an interval of time.

The robots are currently limited to knowing how to move, who they are individually (they have no concept of coalitions), and that commands come from external sources. They plan paths to goal locations by finding a sequence of intermediate locations that avoid obstacles, and represent spatial regions as rectangular areas. They use knowledge about their movement speeds to compute the time needed to move between locations. If they decide that their behaviors might lead to collisions, the agents can search along either the temporal or spatial dimensions to change their behaviors.

Using the Protocol

To implement the protocol for our experiments, we have made several assumptions to promote structured communication and algorithmic convergence. First, we assume that each robot is given a unique authority value, so that robots can decide, when their plans are equally ranked, whose should get priority. Second, we assume that there are distinct intervals during which behavioral information is exchanged and modified. Third, we assume that robots modify their behaviors either by introducing delays in their actions or by choosing alternative routes. This choice is currently based on a user-supplied parameter.

The process begins by each robot broadcasting a message indicating who it is and what goal it wants to pursue. After enough time elapses, the robots build a list of robots sorted by their goal priorities. When robots have equal priority goals, they are sorted by their unique authority values. Because robots develop identical lists, the highest authority (superior) robot knows who it is. This robot broadcasts its most abstract behavioral information to the other (inferior) robots, who in turn broadcast theirs back to acknowledge receipt. Each of the inferior robots compares its local information with what it has received, and checks for conflict. This is simply computed by determining whether it expects to be in an overlapping region at an overlapping time interval. If no conflict is possible, it sends an empty message back to the superior who records that no conflict exists.

If a conflict exists, the inferior can either resolve it or more fully explore it. It can resolve the conflict by either delaying its activities until after the superior has completed its own, or by moving through another spatial region (if possible). Alternatively, it can explore the conflict by comparing its more detailed anticipated behavior (at the next level of the hierarchy) with what it knows of the superior, and then sending information about those aspects of its behavior that might be conflicting back to the superior. The superior compares these with its more detailed behaviors, and sends back only the more detailed behaviors that might interact. This process continues until either the inferior resolves the conflict at some level or it discovers on closer inspection that no conflict will occur. Either way, it sends an empty message back to the superior who records that no conflict exists anymore.

When the superior has heard from all inferiors that no conflicts exists, it passes control to the next robot in the pecking order. The process repeats, and the new superior interacts with the remaining inferiors to remove conflicts. When done, the process repeats again, and so on. At each repetition, the set of inferiors decreases, until no inferiors remain. At this point, all of the conflicts have been removed.

However, there is one wrinkle: When an inferior modifies its behaviors to avoid conflicting with the current superior, it might introduce behaviors that conflict with previous superiors. For this reason, before an inferior sends a message to the current superior indicating that no conflict exists anymore, it first communicates with previous superiors to ensure no new conflict has been introduced. If a new conflict has occurred, the inferior uses the same techniques as before to resolve the conflict with the previous superior, and then checks for new conflicts with the current superior. We can guarantee that we will not enter infinite loops of refinement because of our assumption that a robot changes its behavior either by introducing delays (always postponing actions) or by choosing a route that no superior has chosen. Using temporal modification, the robots could at worst serialize their actions so that no parallelism remains. Fortunately, such worst-case scenarios need seldom occur. Using spatial modification, the robots at worst pursue very roundabout routes.

When the last robot in the pecking order is given control, it knows that all conflicts have been resolved and broadcasts this fact to the others. The robots synchronize and then begin carrying out their planned activities. As each completes its activities, it again broadcasts this fact (along with its next goal's priority) to the others; when all have completed their activities, the protocol is initiated again for the next goals.

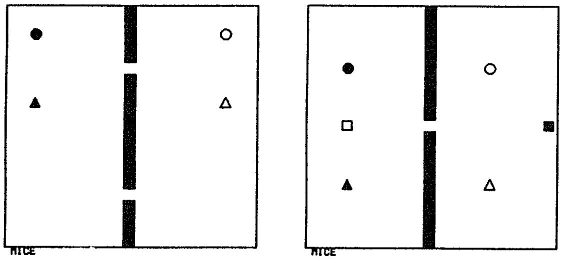
This is just one possible implementation of the protocol, and is primarily intended to test the implications of communicating at different levels of the behavior hierarchy. To maintain computational tractability, to ensure convergence, and to guarantee that conflicts will be resolved ahead of time (which typically requires that some agent ultimately has authority), this implementation assumes a total order on the agents and restricts how agents can modify their behaviors. We are developing more robust implementations in which new agents can join in at any time, agents can recover if one of them fails during the protocol process, and agents can dynamically allocate authority based on current circumstances.

Experiments

To investigate the performance of this implementation of our protocol, we simulated 2 environments in MICE (the Michigan Intelligent Coordination Experiment testbed [Durfee and Montgomery, 1989]). The first environment consists of 2 robots in 2 rooms that are joined by 2 doorways. Each doorway is only large enough for one robot (Figure 3a). The second environment consist of 3 robots in 2 rooms joined by 1 doorway (Figure 3b). In these environments, if the robots attempt to occupy the same location at the same time, they "bounce off" each other and return to their previous locations. Robots require 1 simulated time unit to move in any direction.

Each robot has its own distinct blackboard-based architecture, implemented in GBB [Corkill *et al.*, 1986]. The robots are given appropriate knowledge sources for planning paths to goal positions, for building behaviors based on these plans, for exchanging information at various levels of their behavior hierarchy, and for modifying behaviors along temporal and spatial dimensions to avoid potential collisions.

Communication delay between robots is 1 simulated time unit. To measure the protocol's communication overhead, we record the number of behaviors exchanged. Similarly, to measure the computa-



(a) (b)

In (a), RobotA (solid circle) is trying to get to the location marked with a hollow circle, RobotB (solid triangle) to the hollow triangle. In (b), RobotC (solid square) is also trying to get to the location marked with the hollow square.

Figure 3: Experimental Scenarios in MICE

tional overhead, we use MICE's capabilities for charging robots for time spent reasoning in the following way. As it compares behavioral information, a robot keeps track of how many comparisons between local and received behaviors it makes. For every n comparisons (where n is a user-modifiable parameter), it tells MICE to increment its clock by 1 simulated time unit. We also measure the simulated time spent carrying out the activities (from the earliest time an agent begins moving to the latest time an agent arrives at its goal destination), and the overall time to achieve all of the goals. For simplicity, we only give each robot one goal.

In our first set of experiments, consider the 2 robot case (Figure 3a). Unhindered, each robot requires 22 time units to move from start to goal locations. The results of our experiments are given in Table 1. Without any exchange of behavioral information (experiment 1), the robots spend a small amount of time initializing their plans. The time that robots spend moving to their goal destinations, however, is infinite, because the robots perpetually collide at the doorway.

When we allow the exchange of behavioral information, the robots can successfully make it to their goal destinations. First, consider the case where robots can make 5 comparisons of behaviors per simulated time unit. In experiment 2, the robots resolve potential conflicts between their most abstract behaviors. The overall expected regions of movement for the robots overlap at the doorway, so RobotB (the lower authority robot) changes the timing of its activities so that it does not begin moving anywhere in its overall region until RobotA expects to have finished. In essence, the robots have serialized their activities (with 1 extra time unit imposed between their activities to ensure no overlap).

In experiment 3, we make the robots resolve potential conflicts at an intermediate level. At this level, each has broken its overall behavior into 3 parts: to get in front of the door, to get to the other side of the door, and to get to the goal location. In the initial, abstract behavior exchange, they detect the potential conflict, and so RobotB requests intermediate information from RobotA. At first, RobotB changes its activities so that it will not begin its first intermediate behavior (going

Table 1: Experiment Summary.

Ex	En	M	Cps	Rs	Ex	TP	TA	Done
1	a	-	-	none	0	5	∞	∞
2	a	t	5	abs	2	10	45	55
3	a	t	5	int	9	16	35	51
4	a	t	5	det	16	77	24	101
5	a	t	50	abs	2	10	45	55
6	a	t	50	int	9	13	35	48
7	a	t	50	det	16	23	24	47
8	a	s	5	abs	2	10	32	42
9	a	s	5	int	8	14	32	46
10	a	s	5	det	13	47	32	79
11	a	s	50	abs	2	10	32	42
12	a	s	50	int	8	13	32	45
13	a	s	50	det	13	17	32	49
14	a′	-	-	none	0	5	22	27
15	a′	t	5	abs	2	10	43	53
16	a′	t	5	int	8	15	33	48
17	a′	t	5	det	7	33	22	55
18	a′	t	50	abs	2	10	43	53
19	a′	t	50	int	8	13	33	46
20	a′	t	50	det	7	15	22	37
21	a′	s	5	abs	2	10	32	42
22	a′	s	5	int	7	14	32	46
23	a′	s	5	det	7	33	22	55
24	a′	s	50	abs	2	10	32	42
25	a′	s	50	int	7	13	32	45
26	a′	s	50	det	7	15	22	37
27	b	-	-	none	0	5	∞	∞
28	b	t	5	abs	6	14	63	77
29	b	t	5	int	23	27	54	81
30	b	t	5	det	68	178	25	203
31	b	t	20	abs	6	14	63	77
32	b	t	20	int	23	21	54	75
33	b	t	20	det	68	65	25	90
34	b	t	50	abs	6	14	63	77
35	b	t	50	int	23	21	54	75
36	b	t	50	det	68	44	25	69

Abbreviations

Ex:	Experiment number
En:	Environment
M:	Modification (temporal or spatial)
Cps:	Comparisons per simulated time unit
Res:	Level where behavior conflicts are resolved: ABStract, INTermediate, or DETailed.
Ex:	Number of behaviors exchanged
TP:	Time for planning non-conflicting behaviors
TA:	Elapsed time for robot activities
Done:	Time at which all robot goals achieved

to the door) until RobotA begins its second behavior (entering the door). However, because RobotA's second behavior includes being in front of the door, and because this overlaps with the region of RobotB's first behavior, RobotB further delays its first behavior until RobotA has actually moved through the door. RobotB propagates the effects of this delay on the rest of its behaviors, and because subsequent time intervals are modified, no further conflicts between the robots' behaviors exist. The data shows that resolving potential conflicts at this level incurs more communication and further delays the start of the plans (compared to experiment 2), but results in more movement parallelism in the robots' actions. As a result, the overall time to achieve the goals is lower.

In experiment 4, the robots exchange abstract, then intermediate, and finally fully detailed behaviors (down to locations they will occupy at specific times). Based on this information, RobotB again modifies the intervals of its behaviors, but this time recognizes that it need not delay from the very start. Instead, it determines that trying to get in front of the door is what conflicts with RobotA, and delays this detailed behavior. Propagating this change removes all other conflicts. The movements of the robots in this case have substantial parallelism so the time spent carrying out the plans is reduced, but the amount of communication and the time spent resolving the conflicting plans is much greater. In fact, the overall time is much worse than the previous 2 experiments.

Experiments 5-7 are the same as 2-4, except the number of comparisons per simulated time unit is 50. This serves to make computation cheaper with respect to movement, but does not change the quality of coordination. The implication of this change is that exerting computational effort to increase parallel movement is more worthwhile, so resolving conflicts at more detailed levels is better.

In experiments 8-13, RobotB resolves the conflict along the spatial dimension by moving through the further doorway. This results in identical plans regardless of the level at which the conflicts are resolved, because once RobotB plans its new path, the robots' behaviors no longer interact at any level of abstraction. To save on overhead, resolving conflicts at less detailed levels is thus better.

Experiments 14-26 are the same as 1-13, but now RobotA starts slightly closer to the door (environment a′). As a result, the robots' actions will not conflict, and the best results are achieved without any coordination (experiment 14). When the robots coordinate at the abstract and intermediate levels, they cannot ensure that no conflict will occur, so they modify their plans either temporally (experiments 15, 16, 18, 19) or spatially (experiments 21, 22, 24, 25). At the detailed level, the robots recognize that no resolution is needed, but only after substantial investment of effort. Sometimes this effort is worthwhile (experiments 20, 26) while at other times it is not (experiments 17, 23).

Finally, experiments 27-36 are based on the 3 robot environment (Figure 3b). Without any conflict avoidance, the robots once again perpetually collide at the doorway (experiment 27). The addition of a third robot can dramatically increase the amount of information exchanged at the detailed and intermediate levels. As a result, when robots can do only 5 comparisons per simulated time unit (experiments 28-30), resolving conflicts at the abstract level is better than the intermediate level which is better than the detailed level. As the number of comparisons per simulated time unit increases (experiments 31-36), first the intermediate and then the detailed levels are best. Note that the time spent resolving conflicts at more detailed levels is always greater regardless of comparisons allowed per time unit, because of the communication delay. Several messages might be in transit at the same time, however, so the time spent resolving conflicts can be less than the number of behaviors exchanged (such as in experiments 35 and 36).

In summary, these experiments highlight the fact that different circumstances demand different amounts of communication and coordination. Sometimes not communicating or coordinating is best (experiment 14), but at other times such a strategy can be catastrophic (experiments 1 and 27). Sometimes it is more cost effective to coordinate at a high-level, because it involves less overhead and results in relatively acceptable coordination (such as in experiments 8, 11, 21, and 28). Coordinating at detailed levels has advantages in situations where the chances for parallelism are greater (such as in experiments 7 and 36), or where the robots can only ensure that they will not conflict when the examine their behaviors in detail (such as in experiment 26). And coordinating at an intermediate level is sometimes the best policy (such as in experiments 3 and 32). In short, conflicts should be resolved at different levels in different cases. Unlike previous protocols that work at a single level of detail, our new protocol allows the robots to coordinate at different levels flexibly.

The experiments also point toward further research directions. Issues in mediating between different perspectives (which our current implementation avoids by imposing a total ordering on agents) and in deciding along what dimensions and at what level to resolve conflicts (recall our current implementation depends on user-supplied parameters) are focuses of our current research. In addition, using other dimensions for resolving conflicts (or promoting cooperation) is important, and we are exploring issues in getting agents to adopt each other's (or joint) goals or having them choose methods that achieve their goals in non-conflicting ways. The challenge in this research is in exploiting the richness of the behavior hierarchy while heuristically restricting the search among behaviors to maintain computational tractability.

Conclusions

As we have demonstrated experimentally, our protocol allows a form of coordination that is well suited to domains where interacting agents do not know, *a priori*, with whom they will interact. This is accomplished through a dialogue between the agents in which they are free to exchange information at different levels of abstraction about their anticipated behavior. In essence, this dialogue is a rudimentary form of negotiation between the agents: Although our specific implementation identified which agent involved in a conflict should modify its behavior, our protocol admits to more flexible (and arguably more computationally expensive) encounters, where each agent involved in a conflict moves along some of its behavioral dimensions until a compromise is found that eliminates the conflict and yet allows each agent to retain important behavioral attributes. Our work thus paves the way for a new investigation into intelligent negotiation.

Another observation that we have made is that the dimensions of our behavior hierarchy correspond to problem decomposition directions found in human organizations [Malone, 1987]. If we decompose a behavior along the *what* dimension, for example, we are decomposing based on results (or products) of behavior, which leads to a *product hierarchy*. Similarly, decomposing along the *how* dimension leads to a type of *functional hierarchy*. From other dimensions, we see personnel hierarchies, temporal hierarchies, spatial hierarchies, and even motivational hierarchies. We are intrigued by the possible relationship between our approach and organizational theory, and indeed are working toward experiments where groups of robots team up and represent their overall behavior in the behavior hierarchy. As a result, groups (or organizations) of individuals are viewed as single entities, and can negotiate as a unit; yet, by traversing the hierarchy the behaviors of individuals are still represented. Based on these insights, we are currently exploring how to integrate organizational and planning theories within our single protocol.

References

[Ashley, 1988] Kevin D. Ashley. *Modelling Legal Argument: Reasoning with Cases and Hypotheticals.* PhD thesis, University of Massachusetts, February 1988.

[Cammarata *et al.*, 1983] Stephanie Cammarata, David McArthur, and Randall Steeb. Strategies of cooperation in distributed problem solving. In *Proceedings of the Eighth International Joint Conference on Artificial Intelligence*, pages 767–770, Karlsruhe, Federal Republic of Germany, August 1983.

[Conry *et al.*, 1988] Susan E. Conry, Robert A. Meyer, and Victor R. Lesser. Multistage negotiation in distributed planning. In Alan H. Bond and Les Gasser, editors, *Readings in Distributed Artificial Intelligence*, pages 367–384. Morgan Kaufman, 1988.

[Corkill and Lesser, 1983] Daniel D. Corkill and Victor R. Lesser. 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*, pages 748–756, Karlsruhe, Federal Republic of Germany, August 1983.

[Corkill *et al.*, 1986] Daniel D. Corkill, Kevin Q. Gallagher, and Kelly E. Murray. GBB: A generic blackboard development system. In *Proceedings of the National Conference on Artificial Intelligence*, pages 1008–1014, Philadelphia, Pennsylvania, August 1986.

[Davis and Smith, 1983] Randall Davis and Reid G. Smith. Negotiation as a metaphor for distributed problem solving. *Artificial Intelligence*, pages 63–109, 1983.

[Davis, 1981] Randall Davis. A model for planning in a multiagent environment: Steps toward principles for teamwork. AI Working Paper 217, Artificial Intelligence Laboratory, Massachusetts Institute of Technology, Cambridge, Massachusetts 02139, June 1981.

[Decker and Lesser, 1989] Keith S. Decker and Victor R. Lesser. Some initial thoughts on a generic architecture for cdps network control. In *Proceedings of the 1989 Distributed AI Workshop*, pages 73–94, September 1989.

[Durfee and Lesser, 1987] Edmund H. Durfee and Victor R. Lesser. Using partial global plans to coordinate distributed problem solvers. In *Proceedings of the Tenth International Joint Conference on Artificial Intelligence*, pages 875–883, Milan, Italy, August 1987.

[Durfee and Montgomery, 1989] Edmund H. Durfee and Thomas A. Montgomery. MICE: A flexible testbed for intelligent coordination experiments. In *Proceedings of the 1989 Distributed AI Workshop*, pages 25–40, September 1989.

[Durfee *et al.*, 1987] Edmund H. Durfee, Victor R. Lesser, and Daniel D. Corkill. Coherent cooperation among communicating problem solvers. *IEEE Transactions on Computers*, C-36(11):1275–1291, November 1987.

[Durfee, 1988] Edmund H. Durfee. *Coordination of Distributed Problem Solvers.* Kluwer Academic Publishers, 1988.

[Lansky, 1985] Amy L. Lansky. Behavioral specification and planning for multiagent domains. Technical Report 360, SRI International, Menlo Park CA, 1985.

[Malone, 1987] Thomas W. Malone. Modeling coordination in organizations and markets. *Management Science*, 33(10):1317–1332, 1987.

[Steeb *et al.*, 1986] R. Steeb, S. Cammarata, S. Narain, J. Rothenburg, and W. Giarla. Cooperative intelligence for remotely piloted vehicle fleet control. Technical Report R-3408-ARPA, Rand Corporation, October 1986.

[Steeb *et al.*, 1988] R. Steeb, S. Cammarata, F. Hayes-Roth, P. Thorndyke, and R. Wesson. Architectures for distributed air-traffic control. In Alan H. Bond and Les Gasser, editors, *Readings in Distributed Artificial Intelligence*, pages 90–101. Morgan Kaufman, 1988.

[von Martial, 1989] Frank von Martial. Multiagent plan relationships. In *Proceedings of the 1989 Distributed AI Workshop*, pages 59–72, September 1989.