In
Yves DEMAZEAU and Jean-Pierre MÜLLER (Eds.)
Decentralized Artificial Intelligence, Volume 2
Elsevier Science Publishers B. V. (North-Holland), 1991

GROUP ABILITY AND STRUCTURE

Munindar P. Singh

Dept of Computer Sciences and Artificial Intelligence Lab
University of Texas, Austin, TX 78712-1188, USA
MCC, Austin, TX 78759, USA

Groups of intelligent agents are the object of study in several subareas of AI, notably, autonomous agents, multiagent planning and action, discourse understanding, and cooperative work. I present a formal theory of the ability of a group of agents that analyzes ability directly in terms of the individual skills of the group's members, and its internal structure or organization. Since this theory does not reduce "*know-how*," the knowledge of skills, to "*know-that*," the knowledge of facts, it avoids many of the problems that plague traditional theories of knowledge and action. In using the notion of "strategies" (as described in the paper), this theory presents a realistic two-layered view of action by, and interaction among, groups of agents. This theory has important ramifications in the study of the intentions and beliefs of groups of agents, and of the combination of their expertise.

1 Introduction

Several interesting and important subareas of AI involve *groups* of intelligent agents, with varying degrees of autonomy, who share a fragment of the world and affect one another through their actions. These subareas include autonomous agents, multiagent planning and action, discourse understanding, and cooperative work [4, 5, 10, 12, 15, 16, 17]. I present a formal theory of the objective *ability* or *know-how* of a group of agents that accounts for its internal *structure*. Since even individual agents can be treated as groups of subagents [8], this theory also applies in areas such as real-time AI systems and robotics. It also helps clarify the question of the joint intentions of a group of agents [4, 12, 24, 26]. A longer term research interest that this paper is a step towards is to formulate a logically well-founded methodology for the design of systems of intelligent agents.

Traditional formal theories of knowledge and action [17, 20, 21] stress the conception of knowledge corresponding to *know-that*, or the knowledge of facts, rather than *know-how*, the knowledge of skills. It is assumed that *know-how* can be reduced to *know-that*. In §3, I briefly review the arguments of [27] that *know-how* is important, and cannot be so reduced. In §2, I argue that the internal structure of groups, also neglected by traditional theories, is important. In §4, I present a formal model that I use for successively better definitions of the *know-how* of a group in §4.1, §5 and §6. This definition involves the notion of abstract strategies (defined in §5), and the ways in which they may be reactively followed. In §6, I argue that the structure of a group can be best captured in terms of the "strategic" and "reactive" interactions of its members. In §6.1, I briefly consider an example where a theory such as the one proposed here may be applied. Finally, in §7, I list some important formal consequences of the theory of this paper.

This paper largely focuses on the clarification and formalization of our pretheoretic intuitions about the structure and ability of groups; it is successful to the extent that it captures those intuitions, and provides a framework for further technical development. Briefly, this paper seeks to identify the "external" or objective requirements for the ascription of *know how* to groups of intelligent agents. By "external" I mean the requirements that are determined by the behavior of the agents, rather some claims about what their internal structure must be like (e.g., I shall not restrict myself to agents who have symbolic knowledge bases of facts, and reason by using a general purpose inference engine). A consequence of this approach is that this theory, unlike [4, 12], does not require a group of agents to have mutual beliefs (or common knowledge) to cooperate. Common knowledge is impossible to achieve in most realistic scenarios; e.g., where communication is not (known to be) guaranteed with only a finite delay [9, 13].

2 Groups

Traditional theories of multiagent intentions and action tend to ignore the *structure* of the group of agents being considered [4, 8, 12, 24], and assume that agents are equally capable and perfectly cooperative. However, the knowledge and skills of a group depend greatly on its structure, and on the knowledge and skills of its members. E.g., consider a football team that includes a pass-thrower (i.e., a quarterback), and a pass-receiver, trying to move the ball several yards. This team *knows how* to make this play (against some opponent, say) because its members have the required skills: the quarterback can throw well, the receiver can catch well, and so on. The players' *know-hows* combine to yield the *know-how* of the team. But the internal structure of the team also matters. Indeed, if the players were not to cooperate, or if the quarterback and receiver were to exchange their positions, the team may not *know how* to make the play even though individual abilities would remain unchanged. Analogously, the team may *know that* it is best to throw the ball in some situation because the quarterback does, and may not if he does not, even if the receiver does. The individual players' knowledge and skills "surface," and influence the knowledge and skills of the whole group, only because they play in specific positions and cooperate with each other; i.e., they interact so as to bring out each others' skills.

The structure of a group of agents emerges from the collective actions performed by its members. In providing a formal action-based theory of the structure of a group of agents, it is methodologically advantageous to consider not just the actions that are actually done, but all actions that may potentially be done. In other words, we need to consider the ability or *know-how* of a group of agents. I propose to consider the abilities of groups of agents, and define the structure of a group, in terms of how its members act and interact. This approach allows us to model externally the commitments of members to each others' goals; i.e., model groups whose members are only partially cooperative, or are even adversaries. E.g., two opposing football teams interact such that the ability of the two teams together to score a goal is much reduced than that of any one team by itself (which could just walk straight with the ball to the goal).

Two observations are in order here. (1) A group (e.g., a sports team) may be considered as a single unstructured monolithic agent from without; i.e., groups are "Hobbesian corporate persons," in Hamblin's term [14, pp. 60, 240]. The *know-how* of a group of agents is the same kind of thing as the *know-how* of individual agents: the only difference is one of extent: one would typically expect a cooperative group to have "greater" *know-how* than any of its proper subgroups. One consequence of this is that groups (e.g., business corporations) may recursively contain other groups. (2) Even though groups (e.g., teams or corporations) seem monolithic from without, they are not homogeneous: the members of a group may have not just different abilities, but also make different contributions to the *know-how* of the group, depending on the manner in which they participate in it.

I plan to define the *know-how* of a group in terms of its structure; correspondingly, restrictions on the combination of the members' *know-hows* would give us a handle on the structure of the group in the most significant way of all—its ability to act.

3 Know-How

We can attribute intelligence only to agents who can act effectively and skillfully. Indeed, if agents did not act at all, it would be difficult even to assign any knowledge of facts to them, except by arbitrarily interpreting their internal states [23]; e.g., we can say that a sunflower *knows that* the sun is in a certain direction only because it *turns* to point that way. A book or a library is not intelligent because neither can act. This much is accepted in traditional AI; however, traditional AI theories assume that a knowledge base of facts or beliefs coupled with an interpreter can be used to model all interesting forms of intelligence. This approach leads to the view of *know-how* described in §3.1. Unfortunately, this approach, while formally elegant, embodies many unrealistic assumptions; problems with it are outlined in §3.2.

3.1 The Traditional Theory

Let K_{how} and K_{that} be predicates. Traditionally, an agent, x, *knows how* to achieve A by doing an action (description), P, iff x *knows that* P achieves A, and x can execute P [20, 21]. Quantifying over action descriptions, and in the notation of this paper (with M the model, s the current "situation," and 'result' a predicate giving the result of executing a plan in a situation), we obtain [20, p. 347]:

- $M \models_s K_{how}(x, A)$ iff
  $(\exists P : M \models_s \text{can-execute}(x, P) \wedge M \models_s K_{that}(x, \text{result}(P, s) \rightarrow A))$

The agent, x, proves that P will yield A, and always explicitly represents both P and all the knowledge required to instantiate and execute it. The theory of [11] is in the same spirit, since it requires the agents in a group to have mutual beliefs about their "shared plans," and about how they may execute them.

3.2 Problems with the Traditional Reduction

There are several problems with this reduction; they become even more urgent when group *know-how* is considered. The major shortcomings are considered below.

1. **Multiagent systems:**

   Traditional theories are based on plans, and typically consider only single agent plans: thus they cannot account for the *know-how* of a group of agents. However, organizations *know how* to do certain things, but may not have any explicit plan [15]. And, even in organizations, the actual actions are taken by individual agents, whose knowledge may not be easily combined. A reduction to what *all* members know would be too strong, and that to what just *some* members know too weak.

2. **Reactivity:**

   The reduction stresses explicitly represented action descriptions and plans. However, much *know-how* is reactive; e.g., a football team may *know how* to score a goal, but would not have a plan that will guarantee it, much less one that was explicitly represented and interpreted. It would not, and could not, have a conditional plan either, since all eventualities (e.g., moves of the opponent team, wind drag on the ball) cannot be anticipated.

3. **Basic actions and skills:**

   Traditional accounts assign the same basic actions to all agents in all situations. Expertise can vary only because of differences in the knowledge of facts (e.g., "Bill's telephone number"), not because of differences in abilities: given the same facts, all agents have the same *know-how*. However, groups are useful and successful, at least in part, because members may have differing abilities. I submit that 'basic' actions are actions done by an agent with a single *choice,* and depend on its physical capabilities and skills. The effect of an action depends on the situation it is done in, and on the other events that take place with it. The agents' procedural skills, habits, and practical wisdom in selecting their actions jointly determine their *know-how.*

4. **Complexity of decision making:**

   Real-life agents have to act rapidly on the basis of incomplete information, and lack the time and knowledge required to form complete plans before acting. Further, they cannot choose optimal actions in real-time, so the assumption of pure reactivity is not realistic. Abstract "strategies" (discussed in §5) simplify reactive decision making by letting an agent have partial plans that can be effectively followed in real-time.

3.3 The New Analysis

A group, G, *knows how* to achieve A, in loose parlance to "do" A, if it is usually able to bring about the conditions for A through its actions. The world may change rapidly and unpredictably, but G is able to *force* the situation appropriately. The group has a limited knowledge of its rapidly changing environment. It has too little time to decide on an optimal course of actions, and can succeed only if it has the required physical resources and is appropriately attuned to its environment. While this "situated" theory does not use plans, it can accommodate the richer notion of strategies (see §5). Therefore, it contrasts both with situated theories of *know-that* [22], and informal, and exclusively reactive, accounts of action [1].

4 The Formal Model

The formal model is based on possible worlds.[1] Each possible *world* has a branching *history* of *times*. At each time, environmental processes and agents' actions occur. Agents influence the future by acting, but the outcome also depends on other events. The actions an agent can do can differ at each time—this allows for learning and forgetting, acquiring and losing physical capabilities, and changes in the environment.

Let $M = \langle F, N \rangle$ be an intensional model, where $F = \langle \mathbf{W}, \mathbf{T}, \mathbf{I}, \mathbf{A}, \mathbf{U} \rangle$ is a frame, and $N = \langle [\![]\!], \mathbf{B} \rangle$ an interpretation. Here $\mathbf{W}$ is a set of possible worlds; $\mathbf{T}$ is a set of possible times; $\mathbf{I}$ is the class of domains of the various possible worlds; $\mathbf{A}$, the class of agents in different possible worlds, is a subclass of $\mathbf{I}$; $\mathbf{U}$ is the class of basic actions; as described below, $[\![\,]\!]$ assigns intensions to predicates and actions. $\mathbf{B}$ is the class of functions assigning basic actions to the agents at different worlds and times. Each world $w \in \mathbf{W}$ has exactly one history, constructed from the times in $\mathbf{T}$. Histories are sets of times, partially ordered by $<$. They branch into the future; each linear closed suborder of a history is called a scenario. The sets of times in the history of each world are disjoint. "Times" could be instants or periods. A world and time are a "situation."

A *scenario* at a world and time is any maximal set of times containing the given time, and all times that are in a particular future of it; i.e., a scenario is any single branch of the history of the world that begins at the given time, and contains all times in some linear subrelation of $<$. I define a *skeletal scenario* as an eternal linear sequence of times in the future of the given time; i.e., SS at w, t is any sequence: $\langle t = t_0, t_1, \ldots \rangle$, where $(\forall i : i \geq 0 \rightarrow t_i < t_{i+1})$ (linearity) and $(\forall i, t' : t' > t_i \rightarrow (\exists j : t' \not> t_j))$ (eternity). Now, a scenario, S, for w, t may be defined as the "linear closure" of some skeletal scenario at w, t. Formally, S, relative to some SS, is the minimal set that satisfies the following conditions:

Eternity: $SS \subseteq S$

Linear Closure: $(\forall t'' : t'' \in S \rightarrow (\forall t' : t_0 < t' < t'' \rightarrow t' \in S))$

[1]This formal model resembles those of McDermott [19] and Shoham [25]. However, those accounts emphasize a language that might be used for representing and reasoning about temporal information. The stress here is on giving the formal definition for *know-how*. A significant difference is that I consider the basic actions of agents explicitly, and describe how the reactive behavior of agents may be modeled. I also consider strategies as abstract specifications of the behavior of individuals and groups (see §5).

This definition applies to arbitrary histories, not just discrete ones. $\mathbf{S}_{w,t}$ is the class of all scenarios at world w and time t. $\langle S, t, t' \rangle$ is a subscenario of S from t to t', inclusive.

Basic actions may have different durations relative to one another in different scenarios, even those that begin at the same world and time. The intension of a predicate is, for each world and time, the set of the relations that model it; that of an action is, for each agent x, the set of subscenarios in the model in which an instance of it is done (from start to finish) by x; e.g., $\langle S, t, t' \rangle \in [\![a]\!]^x$ means that agent x does action a in the subscenario of S from time t to t'. An agent (or subgroup) could do several actions at once; since the $[\![]\!]$ is a part of the model, the actions that are done simultaneously are automatically mutually compatible. I assume that $[\![]\!]$ respects $\mathbf{B}$; i.e., $a \in \mathbf{B}_{w,t}(x)$. Although I will not pursue this here, restrictions on $[\![]\!]$ can be used to express the limitations of agents, and the ways in which their actions may interfere with those of others; e.g., x cannot pick up three glasses at once and at most one person can enter an elevator at a time. The habits of agents can be similarly modeled; e.g., x always brakes before turning.

4.1 Action Sequences and Trees

It is useful for the definitions that follow to extend the definition of intension of an action in the following ways. Let $G = \{x_1, \ldots, x_n\}$ be a group. Let '$\text{seq}_i = \langle a_0, \ldots, a_{m-1} \rangle$' be a sequence of actions of member x_i. Then $[\![\text{seq}_i]\!] = \{\langle S, t, t' \rangle | (\exists t_0 \leq \ldots \leq t_m : t = t_0 \wedge t' = t_m \wedge (\forall j : j \in [1 \ldots m] \rightarrow \langle S, t_{j-1}, t_j \rangle \in [\![a_{j-1}]\!]^{x_i}))\}$; i.e., the set of subscenarios over which it is done. A sequence for a group is a set of member sequences executed in parallel; formally, $\text{seq} = \langle \text{seq}_1 \parallel \ldots \parallel \text{seq}_n \rangle$. The intension of 'seq' is the set of all subscenarios in which the member sequences are executed starting together (and ending in any order). $[\![\text{seq}]\!] = \{\langle S, t, t' \rangle | (\exists e_0, \ldots e_{m-1} : \{e_0, \ldots e_{m-1}\} = \{0, \ldots, m-1\} \wedge t_{e_0} \leq \ldots \leq t_{e_{m-1}} \wedge (\forall j : j \in [1 \ldots m] \rightarrow \langle S, t, t_{e_j} \rangle \in [\![\text{seq}_j]\!]))\}$. $\langle S, t, t' \rangle \in [\![\text{seq}]\!]$ means that 'seq' starts at t and ends at t'; $[\![\text{seq}]\!] = \emptyset$ means that a subset of the member sequences is inconsistent. Sets of simultaneous actions can be treated as sets of sequences of length unity.

The formal language here is the predicate calculus, augmented with a two-place predicate, K_{how}(agent, formula). Here 'agent' could itself be a group. The semantics is given relative to intensional models: it is standard for the predicate calculus; the predicate K_{how} is considered below. Some auxiliary predicates are defined where needed.

In order to put a bound on the number of actions required by a group to achieve something for it to have the corresponding *know-how*, it is useful to define *know-how* relative to a 'tree' of actions. A 'tree' of actions consists of a set of action-sets (one action-set per member of the given group), and a set of subtrees. The set of action-sets is called the 'root' of the tree in the sequel; it is convenient to let 'root' be a sequence as defined in the previous paragraph, rather than just a set of action-sets. The idea is that the group does 'root,' i.e., the initial action-sets first (each member doing his part), and then picks out one of the available subtrees to pursue further. A group *knows how* to achieve A relative to a tree of actions iff on *all* scenarios where the root of the tree is done, either A occurs or the group can choose one of the subtrees of the tree to pursue, and thereby achieve A. The definition requires A to occur on all scenarios where the root is done. The group gets to "choose" one subtree after the root has been done. This is to allow the choice to be based depending on how the group's environment has evolved.

However, modulo this choice, the group must achieve A by forcing it to occur—i.e., the group cannot take advantage of luck. This definition requires the root of the tree to be done entirely, but allows A to occur fortuitously, before it has been completed. This definition is best applied to models that consist only of "normal" scenarios (see [27] for a discussion). It is important to note that the tree need not be represented jointly or severally by the members of the group, and is intended not to be so represented. The tree just encodes for the group of agents the selection function it may be deemed to be using in picking out their actions at each stage. The agents just happen to behave according to this function—they don't need to symbolically represent or compute it any special part of their architecture.

Formally, a *tree* is either (1) $\emptyset$, the empty tree or (2) $\langle$root, $\langle$subtrees$\rangle\rangle$, where 'root' is a non-empty sequence of actions (as described above), and 'subtrees' is a non-empty set of trees. In the sequel, I assume that the variable 'tree' refers to a tree of this form.

- $\langle M, \emptyset\rangle \models_{w,t} K_{how}(G, A)$ iff $M \models_{w,t} A$
- $\langle M, \text{tree}\rangle \models_{w,t} K_{how}(G, A)$ iff $(\forall S \in \mathbf{S}_{w,t} \wedge (\exists t' : \langle S, t, t'\rangle \in [\![\text{root}]\!]) \rightarrow (\exists t' : \langle S, t, t'\rangle \in [\![\text{root}]\!] \wedge [(\exists t'' : t < t'' \leq t' \wedge M \models_{w,t''} A) \vee (\exists\, \text{sub} \in \text{tree.subtrees} : \langle M, \text{sub}\rangle \models_{w,t'} K_{how}(G, A))]))$

One could possibly say that a group *knows how* to achieve a condition, if it can achieve it relative to some tree of actions that it can do. However, this definition seems to require that the members of a group be able to coordinate their activities at the level of their basic actions (in order for each to do its part of the tree, in particular to "select" the same subtree at once); otherwise, it would attribute *know-how* to a group just because the right choices are open to it, even though its members would not be able to make them. Because of this reason, this approach yields an extreme definition of *know-how*. It can certainly be applied purely externally on groups that are somewhat rigidly structured, but then it de-emphasizes the inherent autonomy of complex intelligent agents. However, it is not acceptable to require that groups of agents explicitly represent and interpret shared plans. This motivates us to look at abstract strategies that need not be explicitly represented, but which can characterize the coordinated behavior of groups whose members are largely autonomous.

5 Group Strategies

Strategies abstractly characterize the behavior of agents and, as I show in §6, the structure of groups of agents (in terms of their interactions). In more detail, the behavior and habits of agents are characterized by how they do sequences of actions (derived from the tree characterizing their selection) in following their strategies. When strategies are construed as abstract specifications, we can be agnostic about whether they are hard-wired into the agents who belong to the given group, or learned or obtained through planning by one or more of its members. They need not even be explicitly represented or computed before action begins. It is tempting to say that G *knows how* to achieve A, if it can achieve A whenever it so "intends." Strategies are a simple, albeit incomplete, way of analyzing intentions (see [26] for a formalization of this idea in the same general framework as this paper). I now let each group have a strategy that it follows in the current situation. Intuitively, a group *knows how* to achieve A if it has a strategy Y such that its members possess the skills required to follow Y in such a way as to achieve A. Thus *know-how* is partitioned into two components: the "ability" to have satisfactory strategies, and the "ability" to follow them.

Let Y be a strategy of group G; 'current(Y)' the part of Y now up for execution; and 'rest(Y)' the part of Y remaining after 'current(Y)' has been done. I will define strategies, 'current,' 'rest' and the *strategy-relative intension* of a tree (i.e., $[\![]\!]_Y$) shortly, but first I formalize *know-how* using the auxiliary definition of *know-how* relative to a strategy.

- $\langle M, Y\rangle \models_{w,t} K_{how}(G, A)$ iff $(\exists$ tree: $(\exists S, t' : \langle S, t, t'\rangle \in [\![\text{tree}]\!]_{\text{current}(Y)}) \wedge \langle M, \text{tree}\rangle \models_{w,t} K_{how}(G, A) \vee (\forall S : S \in \mathbf{S}_{w,t} \wedge (\exists t' : \langle S, t, t'\rangle \in [\![\text{tree}]\!]_{\text{current}(Y)}) \rightarrow (\exists t' : \langle S, t, t'\rangle \in [\![\text{tree}]\!]_{\text{current}(Y)} \wedge \langle M, \text{rest}(Y)\rangle \models_{w,t'} K_{how}(G, A))))$

In English, this definition says that a group, G *knows how* to achieve A iff there is a tree of actions for it such that it can achieve the 'current' part of its strategy by following that tree, and that on all scenarios where it does so it either achieves A or can continue with the 'rest' of its strategy (and *knows how* to achieve A relative to that). The agents act in parallel, but the entire group has only one strategy at this point; i.e., the activities of the group as a whole are abstracted, and its members always follow the current part of the group strategy. Now satisfaction *simpliciter* may be defined as follows:

- $M \models_{w,t} K_{how}(G, A)$ iff $(\exists Y : \langle M, Y\rangle \models_{w,t} K_{how}(G, A)) \wedge (\exists S \in \mathbf{S}_{w,t} \wedge (\forall t' \in S : M \not\models_{w,t'} A))$

Here the second conjunct excludes inevitable conditions. This definition meets many of our desiderata. It allows us to incorporate abstract specifications of the group's actions from the designer's point of view, and yet is able to admit the purely reactive aspects of their behavior. No requirements are imposed as to what the agents must be able to represent explicitly or symbolically; thus it applies to the wider range of intelligent agency that we wished to consider.

As a first approximation, I define a strategy, Y, recursively as follows.

0. skip: the empty strategy
1. do(A): a condition to be achieved
2. wait(A): a condition to be awaited, (for synchronization with other events)
3. $Y_1; Y_2$: a sequence of strategies
4. if A then Y_1 else Y_2: a conditional strategy
5. while A do Y_1: a conditional loop

The 'current' part of a strategy depends on the current situation. For case (0), 'current(Y)' is 'skip'; for cases (1) and (2), it is 'Y' itself; for (3), it is 'current(Y_1)'; for (4), it is 'current(Y_1)' if A holds in the current situation, and 'current(Y_2)' otherwise; for (5) it is 'current(Y_1),' if A holds (in the current situation), and 'skip' otherwise. The 'rest' of a strategy is what is left after the current part is performed. For cases (0), (1) and (2) 'rest(Y)' is 'skip'; for (3), it is 'rest(Y_1); Y_2'; for (4), it is 'rest(Y_1),' if A holds and 'rest(Y_2)' otherwise; for (5), it is 'rest(Y_1); while A do Y_1,' if A holds, and 'skip' otherwise. It can easily be seen that relative to the standard semantics for the constructs introduced above (e.g., see [18]), 'Y' is equivalent to 'current(Y); rest(Y).' Some other obvious, but useful consequences of these definitions are:

Lemma 1: 'current(Y)' is always of the form 'skip' or 'do(A)' or 'wait(A)'

Lemma 2: 'current(current($Y_1; Y_2$))' = 'current(Y_1)'

Lemma 3: 'rest(current(Y_1); Y_2)' = 'Y_2'

By Lemma 1, and since 'skip' is the empty strategy, $[\![\text{tree}]\!]_{\text{current}(Y)}$ is invoked (for a given tree) only for cases (0), (1) and (2), and is defined for them below. Here I assume that *Constr* is the conjunction of background constraints (e.g., to not run into a car, and to not die) that must never be violated. The expression $[\![\text{tree}]\!]_{\text{current}(Y)}$ denotes a restricted intension of 'tree' relative to an agent or group (implicit here) and the substrategy it is to achieve by following 'tree'—in the sequel, I refer to it as the *strategy-relative intension* of 'tree.' This expression considers only those subscenarios where the success of the given substrategy is assured, i.e., forced, by the agent—fortuitously successful subscenarios are excluded. This expression captures the inherent reactivity of real-life agents—typically, a sequence of actions derived from a tree that just includes what a strategy prescribes would violate some constraint, or fail to achieve the relevant condition.

This leaves $[\![\text{tree}]\!]_{\text{current}(Y)}$ to be defined for all the possible cases of 'current(Y).' Briefly, $[\![\text{tree}]\!]_{\text{current}(Y)}$ is the set of subscenarios corresponding to executions of 'tree' such that they lead to the strategy 'current(Y)' to be forced. The three cases are (for different values of 'current(Y)') as follows.

$\langle S, t, t' \rangle \in [\![\text{tree}]\!]_{\text{current}(Y)}$ iff (here $t, t' \in S$; the world is w)

0. skip: The empty strategy, 'skip' is achieved by the empty tree. It would seem that any tree would achieve 'skip,' but that would only complicate the definition where this expression is used, because the recursive condition applies at the end of any subscenario in the set.

   tree $= \emptyset$ and $t = t'$

1. do(A): 'Tree' follows 'do(A)' iff the group *knows how* to achieve A in doing it, and all subscenarios over which it is done preserve the background constraints. This allows agents to overact, i.e., to do more than A requires (because of their habits, perhaps).

   if tree $= \emptyset$ then $t = t' \wedge M \models_{w,t} A$ else $\langle M, \text{tree}\rangle \models_{w,t} K_{how}(G, A) \wedge (\exists t'' : t < t'' \leq t' \wedge \langle S, t, t''\rangle \in [\![\text{root}]\!] \wedge (\exists\, \text{sub} : \text{sub} \in \text{tree.subtrees} \wedge \langle S, t'', t'\rangle \in [\![\text{sub}]\!]_{\text{current}(Y)})) \wedge (\forall t'' : t \leq t'' \leq t' \rightarrow M \models_{w,t''} \mathit{Constr})$

2. wait(A): 'Tree' follows 'wait(A)' iff it terminates just when A occurs, and all subscenarios over which it is done preserve the background constraints.

if tree $= \emptyset$ then $t = t' \wedge M \models_{w,t} A$ else $(\exists t'' : t < t'' \leq t' \wedge \langle S, t, t'' \rangle \in [\![\text{root}]\!] \wedge (\exists\, \text{sub} : \text{sub} \in \text{tree.subtrees} \wedge \langle S, t'', t' \rangle \in [\![\text{sub}]\!]_{\text{current}(Y)})) \wedge (\forall t'' : t \leq t'' \leq t' \rightarrow M \models_{w,t''} \mathit{Constr})$

Now for some intuitions about the definition of *know-how* relative to a strategy. The execution of a strategy by a group of agents is equivalent to its being unraveled into a sequence of substrategies of the form do(A) and wait(A). Each of these substrategies is followed by the group—it follows each by doing actions prescribed by some tree. Thus the substrategies serve as abstractions of trees of basic (or reactive) actions. In this way, the definition of *know-how* exhibits a two-layered architecture of agents: the bottom layer determining how substrategies of limited forms are achieved, and the top layer how they are strung together to form complex strategies.

6 Group Structure

The theory, as developed so far, ignores the internal structure of groups. Also, it considers only the strategies of the entire group, but not of its members individually; thus the abstraction of behavior with strategies, which is useful in both the design and analysis of multiagent systems, cannot be applied recursively to subgroups. With the inclusion of member strategies, the framework could be used (1) internally as a rationale for limiting the requirements on the perceptual hardware, knowledge, and reasoning abilities of members to just those that their own strategies (and their interactions with others) require, and (2) externally to motivate the combination of the knowledge of different members as well as communications among them. Therefore, I now extend the theory to allow individual strategies for the members of a group. When abilities are parsed into strategies and reactive actions, it is natural to consider the structure of a group in terms of interactions among its members at the levels of their strategies and their reactive actions. Accordingly, I now define the structure of a group in terms of two kinds of interactions among its members. The interactions among the members of a group can be seen as determining their respective "roles" in the group. Specifically, strategic interactions are the external correlates of the representational roles of Werner [29], where they are seen as intentions to cooperate. The present approach is more general in allowing groups of agents whose interactions are at the level of reactive actions, rather than intentional states, and in allowing groups whose members are not cooperative (e.g., two opposing tennis players constitute a non-cooperative group).

1. **Strategic Interactions:**

   A set of strategies (one for each member of a group) becomes a strategy of the group only if it satisfies the restrictions assigned by the structure of the group. Whether these restrictions are in fact met cannot be determined from the strategies alone, but depends on how the world evolves, and how the strategies are followed. It is instructive to see what kinds of interactions among strategies are possible. Some of these abstract interactions among agents may involve queries, requests for certain (possibly abstract) actions, commands to do certain (possibly abstract) actions—in short, all kinds of illocutionary acts [2]. The key point is that all of these involve the flow of abstractly specified information of some sort, be it a command or a declaration, from one member to another. Note that information need not be passed linguistically: nonlinguistic conventions, such as gestures, could serve as well. Other abstract interactions involve the establishment of various conditions in the world by some members' strategies that other members' strategies rely on. E.g., in a successful football team, the receivers run the patterns that the quarterback asks them to, some players are supposed to clear a path for their teammates, and whenever a defensive player identifies a move of the opposing team, he acts to block it.

2. **Reactive Interactions:**

   The structure of a group also determines the joint habits of its members. These are not specified in terms of interactions of the members' strategies, but rather in terms of their interactions while acting reactively as they execute their strategies. These may be implicit in the way in which a particular group acts. E.g., several interactions that make a team successful are not present in the constraints on the strategies of the members. A player may simply obstruct an opposing player trying to tackle his teammate, even though his own strategy is to run; good players react to their opponents' moves by pushing as hard as possible without running into their teammates or committing fouls. The joint habits of agents alluded to here show up in the formal model in two places. One is as constraints that specify inadmissible sets of actions or action sequences (by different members of a group); e.g., a member of a group would not perform his basic action of running a step in a direction when that would mean collision with a teammate. Another way in which joint habits show up is as positive constraints on the ordering of basic actions of different members of a group; e.g., a football player would block an opponent bearing down on his teammate if the latter is carrying the ball.

Formally, let a group, $G = \langle\langle x_1, \ldots, x_n\rangle, I_S, I_R\rangle$, where the x_i are the members of G, and I_S and I_R are, respectively, the sets of restrictions on the strategic and reactive interactions among the members of G. Then, the group strategy, Y_G, is an ordered set of member strategies, $\langle Y_1, \ldots, Y_n\rangle$ (where Y_i is the strategy of x_i), such that the restrictions in $I_S \cup I_R$ are 'met' as each x_i follows strategy Y_i. The 'current' parts of the strategies of different members could proceed differently and, in particular, not end synchronously. Define the 'current' of a group strategy as just an ordered set of the 'current' of each member strategy. I now introduce a predicate, 'continuation,' on strategies. Intuitively, if the 'current' part of a strategy is in progress, its continuation is the entire strategy; if the 'current' part of a strategy is over, its continuation is just the 'rest' of it. This is motivated by the lemmas of §5. I define 'continuation(D, Y_G)' as the group strategy whose ith element, for $i \in D$, is 'rest(Y_i),' and for $i \notin D$, 'Y_i.' Intuitively, D is the subset of members of G who have *done* their 'current' parts.

I also need to extend the definition of trees of actions so it would apply to groups of relatively autonomous agents. For a group, a tree is defined as a set of trees, one for each member of the group. For a set of 'current' parts of strategies (one for each agent), a tree containing a set of trees ($\{\mathrm{tree}_1, \ldots, \mathrm{tree}_n\}$—also one for each agent) has a strategy-relative intension that contains all subscenarios $\langle S, t, t' \rangle$ such that $(\forall i : 1 \leq i \leq n : (\exists t'' : t \leq t'' \leq t' \wedge \langle S, t, t'' \rangle \in [\![\text{tree}_i]\!]_{\text{current}(Y_i)}))$. The members' actions are thus described independently of each other to the extent of the following of their respective strategies. The required interactions are taken care of by the group restrictions described above. Then,

- $\langle M, Y_G \rangle \models_{w,t} K_{how}(G, A)$ iff $(\exists$ tree: $(\exists S, t' : (\forall i : 1 \leq i \leq n \rightarrow \langle S, t, t' \rangle \in [\![\text{tree}_i]\!]_{\text{current}(Y_i)}) \wedge (\forall r : r \in (I_S \cup I_R) \rightarrow \text{meets}(r, t', Y_G, \text{ tree})) \wedge \langle M, \text{tree} \rangle \models_{w,t} K_{how}(G, A) \vee (\forall S : S \in \mathbf{S}_{w,t} \wedge (\exists t' : \langle S, t, t' \rangle \in [\![\text{tree}]\!]_{\text{current}(Y)}) \rightarrow (\exists t'', D : D \subseteq [1, \ldots, n] \wedge (\forall i : i \in D \rightarrow \langle S, t, t'' \rangle \in [\![\text{tree}_i]\!]_{\text{current}(Y_i)}) \wedge (\forall j, t''' : j \in ([1, \ldots, n] \setminus D) \wedge \langle S, t, t''' \rangle \in [\![\text{tree}_j]\!]_{\text{current}(Y_j)} \rightarrow t'' < t''') \wedge \langle M, \text{continuation}(D, Y_G) \rangle \models_{w,t''} K_{how}(G, A))))$

In words, a group *knows how* to achieve A relative to strategy Y_G iff its members can follow the 'current' parts of their individual strategies while interacting in the right way, and either (i) achieve A or, (ii) as they finish their 'current' parts, go on to do the 'rest' of their respective strategies (this is captured by 'continuation'), and *know how* to achieve A relative to the 'continuation' of the group strategy. The recursive condition is applied every time a subset of the members (whose indices are in D) complete performing the sequence for the 'current' part of their strategies: those members then move on to the next part of their strategies as the others simply carry on; i.e., the member strategies need not proceed in lockstep. Satisfaction *simpliciter* is defined as before. This definition is perhaps stronger than necessary since it requires that the restrictions on the actions of different members of the group be met within each tree that they follow to achieve the 'current' parts of their strategies. A weaker option would be to consider groups of extended "memory," whose members would take care of their interactions over their entire sequences of actions for following their strategies.

The new predicate, 'meets,' is meant to accommodate the restrictions on the interactions among agents, and has to be defined for each kind of restriction in $I_S \cup I_R$. The kinds of restrictions that would be of interest depend on the domain that is being modeled. I consider two important classes of restrictions that occur in many application domains. The first kind involve the temporal ordering of the substrategies of the members of G (and is then in I_S), or the ordering of their actions (and is then in I_R). For I_S, I define "labels" as predicates that are true exactly when the substrategy they name has been successfully followed. This allows us to express temporal conditions such as 'l_p before l_q,' which states that l_p always occurs before l_q. Other conditions can be similarly handled. Note that both this and the next definition, though syntactically defined for any strategy Y, rely on the fact that they are invoked only with respect to a strategy of one of the forms of 'current(Y),' which are the forms for which the strategy-relative intension of a tree is defined.

- $M \models_{w,t}$ meets('l_p before l_q', t', Y, tree) iff $(\forall S : S \in \mathbf{S}_{w,t} \wedge \langle S, t, t' \rangle \in [\![\text{tree}]\!]_Y \rightarrow (\forall t'' : t \leq t'' \leq t' \wedge M \models_{w,t''} l_q \rightarrow (\exists t''' : t \leq t''' < t'' \wedge M \models_{w,t'''} l_p)))$

Other kinds of restrictions in I_S correspond to the different illocutionary acts that members are able to perform. Different restrictions would state that a member's commands to another would always be obeyed, or that a member's request to another would be accepted if certain conditions obtain, and so on. I have space here only to outline the approach for commands. Commands can be given only by agents to their subordinates: I extend I_S to include statements of hierarchy: 'x_i commands x_j' (in other words, x_i is the master and x_j the slave). Now let "command(agent, condition)" be an action that a member's (in this case, the master's) strategy may require to be 'done.' Then,

- $M \models_{w,t} \text{meets}(\text{'}x_i \text{ commands } x_j\text{'}, t', Y, \text{tree})$ iff $(\forall S : S \in \mathbf{S}_{w,t} \wedge \langle S, t, t' \rangle \in [\![\text{tree}]\!]_Y) \rightarrow (\forall t_1, t_2 : t \leq t_1 \leq t_2 \leq t' \wedge \langle S, t_1, t_2 \rangle \in [\![\text{do}(\text{command}(x_j, A))]\!]^{x_i} \rightarrow (\exists t_3 : t_2 \leq t_3 \leq t' \wedge M \models_{w,t_3} A)))$

Thus 'tree' 'meets' a command restriction, if all commands by superiors (as defined by that restriction) to their subordinates are eventually obeyed within the 'tree.'

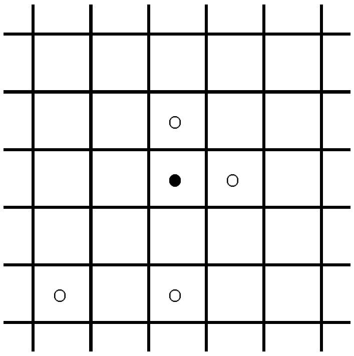
While, as I had indicated in §4, it is possible to treat the "habits" of agents as restrictions on $[\![]\!]$, it is often convenient to treat the reactive interactions in a group explicitly. Conditions that involve the relative ordering of actions of members can then be handled in the formalism in the same manner as the ordering of substrategies, though their implementational import is quite different.

6.1 Example: The Pursuit Problem

The theory developed above may be used in modeling the ability of multiagent systems. I now apply it in the context of a well-known problem in DAI: the pursuit problem. Parts of this section are borrowed from [26] which considers the same problem from the viewpoint of the agents' intentions. The pursuit problem problem was introduced by Benda *et al.* [3], but has been extensively studied by others [6, 28]. We are given a finite two-dimensional grid of points (see Figure 1). Each point may be occupied by either an agent called 'Red' (the "adversary") *or* up to four 'Blue' agents. At each cycle, each agent can stay in its location or move one square up, down, left, or right. The pursuit starts in some arbitrary configuration and ends in either the Blue agents winning (when they occupy the four locations surrounding Red) or losing (if Red gets to the edge of the grid).

Let the Blue agents be notated as B^1 through B^4, Red as R, and any of the five as A. Let A_x denote the x coordinate of A's location, and A_y the y coordinate. Initially, let the agents be specified to get on different sides of R; e.g., B^1 above it, B^2 to its right, B^3 below it, and B^4 to its left. One could talk of the *know-how* of these agents in the purely reactive sense, and the *know-how* of the group that results from it. Instead it is convenient to directly go to the case of strategic *know-how.* Each of the Blue agents could be said to have some abilities on the basis of the strategies they can possibly have. The group as a whole would have a strategy whose substrategies are the agents' strategies. The strategies that the agents can have are determined by how they are designed. But even for a rigid design one can describe the agent as having several strategies at different levels of detail.

In the simplest case, each of the Blue agents could be said to have a strategy that if followed successfully would take it to its proper slot; e.g., B^1's strategy, Y_1, could just be 'do(get-above-R)'; the others would be analogous. This specification assumes that lower layers of the design are available to do the required actions. These layers would also ensure that no constraint is violated (e.g., collision with Red is avoided). First consider the case



(• = Red, ○ = Blue)

Figure 1: The Pursuit Problem: an example configuration

where no restrictions have been imposed on the group. Thus the group has the *know-how* to achieve the condition that B^1 occupy the location above Red, B^2 the one to its right, and so on. In other words, the condition the group as a whole has the *know-how* for is that it *win* (or equivalently that Red lose). It is obvious from the construction that none of the Blue agents can individually have the *know-how* for anything beyond occupying a certain location. Yet the group *knows how* to achieve a win. The group's *know-how* derives simply from the combinations of its members' *know-hows*.

At the next level of detail, we might assign strategies to the Blue agents as follows. Let Y_1 be

while $\neg$over$_1$ do
    if $B^1_y > R_y + 1$ then do(move-down)
    else if $B^1_x < R_x$ then do(move-right)
        else if $B^1_x > R_x$ then do(move-left)
            else if $B^1_y < R_y$ then do(move-up) else skip

Figure 2: An Example Strategy for a Blue Agent

The other Blue agents' strategies are analogous. Here I assume that 'over' is true when the given Blue agent abuts Red on the appropriate side for two clock cycles. As before, I also assume that the lower layers of the architecture ensure that no constraint is violated. The above strategies provide a more detailed specification of what a Blue agent would do, and thus of its internal architecture; i.e., they are quite close to its implementation.

In both these cases *know-how* was attributed to the group of Blue agents quite independently of whether the individual agents knew of it or planned to achieve it. Indeed, the group itself was described independently of whether its members knew they participated in it. A more elaborate system in which the agents took on goals depending on their location could also be described in this framework: their strategies would have to be more complex; e.g., they could each decide where to go depending on the globally optimal assignment. At greater levels of detail, the strategies assigned to the agents could include the aspects of interaction required for them to adopt non-conflicting goals. For concreteness, consider a system where B^1 becomes the controller, and the other Blue agents its slaves. Now Y_1 could begin by first making an assignment of the locations surrounding Red to different Blue agents; then *command* the other Blue agents to go there. Y_2, Y_3 and Y_4 could then simply be strategies to go to any specified location about Red. The group would impose the strategic restriction that all commands from B^1 are obeyed. Again, the *know-how* ascribed previously could be ascribed, along with that for some more complex conditions, e.g., about the specific order in which actions are done by different agents.

The moral that I wish to derive from these examples is that while adding strategies, and further constraints on the interactions between the members of a group does not seem to add to its absolute *know-how*, it does serve a useful purpose from the point of view of designing a group (and the agents in it). The process of design has not been formalized here but is an important focus for future research. The idea is that while an unconstrained group has the most *know-how*, the designer's obligation so that member agents can meet the specification is also the strongest—these agents must all be able to figure out what to do when, and be able to make the relevant observations on the fly. For a constrained group, the designer can manage to meet the specifications with a considerably simpler design than otherwise.

7 Some Consequences

Some important consequences (both positive and negative) of the theory described in this paper are enumerated below. These consequences help characterize the theory of this paper in terms of the most important predicates and abstract out many of the details of the model. They will also be useful in guiding the formulation of design principles for multiagent systems. As usual, '$\models p$' indicates that p is valid, and '$\not\models p$' that it is invalid. Following Emerson [7], AG means "at all times in all futures," and P "sometimes in the past." These and other temporal operators could easily be given formal definition in the model, if that is desired.

1. **Meta Know-how Implies Know-how:**

   $\models K_{how}(G, K_{how}(G, A)) \rightarrow K_{how}(G, A)$

   If a group *knows how* to come to *know how* to do A then it must already *knows how* to do A. In particular, one sure way G has to achieve A is to achieve $K_{how}(G, A)$ first, and then (using that *know-how*) to achieve A. This falls out as a simple consequence of the formal definition of *know-how*. In a sense, this flattens the metaplanning vs planning distinction.

2. **Monotonicity Through Non-interactive Group Formation:**

   $\models (\exists i : 1 \leq i \leq n \wedge K_{how}(x_i, A)) \rightarrow K_{how}(\langle\langle x_1, \ldots, x_n\rangle, \emptyset, \emptyset\rangle, A)$

   The group consisting of just the given agents, and imposing no restrictions on their interactions, has the *know-how* of each of them. This is because membership in a group does not affect the agents' basic actions; it only constrains how they may use them (to make sure that all interactions are legitimate).

3. **Cooperation By Chaining:**

   $\models K_{how}(x, K_{how}(y, A)) \rightarrow K_{how}(\langle\langle x, y\rangle, \emptyset, \emptyset\rangle, A)$

   Thus x and y can collaborate by "chaining" their *know-hows*—x can first exercise his *know-how* to bring about it that y has the *know-how* to achieve A. Then, since y has the *know-how* to achieve A, he can do so independently of x's actions.

4. **Command Restriction:**

   $\not\models [\mathsf{AG}(q \rightarrow \mathsf{AG}\neg p) \wedge \mathsf{AG}(K_{how}(y, q))] \rightarrow$
   $K_{how}(\langle\langle x, y\rangle, \{x \text{ commands } y\}, \emptyset\rangle, (\mathsf{P}\ \text{command}(y, p) \wedge \neg \mathsf{P}p \wedge q)])$

   If x commands y to achieve p, then y must achieve p, and thus q (whose occurrence would exclude p forever) cannot be achieved before p. If the group lacked the 'commands' restriction, the above implication would be valid.

5. **Temporal Order Restriction:**

   $\not\models [K_{how}(x, p) \wedge \mathsf{AG}(K_{how}(y, q)) \wedge \mathsf{AG}(q \rightarrow \mathsf{AG}\neg K_{how}(x, p))] \rightarrow$
   $K_{how}(\langle\langle x, y\rangle, \{q \text{ before } p\}, \emptyset\rangle, [(p \wedge q) \vee (p \wedge \mathsf{P}q) \vee (\mathsf{P}p \wedge q)])$

   If the group restriction were removed, x could achieve p, and then y could achieve q. However, in the above case, q must occur before p, and that erases x's *know-how* for achieving p.

6. **Emergence of Group Know-how:**

   $\not\models K_{how}(G, A) \rightarrow (\exists i : x_i \in G \wedge K_{how}(x_i, A))$

   A group's *know-how* arises from the interactions of its members; it is possible (and, in practice, quite likely) that no single member of a group has the same *know-how* as the entire group.

7. **Loss Due to Interactions (Weak):**

   $\not\models (\exists i : x_i \in G \wedge K_{how}(x_i, A)) \rightarrow K_{how}(G, A)$

   The strategy that the given member x_i would follow to achieve A may be inadmissible as a strategy in the definition of group *know-how* because it is not possible to follow it appropriately (and meet all the restrictions imposed by the group's structure).

8. **Loss Due to Interactions (Strong):**

   $\not\models (\forall i : x_i \in G \rightarrow K_{how}(x_i, A)) \rightarrow K_{how}(G, A)$

   The same reason as in the previous case. Thus a group has greater *know-how* than all its members, even if they are considered together.

As remarked in §6.1, the inclusion of strategies and group structure into the theory seems to reduce the *know-how* of groups. This is to be expected since strategies and group structure allow us to model agents with limited abilities for perceiving, reasoning, and communicating, who must also obey certain social constraints. The *know-how* of real-life groups is not really reduced; rather, a more accurate estimate of it is obtained.

8 Conclusions

The theory presented here refines and formalizes some intuitions about the abilities and structure of groups, and stresses the connection between them. I started out with a set of obvious intuitions about the abilities and structure of groups of intelligent agents, and with a simple formal model of action and time that allows concurrent actions by different agents. This yielded a direct definition of the *know-how* of a group treated as a set of purely reactive agents. However, it was felt that real-life groups (who are complex, and need to act in a complex world) would of necessity have (or at least be designed to have) some internal (i.e., architecture) structure. This structure should be accounted for in a theory of action and ability, and could in fact be exploited in giving a more realistic account of a group's *know-how*.

Further work, however, is needed to obtain a more "internal" or architectural view of *know-how*, and to relate it to the general "roles" that different agents may play in a group [29]. An important problem in this framework is to relate abilities and structure to perception and reasoning, especially when reasoning is not thought of as pure symbol-processing. This work also has impact on the development of a unified theory of the intentions, beliefs, and *know-how* of groups of autonomous intelligent agents. Such a theory would be crucial in forming the basis for a philosophically and mathematically sound theory of the design and analysis of groups of agents using folk psychological concepts.

9 Acknowledgements

I am indebted to Michael Huhns, Rob Koons, Elaine Rich and to three anonymous referees for comments. This research was partially supported by the Microelectronics and Computer Technology Corporation, and by the National Science Foundation (through grant # IRI-8719064 to the Center for Cognitive Science, University of Texas).

References

[1] Philip Agre and David Chapman. Pengi: An implementation of a theory of activity. In *AAAI*, pages 268–272, 1987.

[2] John L. Austin. *How to do Things with Words*. Clarendon, Oxford, UK, 1962.

[3] Miroslav Benda, V. Jaganathan, and Rajendra Dodhiawala. On optimal cooperation of knowledge sources. Technical report, Boeing Advanced Technology Center, Boeing Computer Services, Seattle, WA, September 1986.

[4] Philip R. Cohen and Hector J. Levesque. On acting together: Joint intentions for intelligent agents. In *Workshop on Distributed Artificial Intelligence*, 1988.

[5] Y. Demazeau and J-P. Müller, editors. *Decentralized Artificial Intelligence*, Amsterdam, Holland, 1990. Elsevier Science Publishers B.V. / North-Holland.

[6] Edmund H. Durfee and Thomas A. Montgomery. MICE: a flexible testbed for intelligent coordination experiments. In *Proc. 9th Workshop on Distributed Artificial Intelligence*, pages 25–40, September 1989.

[7] E. A. Emerson. Temporal and modal logic. In J. van Leeuwen, editor, *Handbook of Theoretical Computer Science*, volume B. North-Holland Publishing Company, Amsterdam, The Netherlands, 1990.

[8] Ronald Fagin and Joseph Y. Halpern. Belief, awareness, and limited reasoning. *Artificial Intelligence*, 34:39–76, 1988.

[9] Michael J. Fischer and Neil Immerman. Foundations of knowledge for distributed systems. In Joseph Y. Halpern, editor, *Theoretical Aspects of Reasoning About Knowledge*, pages 171–185, 1986.

[10] Les Gasser and Michael N. Huhns, editors. *Distributed Artificial Intelligence, Volume II.* Pitman/Morgan Kaufmann, London, 1989.

[11] Barbara Grosz and Candace Sidner. Distributed know-how and acting: Research on collaborative planning. In *Workshop on Distributed Artificial Intelligence*, 1988.

[12] Barbara Grosz and Candace Sidner. Plans for discourse. In P. Cohen, J. Morgan, and M. Pollack, editors, *SDF Benchmark Series: Intentions in Communication.* MIT Press, Cambridge, MA, 1990.

[13] Joseph Y. Halpern and Yoram O. Moses. Knowledge and common knowledge in a distributed environment (revised version). Technical Report RJ 4421, IBM, August 1987.

[14] C. L. Hamblin. *Imperatives.* Basil Blackwell Ltd., Oxford, UK, 1987.

[15] Carl Hewitt. Organizational knowledge processing. In *Workshop on Distributed Artificial Intelligence*, 1988.

[16] Michael N. Huhns, editor. *Distributed Artificial Intelligence.* Pitman/Morgan Kaufmann, London, 1987.

[17] Kurt G. Konolige. A first-order formalism of knowledge and action for multi-agent planning. In J. E. Hayes, D. Mitchie, and Y. Pao, editors, *Machine Intelligence 10*, pages 41–73. Ellis Horwood Ltd., Chichester, UK, 1982.

[18] Dexter Kozen and Jerzy Tiurzyn. Logics of program. In J. van Leeuwen, editor, *Handbook of Theoretical Computer Science.* North-Holland Publishing Company, Amsterdam, The Netherlands, 1990.

[19] Drew McDermott. A temporal logic for reasoning about processes and plans. *Cognitive Science*, 6(2):101–155, 1982.

[20] Robert C. Moore. A formal theory of knowledge and action. In Jerry R. Hobbs and Robert C. Moore, editors, *Formal Theories of the Commonsense World*, pages 319–358. Ablex Publishing Company, Norwood, NJ, 1984.

[21] Leora Morgenstern. A theory of knowledge and planning. In *IJCAI*, 1987.

[22] Stanley J. Rosenschein. Formal theories of knowledge in AI and robotics. *New Generation Computing*, 3(4), 1985.

[23] Gilbert Ryle. *The Concept of Mind.* Oxford University Press, Oxford, UK, 1949.

[24] John R. Searle. Collective intentions and actions. In P. Cohen, J. Morgan, and M. Pollack, editors, *SDF Benchmark Series: Intentions in Communication.* MIT Press, Cambridge, MA, 1990.

[25] Yoav Shoham. *Reasoning About Change: Time and Causation from the Standpoint of AI.* MIT Press, Cambridge, MA, 1988.

[26] Munindar P. Singh. Group intentions. In *10th Workshop on Distributed Artificial Intelligence*, October 1990.

[27] Munindar P. Singh. Towards a theory of situated know-how. In *9th European Conference on Artificial Intelligence*, August 1990.

[28] Larry M. Stephens and Matthais Merx. The effect of agent organization on the performance of DAI systems. *IEEE Transactions on Systems, Man and Cybernetics*, 1989. Submitted.

[29] Eric Werner. Cooperating Agents: A Unified Theory of Communication and Social Structure. In L. Gasser and M. N. Huhns, editors, *Distributed Artificial Intelligence, Volume II*, pages 3–36. Pitman/Morgan Kaufmann, London, 1989.