Chapter X

Challenges to Scaling-Up Agent Coordination Strategies*

Edmund H. Durfee
Department of Electrical Engineering and Computer Science, University of Michigan, Ann Arbor, MI 48109, durfee@umich.edu

Abstract: There is more to scaling up agent-based systems than simply increasing the number of agents involved. Many of the challenges to making agent-based systems work in more realistic settings arise from the characteristics of the agents' tasks and environment, and the expectations of the systems' users. In this chapter, my goal is thus to emphasize this broader array of challenges to coordinating agent-based systems, as a step both towards extending our understanding of scale-up issues as well as towards developing richer metrics for evaluating the degree to which coordination strategies for agent-based systems can apply to more demanding applications.

Key words: Multi-agent systems; coordination

1. INTRODUCTION

The notion of deploying "intelligent agents" to do peoples' bidding in environments ranging from marketplaces on the internet to robotic exploration of Mars has recently received much attention and speculation. Meanwhile, exactly what an "agent" is and in what senses a computational agent can behave "intelligently" are still undergoing much debate. Rather than confront such thorny issues head on, this article skirts around most of them to focus more squarely on just one of the central concerns of intelligent agency: coordination.

With few exceptions, if an agent is dispatched to an environment, the odds are that it will share the environment with other agents. Even some

* A condensed version of this chapter appeared as "Scaling-Up Agent Coordination Strategies" in *IEEE Computer* 34(7):39-46, July 2001.

proposed strategies for robotic exploration of the planets typically involve sending a team of robots! Thus, a fundamental capability needed by an agent is the ability to decide on its own actions in the context of the activities of other agents around it. This is what we will mean when we refer to coordination. Note that this does not mean that coordination must imply cooperation: an effective competitor will coordinate its decisions to work to its advantage against an opponent, such as a producer of goods timing a product promotion to undercut a competitor. It does not even imply reciprocation: an agent may be coordinating with another who is unaware of it, such as one automobile driver trying to pass a second whose mind is entirely elsewhere.

Without coordination, agents can unintentionally conflict, can waste their efforts and squander resources, and can fail to accomplish objectives that require collective effort. It is therefore no wonder that a variety of strategies for coordination among computational agents have been developed over the years, in an effort to get "intelligent agents" to interact at least somewhat "intelligently."

It does not seem possible to devise a coordination strategy that always works well under all circumstances; if such a strategy existed, our human societies could adopt it and replace the myriad coordination constructs we employ, like corporations, governments, markets, teams, committees, professional societies, mailing groups, etc. It seems like whatever strategy we adopt, we can find situations that stress it to the breaking point. Whenever a coordination strategy is proposed, therefore, a natural question that arises is "How does it scale to more stressful situations?"

In an effort to map the space of coordination strategies, therefore, we need to define at least some of these dimensions in which they might be asked to "scale," and then figure out how well they respond to being stressed along those dimensions. For example, clearly one of the most measurable scaling dimensions is simply the number of agents in the system. Yet, sheer numbers cannot be all there is to it: the coordination strategies employed in insect colonies seem to scale to large numbers of insects, yet they do not seem to satisfy all the needs of large human societies (New York City traffic not withstanding). One of my goals in writing this article, therefore, is to provoke a dialogue about what it means for a coordination strategy to "scale up."

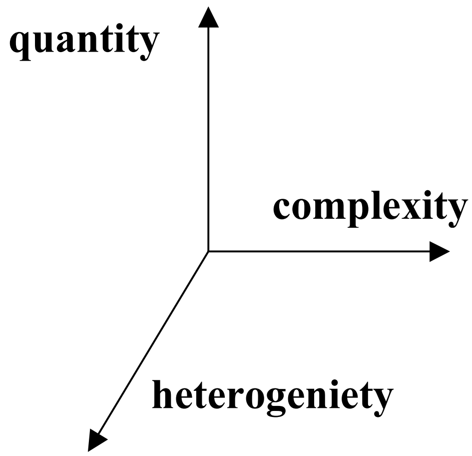
Moreover, as Jennings has suggested, agent-oriented software engineering shows promise for developing complex, distributed systems, but requires the component agents to act and interact flexibly (Jennings 2001). A second goal of this article is therefore to provide some potentially useful starting points for characterizing portions of the space of coordination problems, so as to better understand the capabilities and limitations of strategies developed to support flexible interaction. Toward this end, I'll begin by forming a characterization of the coordination problem space by looking at properties of the agent population, of the task-environment the agents inhabit, and of the expectations about their collective behaviors. I'll then turn to giving a very brief survey of a few (of many) coordination strategies and how they fit into this space. I'll conclude by pointing out gaps in our understanding, and suggest opportunities for progress in the field.

2. SOME DIMENSIONS OF COORDINATION STRESS

There are more factors that influence how difficult it is to bring about coordination than can be covered here. Therefore, this article tries to project the richness of this space while also simplifying enough to allow a reader to grasp portions the space. To that end, I'll limit discussion to three dimensions (so as to allow depiction on a 2-dimensional page) along each of the major properties: of the agents, of the task-environment, and of the solution. It should be noted up front that these dimensions are not necessarily orthogonal; in some cases relationships between them are indicated. Nonetheless, treating them as orthogonal can be useful in characterizing the space of coordination challenges.

2.1 Agent Population Properties

We begin with the most obvious properties that will impact coordination: those of the set of agents that need to coordinate. Certainly, one of the challenges in scaling any coordination strategy, as previously mentioned, is handling larger and larger numbers of agents. Coordination strategies that rely, for example, on a centralized "coordinator" to direct the interactions of the other agents can quickly degrade as the coordinator becomes incapable of processing all of the interactions given increasing numbers of potentially interacting agents. If each agent can potentially interact with every other agent, then the number of pairwise interactions to analyze grows quadratically with the number of agents. More problematically, since interactions often must be viewed in terms of larger groups of agents (not just pairs), the problem can devolve into a problem of exponential size: if each agent could choose among b actions, each potentially having a different impact on other agents, then the space of all possible action combinations will be b^n, for n agents. Even if each of the n agents participated in the



coordination search, rather than depending on a centralized coordinator, an n-fold speedup of a problem that is exponential in n doesn't help much.

A second dimension that often poses challenges in coordination is what is broadly labeled as "heterogeneity." Agents within a population can be different from each other in many possible ways. For example, due to occupying different places in the environment, they might know different things about the current state of the world. If they know different things about the way the world works, then we might say they have heterogeneous expertise. They could have differing abilities to sense the world or change the world. Especially in the case of competitors, they could have different preferences for how the world should be. They could even have different communication languages, ontologies, or internal architectures. Whether a coordination strategy scales to increasingly heterogeneous populations depends on the degree it expects agents to in principle be able to communicate with, share their abilities with, and basically agree with each other.

Finally, the third dimension of agent properties we will consider here is what I term "complexity." While this could mean many things, I'll focus on it as referring to how hard it is to predict what an agent will do because of inherent versatility on the part of an agent. One of the features that arguably makes something an "intelligent" agent is that it is capable of flexibly deciding for itself which goals to pursue at a given time and how to pursue them. Agents that are not complex, under this characterization, are those that can be seen as single-mindedly doing a specialized task. In general, coordinating with such agents is easier (they are much more predictable) than coordinating with agents that could be doing any of a number of things. Couple this with the possibility of overlaps among agents' spheres of interest and ability, and this can put enormous stress on any coordination strategy that that wants to assume unambiguous matches between tasks or roles in the system and the agents to do them.

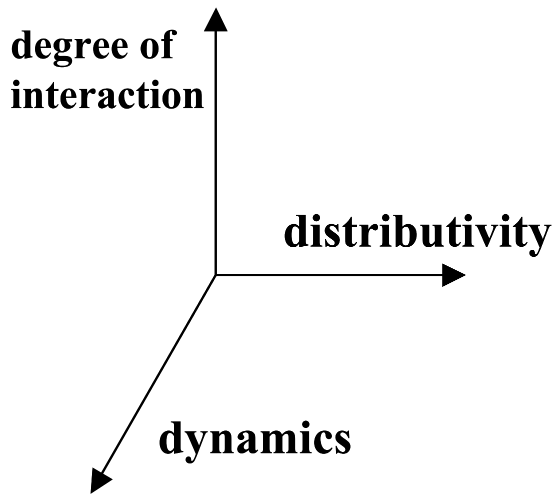
Obviously, scaling along combinations of these dimensions can pose even greater challenges. Handling complex agents is much harder, for example, if they are complex in different (heterogeneous) ways, but easier if there aren't very many of them. Coordination strategies will tend to therefore make assumptions about which dimensions are likely to be stressed for the application domain of interest.

2.2 Task-Environment Properties

The environment in which agents operate, and the tasks they are expected to accomplish within the environment, are another major consideration in developing or choosing a coordination strategy. Real task-environments often introduce complications that blur the understanding of a coordination strategy: for example, in task-environments that require substantial domain expertise, it can be difficult to compare alternative coordination strategies because the differences in performance might be due to the quality of the knowledge given the individuals rather than to the efficacy of the coordination strategy. For this reason, researchers often employ abstract, idealized versions of task-environments such as pursuit problems, transport problems, the Prisoners' Dilemma, and distributed sensor networks (e.g., see (Weiss, 1999) and some of the sidebars associated with this article). Even with abstract task-environments, the possible dimensions for scaling the difficulty of coordination are numerous; again, only three are given here of the many possibilities.

The first dimension we will consider is the degree to which the environment, or the task, leads to interactions among agents that materially impact the agents. Since coordination is all about exerting some control over interactions, a greater degree of interaction implies more need to coordinate. Or, viewed the other way, agents that do not interact need not coordinate. Thinking slightly more concretely, suppose that an interaction involves some "issue" that involves more than one agent. The issue could be about who gets to use a resource, or about what the status of some feature of the world is, or about who is supposed to do what task, etc. The degree of agent interaction increases as more agents are concerned with the same issues, and as more issues are of concern to each agent, so that settling some issues commit agents to interactions that in turn impact how they should settle other issues. As the web of dependencies grows, some coordination strategies can have difficulty scaling.

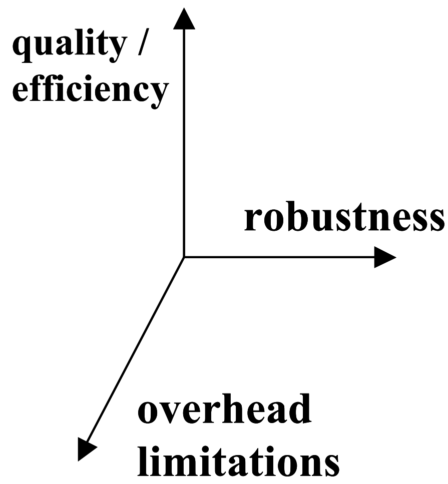
A second dimension that complicates coordination is the dynamics of the task-environment. Coping with changing environments is always difficult; in



a multiagent setting, where different agents might be capable of monitoring only portions of the environment, and where each might change its mind about what goals to pursue or what means to use to pursue goals, the difficulties are compounded. In more static task-environments, the agents have some hope of converging on coordinated activities and then carrying them out. But in more dynamic task-environments, convergence might be impossible: the task-environment might change faster than the coordination strategy can keep up. For this reason, coordination strategies that scale to highly dynamic task-environments are relatively uncommon.

A third dimension, which is related to the first two, as well as to agent heterogeneity, is what here will be called "distributivity." In some task-environments, agents are highly distributed in the (conceptual) environment and tasks are inherently distributed among the agents. In other task-environments, the agents are (conceptually) collected together – such as occupying a common "yellow pages," and tasks originate at one point. Distributivity stresses a coordination strategy because it increases agents' uncertainty about which agents are currently sharing the task-environment and what (if anything) each is, or should be, doing.

Again, scaling along combinations of these dimensions is possible, placing even more substantial demands on a coordination strategy. For example, distributivity compounds the difficulties in a dynamic task-environment, because of the inherent delays in propagating the implications of changes in a highly distributed setting, but lowering the degree of interaction can simplify this by localizing the need to propagate to fewer interested parties.



2.3 Solution Properties

To evaluate how well a coordination strategy deals with the scaling issues that we throw its way, we need to define criteria that we expect of a solution. One of the dimensions for solution properties, for example, is the "quality" of the solution, in terms of how well the interaction is coordinated. The quality might be measured in terms of efficiency – that is, whether the issues have been settled in a manner that permits the efficient use of agent resources and abilities. Higher quality can correspond to closer to optimal coordination. A less demanding level of quality might correspond to achieving a satisficing level of coordination. In some cases, simply avoiding disagreement (conflict) might be good enough. As illustration, if we were to design a coordination strategy for an automobile intersection, we might be satisfied if it prevents crashes, or we might further require that it achieve some measures such as ensuring no car needs to wait longer than some upper bound time, or we could insist that it minimize the expected wait time for all cars. As we demand more, we put greater stress on the coordination strategy.

A second dimension considers how robust we expect a solution to be in the face of uncertainty or dynamics in the task-environment and the agent population. For example, as was pointed out before, a coordination strategy might have trouble keeping up with a particularly dynamic task-environment. The coordination solution might therefore be somewhat out of date. If we demand that a solution nonetheless be robust, then the coordination strategy should anticipate, either implicitly or explicitly, the range of conditions under which the solution it provides will be followed, and not simply the single expected situation. Given that some task-environments might be such that a minor deviation from expectations can lead to severe consequences, finding assured robust solutions can, in some cases, be imperative.

Finally, a third dimension concentrates on the cost of the coordination strategy. A solution to the problem of how to coordinate should account for the costs of doing the coordination. These costs could include the amount of computation required, communication overhead, time spent, and so on. For example, if communication is costly and time-consuming, a coordination strategy might have to reduce its demands for information exchange among agents; beyond some point, it will have to make high-quality coordination decisions lacking information it would otherwise have expected to have. Therefore, questions can arise about whether a coordination strategy can scale well to environments that impose more stringent limits on costs that the strategy incurs.

As for the previous properties, these three dimensions can combine in various ways. For example, one way of improving robustness of a coordination solution without sacrificing quality is to continually monitor and update the solution in response to changes, but this in turn would require that minimizing costs and delays is not a significant objective.

3. CHARACTERIZING COORDINATION STRATEGIES

At this point, I've identified three major types of properties (agent population, task-environment, and solution), and for each I've described three (out of many possible) dimensions in which the property could be scaled to make coordination harder. If we were to qualitatively consider "low" and "high" values along each of the dimensions, we'd have eight possible points to consider for each property, leading to 8^3 combinations across the three properties. It would be tempting to now look at each of these 512 combinations in turn, and consider which coordination strategies make sense for each.

The truth is, however, that even if this book had enough room, and you the reader had enough patience, there isn't sufficient understanding of the entire space of coordination strategies that have (or could have) computational embodiments to fill all of these in. Instead, what follows summarizes just a handful of coordination strategies, highlighting where they fall within this space and the kinds of scaling for which they are particularly well suited. The selection of these strategies should not be viewed as an endorsement that the strategies given are superior to others not given, but rather is based on giving some representative examples across the space.

3.1 Agents

To some people, "scaling up" is equated to being able to handle more agents, and (almost always) handling more agents is harder than handling fewer. Trying to get a large population of complicated, self-interested, and interacting agents to somehow behave efficiently and robustly in a dynamic environment is a tall order. In fact, typically something has to give: usually, coordination strategies that scale well to large numbers of agents do not deal with many of these other confounding dimensions.

For example, cellular automata (Wolfram, 2002) often deal with large numbers of entities that typically use rules to react in simple ways to their very local environments, such as "deactivating" when too few neighbors are active, or "activating" when enough neighbors are active. Patterns of activity can emerge in the population through these very simple local changes. Physics-based models of large computational ecosystems of agents can even lead to designs of metamorphic robots made up many small pieces that shift and flow to adapt to the environment (Bojinov, 2001). Similarly, systems based on insect metaphors assume that each agent is a relatively simple automaton, and that emergent properties of interest arise due to their local interactions (Ferber 1999). These strategies assume little complexity and, often, little heterogeneity in the agent population, focus on very limited (local) kinds of interactions, and are satisfied with emergent, statistical system performance, rather than worrying about each agent being efficiently used or making optimal choices.

More generally, successfully scaling up to large numbers of agents generally requires that each agent only needs to interact with a constant (or slowly growing) number of other agents, and that who needs to interact with whom is preordained based on agents' features such as their physical locations or their tasks/roles. Thus, large numbers of mobile agents can be dispersed for information gathering tasks that can be pursued independently, interacting only indirectly due to contention for bandwidth or server cycles (Gray, 2001). Similarly, large-scale coalition/congregation formation can be viewed as an emergent process involving growing groups incrementally as agents (and agent groups) encounter each other and discover advantages of banding together (Lerman, 2000; Brooks, 2000).

3.2 More Heterogeneity

In the case of scaling up to large agent populations, agent heterogeneity can sometimes help, if agents that are different from each other need not interact. This serves to once again restrict the number of others about which an agent must be aware. More typically, however, heterogeneity is welcomed into a system because it increases the system-wide capabilities, whereby agents with complementary attributes combine their efforts toward objectives beyond what they can individually achieve. Once the agent population is no longer homogeneous, therefore, it becomes important for agents to be able to understand and often describe what they can do, and to find others with whom to work. Coordination strategies that do not support the ability of agents to describe themselves and to find each other, such as by having implicit acquaintanceships among agents "hardwired," have difficulty scaling along the heterogeneity dimension.

A mainstay coordination strategy for handling heterogeneity has been the Contract Net protocol (Smith 1980) and its descendents, whereby agents dynamically assign tasks to others who are available and capable of doing the tasks. In its simplest form, the protocol allows an agent with a task that it needs done to broadcast an announcement of the task, along with criteria by which each of the other agents can decide whether it is eligible to take on the task and, if so, what information to supply in a bid for the task. The agent with the task can choose from among the responses to make an assignment.

The Contract Net protocol scales well to an open system of heterogeneous agents, but as the number of agents increases, the broadcast communication requirements can be problematic. A response to this is to maintain a more centralized registry of agents and their capabilities, which can be used flexibly to discover promising matches between agents with tasks to do and agents that can do them. Strategies that support agent registration and matchmaking (for example, (Paolucci, 2000) or www.sun.com/jini) can allow agents to find each other by describing the kinds of services that they need or provide. More generally, formalisms for communicative acts, such as FIPA (www.fipa.org), can permit a broad array of conversation policies in support of flexible agent interactions among heterogeneous agents. Many of these concepts are being brought together in more comprehensive frameworks for supporting heterogeneous agent-based systems, such as DARPA's Grid (coabs.globalinfotek.com).

3.3 More Complexity

Heterogeneity tends to emphasize the challenges that accrue when "specialist" agents need to identify each other and team to provide broader services. Additional complications arise when agents are individually more complex, typically meaning that they are each more versatile, yet not identically so. Now, each agent must decide which of the possible roles that it could play it should play, and must reason about other agents in terms of the alternative activities they might be engaged in, rather than the specific activity that a "specialist" could be assumed to pursue.

Scaling up to more complex agents means that teaming involves not only finding an available agent with appropriate capabilities, but also selecting from among such agents so as to pick the one whose other talents are least in demand by other teams. Thus, interactions among agents are not localized within smaller teams, but rather the "partial substitutability" of agents for each other leads to complex chains of dependencies: how some teams are formed can color which other teams will be desirable. This means that agents must be increasingly aware of the broader needs of the agent network.

Similarly, even when agents do not need to team up, but merely must co-exist and stay out of each others' way, the increased versatility of each agent makes anticipating what others will be doing much more difficult. Being prepared for anything that another could choose to do might be impossible, so strategies for increasing awareness about other agents' planned activities becomes paramount. Strategies can include using statistics of others' previous behaviors, using observations of them to infer their current plans, or using communication to convey information that permits agents to adequately model each others' intentions.

As an example of the latter, the process by which agents that can accomplish their objectives in several different ways can converge on mutually compatible plans can be viewed as a distributed constraint satisfaction process. This process involves propagating tentative plan choices among agents and, when inconsistencies are detected among the choices of some subset of agents, systematic backtracking is performed by some of the agents. Increased efficiency in this process can stem from techniques that allow parallel asynchronous exploration of the space, and that can dynamically decide which agents should be asked to try alternatives based on measures of which constraints are proving most difficult to satisfy (Weiss, 1999, chapter 4; Yokoo, 2000).

3.4 Higher Degree of Interaction

As was previously stated, the need for coordination arises from agent interactions. As the number and complexity of agent interactions grow, coordination becomes intractable. Therefore, it isn't surprising that on effective means for addressing coordination is to reduce, or if possible eliminate, interactions. As already pointed out, when agents only have to worry about interactions with a small number of local "neighbors," then scaling to large numbers of agents is much easier. So strategies for localizing interactions (Lansky, 1990) can obviate the need for more complicated coordination strategies.

One often-used technique for controlling the degree of interaction is to impose a (relatively static) organizational structure on agents. Each agent is given a role to play in the organization, including its own sphere of control and knowledge of agents playing related roles. Giving each agent the resources it needs to fulfill its role eliminates the need for agents to negotiate over resources, and giving each agent knowledge of the roles of other agents dictates who needs to communicate with whom and about what. An appropriate organizational structure among agents can thus simplify coordination, and permit larger, more complex agent systems to succeed in more challenging task domains. The challenge, of course, is in designing organizations for agents, or having agents design their own organizations, such that the organizations match the agent population and the needs of the task-environment (Prietula, 1998).

Sometimes, however, multiagent tasks cannot be divided into nearly-independent pieces; there are some tasks that absolutely require tight interactions among agents. In the literature, examples of such tasks include the "pursuit" task where predators need to surround a prey (Gasser, 1987), and tasks involving team activities such as combat flight operations (Tambe, 1995). For such applications, interactions are not a side-effect of individuals acting in a shared world, but rather are the purpose of the individuals' actions in the first place. Therefore, an emphasis on agent *teams* is appropriate, leading to frameworks where a system designer explicitly describes recipes for team behavior, with particular attention to which team members should interact, when, and how (Grosz, 1996; Tambe, 2000; Kinny, 1994).

When agents must formulate plans that fit together, but for which no existing recipes are available, techniques for reasoning about how actions of agents can enable or facilitate, or can hinder or even disable, actions of others, are needed (Decker, 1995). Merging the plans of agents, formulated individually, so as to permit the agents to successfully accomplish their activities without interfering with each other is also a useful technique (Georgeff, 1983; Ephrati, 1995; Clement, 1999).

3.5 More Dynamic

Whether viewed as a population of individuals or as a team, a multiagent system that operates in a dynamic task-environment must contend with changes in plans, goals, and conditions in the midst of execution. Tasks that previously could be carried out independently might now interact, such as when a resource becomes unusable forcing contention for other remaining resources. Agreements that have been forged between team members might have to be revisited as some team members change their priorities or recognize that their individual intentions, or those of the team as a whole, are no longer relevant in the new context they find themselves in.

Jennings (Jennings, 1992) has characterized these issues as the challenge in having conventions about what agents should do when they begin to question their commitments due to task-environmental dynamics. A variety of conventions can be specified, including the convention that seeks to ignore dynamics entirely by insisting that agents fulfill their commitments regardless. Alternatives include allowing agents to reneg on commitments if they pay some penalty, or permitting agents to abandon obsolete commitments provided that they notify team members (and thus potentially stimulate to formation of different commitments).

In fact, dynamic task-environments can suggest that agents should never view their (or others') plans as being anything more than tentative. Agents could unilaterally change their minds about their plans and begin acting on new plans before reaching agreement across the team. This has the potential of leading to inefficient collective activities due to information delays and to chain reactions (even race conditions) among changes. However, under some limiting assumptions about how and when agents can make unilateral changes, iterative coordination and execution techniques (e.g., (Durfee, 1991)) can lead to flexible coordinated behavior in dynamic task-environments.

3.6 More Distributed

Even when the interactions between agents requiring coordination are few and not undergoing dynamic changes, a task-environment can stress agents if the interactions requiring coordination are hard to anticipate. In particular, if agents are acting based on privately-held information about goals and methods, then it might take substantial effort to discover who is going to be interacting with whom.

One response to this is to anticipate all of the possible actions that agents might take, across all of the goals and plans that they might adopt, and to impose restrictions on what actions they can take under what conditions so as to prohibit undesirable interactions. Such “social laws” ensure that a law-abiding agent, acting in a population of other law-abiding agents, need never worry about undesirable interactions, no matter what goals and plans are being adopted (Shoham, 1994). In human terms, this is like saying that as long as all drivers obey traffic laws, then they can each eventually get to their desired destinations, wherever those are, without collision.

A second response is to support the process by which agents whose individual actions might interact can efficiently find each other. When interactions are over the exchange of goods, for example, providing agents with loci (auctions) for finding each other helps. Creating agents to represent resources over which agents might contend similarly allows interacting resource demands to be identified. Or agents might discover through experience others with whom they tend to interact, and form persistent aggregations (Azoulay-Schwartz, 2000; Brooks, 2000; Lerman, 2000).

Without identifiable contexts for aggregating, however, it could be that agents must somehow test for possible interactions against all other agents. This could be done through a centralized “coordinator” who collects together information on all agents, and using its global awareness can inform agents of the potential interactions to watch out for. In such a case, the coordinator should accept only as much information as is absolutely necessary to recognize interactions (Clement, 1999). Alternatively, agents could broadcast information to all others, so that each has sufficient awareness of the global picture. Through iterative exchanges, the overall system can cooperatively achieve its objectives (Lesser, 1981).

3.7 Greater Optimality/Efficiency

Coordination that is optimal is generally desirable, though less often feasible. As was mentioned earlier, coordination can sometimes be viewed as a search through the exponential number of combinations of agents’ alternative actions to find a “good enough” combination. Whereas sometimes it is enough to find a combination that does well enough (avoids conflicts among agents, or ensures eventually achieving goals), for some applications the optimal solution is sought. Optimality generally requires substantial computation (and sometimes communication) overhead; especially in dynamic task-environments (where optimal can become obsolete before it is carried out) or those with many agents and/or complex interactions, a satisficing or locally-optimal solution is often acceptable.

Nonetheless, for some restricted types of coordinated decisions, optimal might be within reach. An example commanding much attention in recent years has been in coordinating resource allocation decisions based on market-oriented approaches (Wellman, 1993). Through iterated rounds of bidding in an auction, agents can balance supply and demand to allocate resources to maximize their efficient use, under some assumptions. Active research is ongoing to extend these coordination strategies to "scale" them along other dimensions: not only to handle larger numbers of agents, but to handle higher degrees of interaction (using combinatorial auctions to allocate resources whose values are dependent on how they are acquired in combinations) and greater dynamics (including strategies for clearing auctions without waiting for all prices to settle) (Andersson, 2000; Fujishima, 1999).

Other methods for distributed rational decision making (Sandholm, 1999) include decision theoretic methods based on multiagent extensions of Markov Decision Processes (Boutilier, 1999). This type of method can find an optimal policy for a multiagent system, based on a particular coordination protocol that can be employed at runtime (for example, to increase agents' awareness of the global situation). When each agent follows its portion of the optimal policy, the expected utility of the multiagent system is maximized.

3.8 More Robustness

An optimal coordination solution might break when the world deviates from the coordination strategy's assumptions. Whether a coordination strategy can scale to domains where robust performance is difficult but necessary can thus become important.

One means of increasing the robustness of a coordination solution is to build a solution that contains sufficient flexibility that agents can work around new circumstances within their original coordination agreement. For example, building slack time into scheduled activities, or avoiding committing to details of exactly what will be done and when, can leave each agent with more room to maneuver when the world doesn't proceed according to plan. Typically, more robust coordination decisions are less efficient because they reserve resources for "fall-back" contingencies and therefore might suboptimally divide up tasks among agents for a particular situation. Coordination through organizational structures typically has this feature (Weiss, 1999, chapter 7; Prietula, 1998, chapter 3; Durfee, 1993).

Alternatively, a coordination strategy might expect to monitor the execution of its solution, and repair that solution as needed. These ideas are extensions of single-agent plan monitoring and repair/replan techniques. Teamwork models, with conventions as to how to respond when continued pursuit of joint commitments is senseless, are examples of this (Kumar, 2000). Moreover, in some cases it might be possible to develop generic monitoring and recovery methods for the coordination processes themselves (Dellarocas, 2000).

3.9 Lower Overheads

In application domains where communication channels are limited and where the computational resources available for coordination are minimal demand that attention be paid to reducing the overhead of coordination strategies. As communication bandwidth becomes more limited, for example, coordination decisions must be made without exchanging enough information to maintain a level of global awareness that many strategies might expect.

Techniques that involve the iterative exchange of increasingly detailed information about agents' plans and intentions provide one means of permitting time-constrained coordination, where the communication and computation overheads can be limited at the expense of the quality of the coordination solution (Clement, 1999). Alternatively, agents can choose to continue with outdated but still sufficient coordination decisions to avoid a chain reaction of coordination activities. When communication is at a premium, or might even be impossible, techniques such as using observations to model others, or using reasoning to converge on coordinated decisions (e.g., focal points) can pay dividends (Fenster, 1995).

Sometimes, the availability of coordination resources can be sporadic. Under some coordination regimes, agents can take advantage of opportunities where such resources are plentiful to build more complete models of the roles and contingent plans of each other, that can then be exploited when the agents have moved into situations where further communication and computation to coordinate is unsafe or infeasible (Durfee, 1999; Stone 1999).

4. OPEN CHALLENGES

I was initially inspired to write this piece because of what I saw as a trend toward identifying scaling to large numbers of agents as the most important challenge that can be posed to a multi-agent system. My own experience was that it was easy to develop multi-agent systems consisting of hundreds or thousands of agents, so long as those agents could merrily go about their business with no concern about the activities of others. On the other hand, it could be a tremendous challenge to develop a working system made up of only a handful of agents if the degree to which their activities needed to be dovetailed – and the penalty for failing to get the dovetailing exactly right – were both very high. The takehome messages of this article could thus be viewed as: (1) there are many ways to stress a coordination strategy, each of which pose research challenges and opportunities, and (2) there are already a variety of promising ideas out there for designing coordination strategies, that can be computationally realized, for getting agents to work well together under a broad range of circumstances.

The preceding whirlwind tour of some of the coordination strategies, and the kinds of stresses in agent population, task-environment, and solution criteria for which they are suited, should be viewed only as an introduction to the rich body of work that has gone into addressing the challenges of coordination in the many domains where it is needed. Many coordination strategies, and variations of coordination strategies, have been left out of the preceding. Interested readers should refer to recent books on the subject of multiagent systems (for example, (Weiss, 1999; Ferber, 1999; Wooldridge, 2000)) and to journals such as Autonomous Agents and Multi-Agent Systems (published by Kluwer) and proceedings of conferences such as the past *International Conference on MultiAgent Systems* and the current series of the *International Joint Conference on Autonomous Agents and MultiAgent Systems*.

I should also emphasize that, in the preceding survey, I was not intending that each coordination strategy be pigeonholed as only addressing issues along one of the dimensions. In fact, most can be scaled along multiple dimensions, but each has its limits. The challenge facing researchers in the field is to develop a better (preferably quantifiable) understanding of exactly how far different coordination strategies can scale along the dimensions laid out, as well as along dimensions that are still being identified as being germane to the application of intelligent agent systems to increasingly challenging problems.

5. ACKNOWLEDGMENTS

As part of DARPA's Control of Agent-Based Systems (CoABS), I have worked with several other researchers to understand the relative strengths and weaknesses of various coordination approaches. This article consolidates my take on many of those thoughts. While these colleagues should be held blameless for any oversimplifications and misrepresentations in this article, I'd like to acknowledge their contributions to my thinking along these lines. They include Craig Boutilier, Jim Hendler, Mike Huhns, David Kotz, James Lawton, Onn Shehory, Katia Sycara, Milind Tambe, and Sankar Virdhagriswaran. Milind Tambe also provided valuable feedback on an earlier version of this article. This work was supported, in part, by DARPA through Rome Labs (F30602-98-2-0142).

6. REFERENCES

Andersson, A., M. Tenhunen, and F. Ygge. "Integer Programming for Combinatorial Auction Winner Determination." *Proceedings of the Fourth International Conference on MultiAgent Systems (ICMAS-2000)*, pages 39-46, IEEE Computer Society Press, July 2000.

Azoulay-Schwartz, R. and S. Kraus. "Assessing Usage Patterns to Improve Data Allocations via Auctions." *Proceedings of the Fourth International Conference on MultiAgent Systems (ICMAS-2000)*, pages 47-55, IEEE Computer Society Press, July 2000.

Bojinov, H., A. Casal, and T. Hogg. "Multiagent Control of Self-reconfigurable Robots." *Proceedings of the Fourth International Conference on MultiAgent Systems (ICMAS-2000)*, pages 143-150, IEEE Computer Society Press, July 2000.

Boutilier, C. "Multiagent Systems: Challenges and opportunities for decision-theoretic planning." *AI Magazine* 20(4):35-43, Winter 1999.

Brooks, C. H., E. H. Durfee, and A. Armstrong. "An Introduction to Congregating in Multiagent Systems." *Proceedings of the Fourth International Conference on MultiAgent Systems (ICMAS-2000)*, pages 79-86, IEEE Computer Society Press, July 2000.

Clement, B. J. and E. H. Durfee, 1999. "Top-Down Search for Coordinating the Hierarchical Plans of Multiple Agents." In *Proceedings of the Third Conference on Autonomous Agents*, pages 252-259, May.

Decker, K. S. "TÆMS: A framework for analysis and design of coordination mechanisms." In G. O'Hare and N. Jennings, editors, *Foundations of Distributed Artificial Intelligence*, Chapter 16. Wiley Inter-Science, 1995.

Dellarocas, C. and M. Klein. "An experimental evaluation of domain-independent fault handling services in open multi-agent systems." *Proceedings of the Fourth International Conference on MultiAgent Systems (ICMAS-2000)*, pages 39-46, IEEE Computer Society Press, July 2000.

Durfee, E. H. and V. R. Lesser. "Partial Global Planning: A coordination framework for distributed hypothesis formation." *IEEE Transactions on Systems, Man, and Cybernetics* SMC-21(5):1167-1183, September 1991.

Durfee, E. H. "Organisations, Plans, and Schedules: An Interdisciplinary Perspective on Coordinating AI Systems." *Journal of Intelligent Systems* 3(2-4):157-187, 1993.

Durfee, E. H. "Distributed continual planning for unmanned ground vehicle teams." *AI Magazine* 20(4):55-61, Winter 1999.

Ephrati, E., M. E. Pollack, and J. S. Rosenschein. "A tractable heuristic that maximizes global utility through local plan combination." In *Proceedings of the First International Conf. on Multi-Agent Systems* (ICMAS-95), pages 94-101, June 1995.

Estlin, T., T. Mann, A. Gray, G. Rabideau, R. Castano, S. Chien and E. Mjolsness, "An Integrated System for Multi-Rover Scientific Exploration," *Proceedings of the Sixteenth National Conference of Artificial Intelligence (AAAI-99),* Orlando, FL, July 1999.

Fenster, M., S. Kraus, and J. S. Rosenschein. "Coordination without communication: experimental validation of focal point techniques." *Proceedings of the First International Conf. on Multi-Agent Systems* (ICMAS-95), pages 102-108, June 1995.

Ferber, J. *Multi-Agent Systems: An Introduction to Distributed Artificial Intelligence*. Addison-Wesley, Harlow England, 1999.

Fujishima, Y., Leyton-Brown, K., and Shoham, Y. "Taming the computational complexity of combinatorial auctions: Optimal and approximate approaches." In *Sixteenth International Joing Conference on Artificial Intelligence (IJCAI-99)*. 1999.

Gasser, L., C. Braganza, and N. Herman. "MACE: A flexible testbed for distributed AI research." In M. N. Huhns, editor, *Distributed Artificial Intelligence*, pages 119-152, Pitman Publishers, 1987.

Georgeff, M. P. "Communication and Interaction in multi-agent planning." In *Proceedings of the Third National Conf. on Artificial Intelligence* (AAAI-83), pages 125-129, July 1983.

Gray, R. S., D. Kotz, R. A. Peterson, Jr., P. Gerken, M. Hofmann, D. Chacon, G. Hill, and N. Suri. "Mobile-Agent versus Client/Server Performance: Scalability in an Information-Retrieval Task." Technical Report TR2001-386, Dept. of Computer Science, Dartmouth College, January 2001.

Grosz, B. J. and S. Kraus. "Collaborative Plans for Complex Group Action." *Artificial Intelligence*. 86(2), pp. 269-357, 1996.

Jennings, N. R. "Commitments and Conventions: The foundation of coordination in multi-agent systems." *The Knowledge Engineering Review*, 2(3):223-250, 1993.

Jennings, N. R. "An Agent-based Approach for Building Complex Software Systems." Communications of the ACM 44(4):35-41, April 2001.

Kinny, D., M. Ljungberg, A. S. Rao, E. Sonenberg, G. Tidhar, and E. Werner. "Planned Team Activity." In C. Castelfrachi and E. Werner, editors, *Artificial Social Systems*. Springer-Verlag, Amsterdam, 1994.

Kumar, S., P. R. Cohen, and H. J. Levesque. "The adaptive agent architecture: Achieving fault-tolerance using persistent broker teams." *Proceedings of the Fourth International Conference on MultiAgent Systems (ICMAS-2000)*, pages 159-166, IEEE Computer Society Press, July 2000.

Lansky, A. L. "Localized Search for Controlling Automated Reasoning." In *Proceedings of the DARPA Workshop on Innovative Approaches to Planning, Scheduling, and Control*, pages 115-125, November 1990.

Lerman, K. and O. Shehory. "Coalition Formation for Large-Scale Electronic Markets." *Proceedings of the Fourth International Conference on MultiAgent Systems (ICMAS-2000)*, pages 167-174, IEEE Computer Society Press, July 2000.

Lesser, V. R. and D. D. Corkill. "Functionally accurate, cooperative distributed systems." *IEEE Trans. on Systems, Man, and Cybernetics* SMC-11(1):81-96, 1981.

Paolucci, M., Z. Niu, K. Sycara, C. Domashnev, S. Owens and M. van Velsen "Matchmaking to Support Intelligent Agents for Portfolio Management." In *Proceedings of AAAI2000*.

Prietula, M. J., K. M. Carley, and L. Gasser, editors. *Simulating Organizations: Computational Models of Institutions and Groups.* AAAI Press/MIT Press, Menlo Park, CA, 1998.

Sandholm, T. W. "Distributed Rational Decision Making." Chapter 5 in (Weiss, 1999).

Shoham, Y. and M. Tennenholtz. "On Social Laws for Artificial Agent Societies: Off-line design." *Artificial Intelligence* 72(1-2):231-252, 1994.

Stone, P. and M. Veloso. "Task Decomposition, Dynamic Role Assignment, and Low-Bandwidth Communication for Real-Time Strategic Teamwork." *Artificial Intelligence* 100(2), June 1999.

Tambe, M., W. L. Johnson, R. M. Jones, F. Koss, J. E. Laird, P. S. Rosenbloom, and K. Schwamb. "." *AI Magazine*, 16(1):15-39, Spring 1995.

Tambe, M., and W. Zhang. "Towards flexible teamwork in persistent teams: extended report." *Journal of Autonomous Agents and Multi-agent Systems* 3(2): 159-183, June 2000.

Weiss, G., editor. *Multiagent Systems: A modern approach to distributed artificial intelligence.* The MIT Press, Cambridge MA, 1999.

Wellman, M. P. "A market-oriented programming environment and its application to distributed multicommodity flow problems." *Journal of Artificial Intelligence Research*, **1**:1-23, 1993.

Wolfram, S. *A New Kind of Science*. Wolfram Media Inc., 2002.

Wooldridge, M. *Reasoning About Rational Agents*. The MIT Press, July 2000.

Yokoo, M. and Hirayama, K. "Algorithms for distributed constraint satisfaction: A review." *Autonomous Agents and Multi-agent Systems*, 3(2):189-211, 2000.