

Strategies for Discovering Coordination Needs in MultiAgent Systems*

Edmund H. Durfee

Computer Science and Engineering Division
EECS Department, University of Michigan
Ann Arbor, MI 48109
[durfee@umich.edu](mailto:edurfee@umich.edu)

Abstract

While numerous techniques have been developed by which agents can coordinate over the allocation of scarce resources or the pursuit of interdependent goals, less is understood about how agents discover, in the first place, with whom they should worry about coordinating and about what. We have studied several strategies for making such discoveries, ranging from communicating abstract information to anticipate potential interactions ahead of time, to learning from interactions as they occur to anticipate future interactions. In this paper, we briefly summarize some of the strategies we have been investigating, and opportunities for further exploration.

1. Introduction

When multiple computational agents share a task environment, interactions between the agents generally arise. An agent might make a change to some feature of the environment that in turn impacts other agents, for example, or might commandeer a non-sharable resource that another agent desires. When the decisions that an agent makes might affect what other agents can or should decide to do, agents will typically be better off if they coordinate their decisions.

Numerous techniques exist for coordinating decisions about potential interactions. These include appealing to a higher authority agent in an organizational structure, instituting social laws that avoid dangerous interactions, using computational markets to converge on allocations, explicitly modeling teamwork concepts, using contracting protocols to strike bargains, and iteratively exchanging tentative plans until all constraints are satisfied. There is a rich literature on these and other mechanisms for coordinating agents; the interested reader can see [6].

However, each of these mechanisms takes as its starting point that the agents requiring coordination know, at the outset, either with whom they should coordinate, or what issues they should coordinate about. As examples, an organizational structure inherently defines how agents are related to each other, and a computational market corresponds to some resource or “good” that was somehow known to be contentious.

A central thrust of our research is in pushing back the boundaries of what is assumed known in a multiagent setting in order to bootstrap the coordination process. That is, we want to develop techniques by which agents can discover whom they should coordinate with, or what they should coordinate about, so that the rich variety of coordination techniques can then be employed. This paper briefly summarizes some of our progress, results, and plans on this front.

2. Unintended Conflicts

An important case in which agents need to discover coordination needs is the following. Agents occupy an open, dynamic environment, and each agent has its own independent objectives. Yet, in pursuing its objective, an agent can unintentionally interfere with others, sometimes catastrophically. Therefore, it

* This article was originally published in the DoD Software Tech News, Volume 5, Number 1, www.dacs.dtic.mil.

is important for each agent to discover whether something it is doing needs to be coordinated with others.

We have been studying coalition operations as an example application domain where this kind of problem arises. In a coalition, objectives and responsibilities are distributed among multiple functional teams, where operational choices by one team can infrequently and unintentionally affect another team. The repercussions of unintended interactions can range from merely delaying the accomplishment of objectives (such as waiting for assets that were unexpectedly borrowed by someone else) to more catastrophic outcomes (such as so-called friendly fire). We have been developing computational techniques in which each team is represented by a computational agent, and these agents predict the unintended interactions and resolve them before they occur. The resulting coordinated plans of the agents should be efficient (e.g., agents should not have to wait unnecessarily for others), flexible (e.g., agents should retain room in their plans to improvise around changing local circumstances), and realizable (e.g., agents should not have to message each other at runtime in a manner that outstrips communication capabilities).

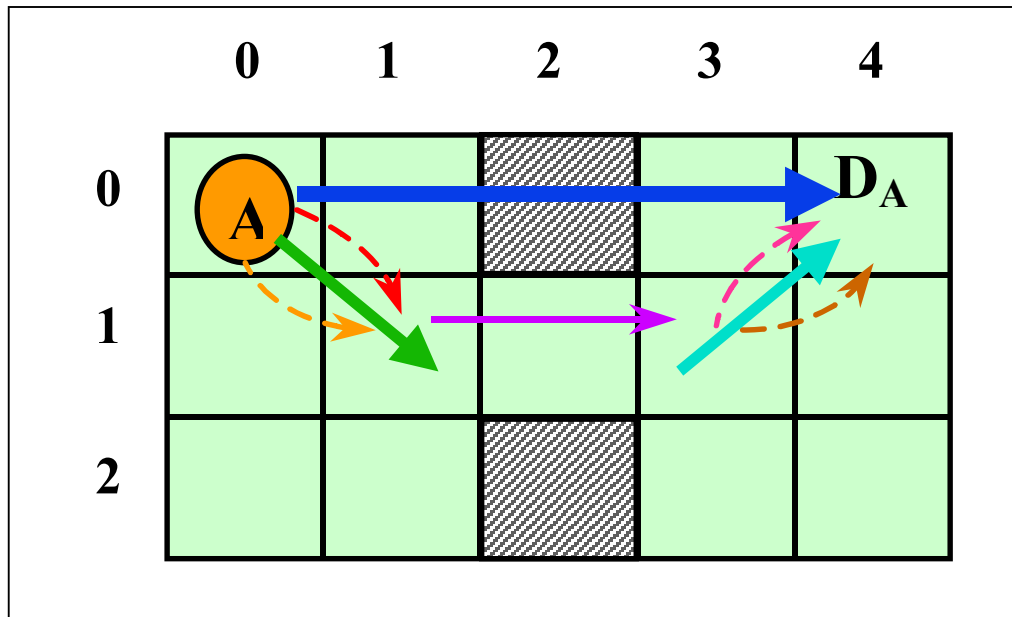
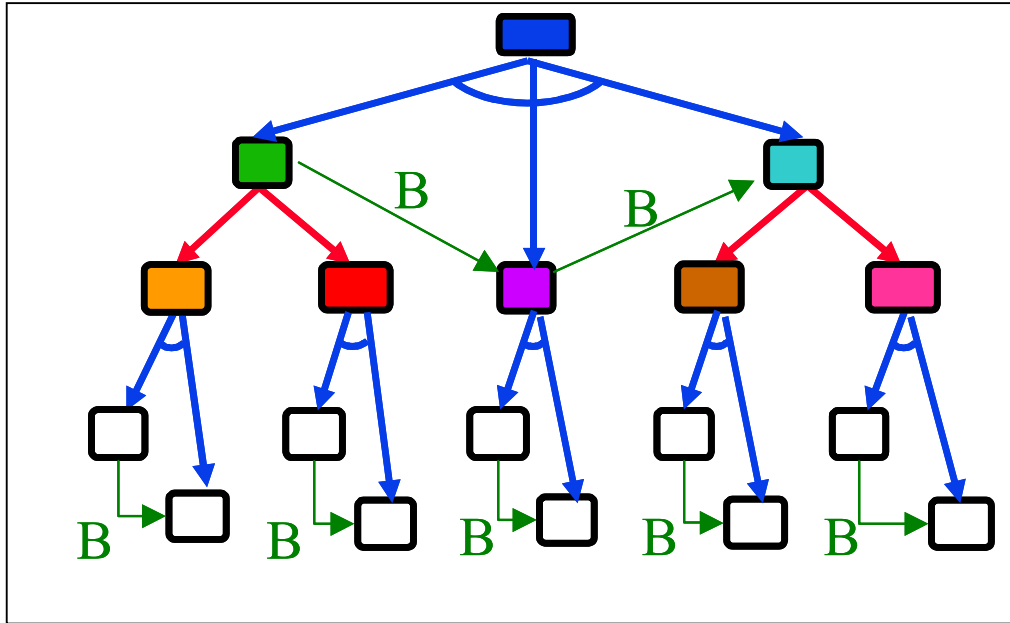


Fig. 1. Example Movement Task

Conceptually, our techniques begin by assuming that each agent can represent its plans in a hierarchical task network (HTN), capturing the possible decompositions of abstract plan steps into more detailed plans. As a simple example, consider the case of agent A moving through a grid world to reach a destination (Figure 1). The HTN for this agent is in Figure 2. At the most abstract level (wide upper arrow in Figure 1, top node in the HTN), the plan is simply to go from the initial location to the destination. This is in turn composed of the three sequential steps of going to the door, through the door, and beyond the door (the other solid arrows in Figure 1, the second tier of nodes in Figure 2). The ordering constraints are captured in the HTN (Figure 2) by the arrows labeled “B” for “before.” For both the first and last step at this level, there are two ways of accomplishing the step. For example, for getting to the door, the upper route or the lower route could be chosen. Each of these in turn can be decomposed into a sequence of two movements; the upper, for example, is to the right and then down.

An advantage of using the hierarchical representation is that each agent has, simultaneously, a model of itself at multiple levels of detail. In an open environment populated by numerous agents, being able to communicate about and exchange abstract information can enable agents to quickly determine which (small) subset of agents in the world they actually could potentially interact with (Figure 3a to Figure 3b). In the simple movement task example, for instance, the grid might be much larger, and the subset of agents is small whose planned movements, even abstractly defined, indicate a potential collision with agent A. For those agents, it might even be possible to impose constraints at the abstract level to ensure against unintended collisions, such as sequentializing the overall plans so that only one of the affected agents moves at a time. Or, for the remaining agents, additional details of the HTNs can be exchanged. As a result, agents that were potentially interacting might be determined to not interact at

all, reducing the number of agents further and introducing constraints between only substeps of plans leaving agents to do their other substeps as they wish (Figure 3c). Finally, further investigation might indicate that the potential conflict can be isolated to a particular choice that an agent might make; a



commitment by the agent to forbid that choice leaves the plans coordinated without imposing any ordering constraints between the agents' plans at all (Figure 3d).

Fig. 2. Example Hierarchical Task Network (HTN)

This example illustrates that, by working from the top down, agents can more efficiently identify and zoom in on the problematic interactions. By digging down deeply, they might be able to impose commitments (on relative timing or on choices of ways in which they will accomplish their tasks) that lead to very crisp coordination. However, in dynamic environments, sometimes it is better to impose constraints at more abstract levels: while this might require more sequential operation than desired, it also allows agents to avoid commitments to details that they might regret. As is intuitive in human coordination, each agent retains more flexibility for improvising when it makes more vague commitments to others. Moreover, digging down deeply requires more rounds of communication and analysis, so coordinating at abstract levels incurs less overhead. Among our ongoing research activities are developing methods for quantitatively evaluating tradeoffs between coordination “crispness”, overhead, and flexibility.

We have developed techniques for formulating summaries in HTNs that permit the kind of top-down reasoning that we have just described, and have shown that such techniques can indeed much more efficiently coordinate agents [3]. These techniques have been shown to be sound and complete. At the cost of completeness, we have also developed a version of these techniques that can be used on-line [5]. The on-line techniques allow agents to postpone decisions about which of the alternative ways they will use to accomplish a task until that task is the next to be done. This in turn provides increased flexibility to the agents, leading to more reliable agent operation in dynamic domains than methods that require agents to make selections before execution begins.

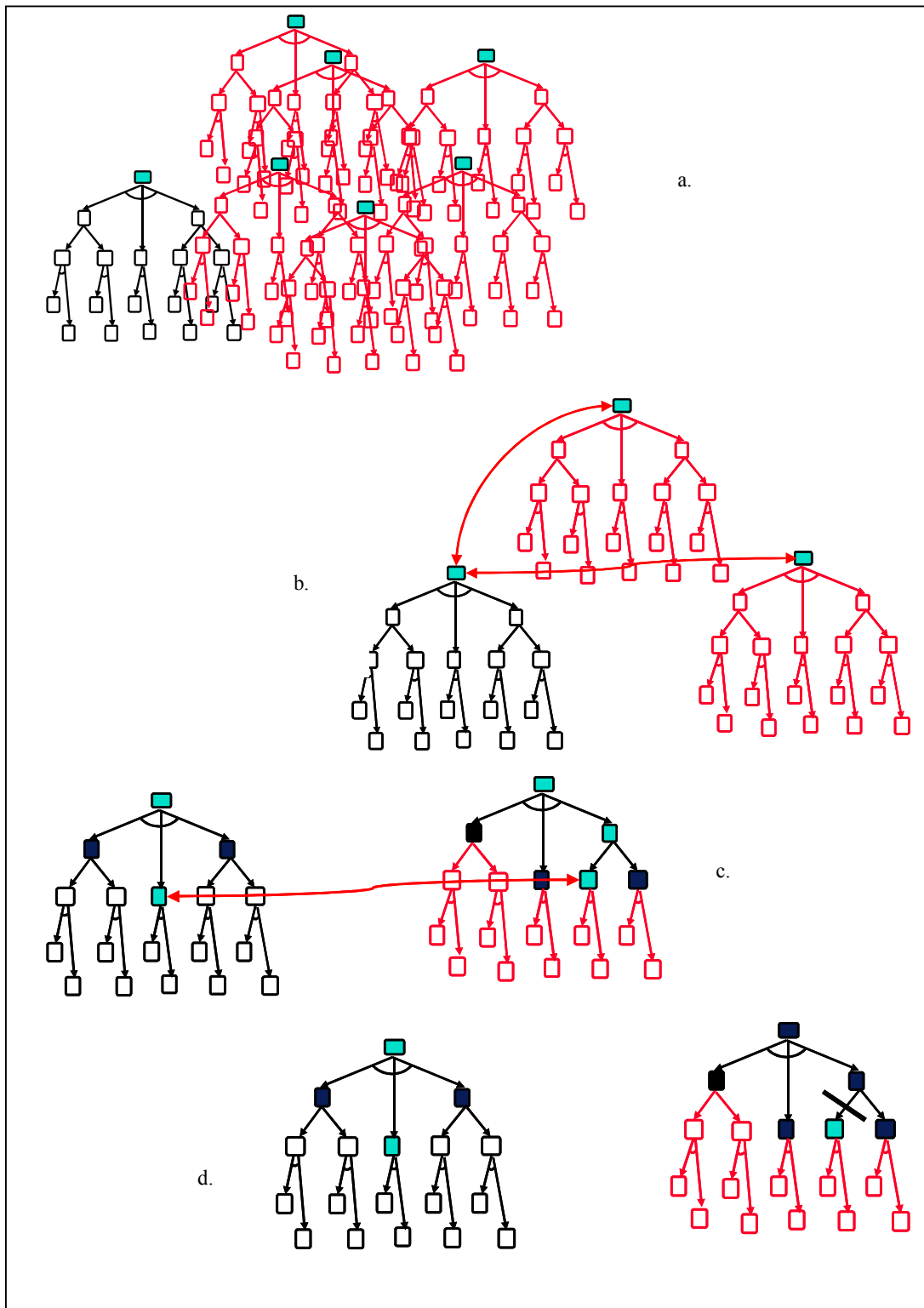


Fig. 3. Top-Down Coordination Protocol Example

3. Dealing With Centralization

The techniques just outlined have the feature that, to ensure that all possible interactions are detected and dealt with, some agent or agents need to compare all agents' most abstract plans. This implies that, at some point, information about all agents needs to be known in one place, which is antithetical to decentralized multiagent systems. Certainly, our current implementations rely on a central coordinator to discover potential agent interactions, although in principle once these are discovered the job of working with the agents to resolve the interactions can be delegated to multiple sub-agents, where each handles a different partition of the agent population.

How can we get around the need for centralization? Well, first of all, it should be noted that our use of centralization for detecting interactions does not imply that authority, or even knowledge about agent preferences, is centralized. In our model, the coordination process merely detects potential interactions and finds possible resolutions (more detailed resolutions as time goes on). To agree upon which resolution to use, the affected agents can employ any of the various coordination mechanisms mentioned at the beginning of this paper. That is, these are appropriate once the agents know about the interactions and who is involved.

In turn, this suggests that one way of eliminating the centralization of the detection process requires that agents are initialized with some knowledge. For example, the organizational structure in which they reside might inherently partition the agents, such that coordination can be carried out in parallel in different partitions. Or agents might be initialized with knowledge of the possible actions of other agents that can be used to anticipate interactions. For example, our research is using these ideas to coordinate resource-limited agents in a multiagent world [4]. In the simplest sense, a resource-limited agent needs to decide how to allocate its limited capabilities in order to meet its performance goals across the scope of worlds that it might encounter. By employing knowledge about what actions other agents might take in particular situations, it can better predict what worlds it might encounter, and can even use its uncertainty to focus communication with those other agents to ask them which of the alternative actions they plan to take for a critical situation. Such communications could also permit agents to avoid taking redundant actions in situations where they would react the same way. In the long run, agents can even engage in negotiation to convince others to favorably change how they react to particular circumstances.

4. Congregating Over Mutual Concerns

An alternative means of determining coordination needs, instead of centralizing information or inherently distributing key coordination knowledge, is to instead permit coordination needs to be discovered through interactions. While this would be inappropriate in applications where uncoordinated interactions could be catastrophic (such as when friendly fire arises in coalition operations), there are many applications where the consequences of poor coordination are not so dire.

Consider, for example, interactions among groups of people with similar interests, such as in an electronic newsgroup. A well-defined group permits an efficient exchange of relevant information among interested people, with a minimum of tangential communications that waste readers' time. A poorly-defined group, on the other hand, wastes readers' time and might lose readership quickly, but with no significant lasting effects on the participants. In this case, then, it is possible that people might congregate around newsgroup topics in an emergent way, through experimentation and exploration in the space, until they converge on relatively stable newsgroups that lead to productive interactions.

We have been conducting research in understanding the dynamic processes of congregating in open environments [2]. In our model, agents move among congregations until they find places where they are satisfied, where satisfaction depends on the other members of their congregation. Since these other agents are also moving around to find satisfactory congregations, agents are engaged in non-stationary ("moving target") search. In general, convergence in such systems is slow if it happens at all, and we have been studying mechanisms that enhance convergence such as: varying the movement costs of agents so that some "hold still" while others move; allowing like-minded agents to move as a coalition; giving agents the ability to remember and return to previously-experienced congregations; and allowing agents in a congregation to summarize their common interests and advertise this information to other agents.

As a specific form of congregating, we have been particularly interested in information economies, where competing producers of information goods must bundle and price their goods so as to attract (a subset of) the information consumers. Where a producer ends up in the product-and-price space is influenced not only by the consumer preferences, but also by the positioning decisions of other producers. Among our research results are that we have defined some of the conditions that promote the discovery of niche markets in the information economy, such that producers engage in stable relationships with an interested subset of the consumer population, and avoid mutually-harmful interactions (price wars) with other producers [1]. As suggested above, the price paid for this decentralized technique for discovering which agents should coordinate (interact) with each other is that, on the way to the ultimate mutually-profitable result, producers will sometimes compete with each other and do poorly temporarily as a result.

5. Summary and Future Directions

In this paper, we have claimed that, while powerful techniques exist for coordinating agents that already know whom or about what to coordinate, there are still many issues that need to be explored in designing efficient mechanisms by which to determine what needs to be coordinated in the first place. We briefly described some mechanisms that we are exploring for this purpose. One of these involves agents iteratively exchanging plan information at increasingly detailed levels to isolate potential interactions and impose effective commitments to resolve conflicts. Another, on the other hand, permits suboptimal interactions to occur, and allows the agent population to self-organize, over time, into congregations that emphasize beneficial interactions.

There are many directions in which we are, or are considering, extending these research activities. We need to develop heuristic means by which agents can decide on the level of detail at which they should coordinate, and metrics for comparing alternative coordination decisions in uncertain environments. We need to extend the soundness and completeness proofs, as well as the complexity analyses, of the techniques as we continue to augment and improve them. Coordination commitments that are derived between agents should be generalized and remembered to form the core of a suite of team plans, and the processes by which coordination needs are discovered should apply not only between agents but also between agent teams. Finally, these techniques need to be implemented and evaluated in the context of challenging applications, such as in the domain of coordinating coalition operations.

References

1. Christopher H. Brooks, Edmund H. Durfee and Aaron Armstrong. "An Introduction to Congregating in Multiagent Systems." In *Proceedings of the Fourth International Conference on MultiAgent Systems (ICMAS-2000)*, pages 79-86, July 2000.
2. Christopher H. Brooks, Edmund H. Durfee and Rajarshi Das. "Price Wars and Niche Discovery in an Information Economy." In *Proceedings of the ACM Conference on Electronic Commerce 2000 (EC-00)*, October 2000.
3. Bradley J. Clement and Edmund H. Durfee. "[Theory for Coordinating Concurrent Hierarchical Planning Agents Using Summary Information](#)." In *Proceedings of the National Conference on Artificial Intelligence (AAAI-99)*, pages 495-502, July 1999.
4. Haksun Li, Edmund H. Durfee, and Kang G. Shin "Multiagent planning with internal resource constraints." To appear in *Proceedings of the AAAI 2002 Workshop on Planning With and For MultiAgent Systems*, July, 2002.
5. Pradeep M. Pappachan and Edmund H. Durfee. "A satisficing multiagent plan coordination algorithm for dynamic domains." (abstract) *Proceedings of the ACM Conference on Autonomous Agents (Agents-01)*, June 2001.
6. Gerhard Weiss (editor). *Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence*. MIT Press: Cambridge Massachusetts, 1999.

Acknowledgements

The ideas and results described in this paper were developed with numerous collaborators. In particular, I'd like to thank my students, including Brad Clement, Pradeep Pappachan, Chris Brooks, Haksun Li, and Jeff Cox. The work was supported, in part, by DARPA under the Control of Agent-Based Systems Initiative (F30602-98-2-0142), by DARPA under the Automated Negotiating Teams Initiative (subcontract to Honeywell on F30602-00-C-0017), and by NSF grant IIS-9872057.