Multiagent Systems to Support Coordinated Emergency Management

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

1. Introduction

Responding well in emergency situations is difficult for a variety of reasons. When the emergency transpires across a wide geographical region (such as occurs with floods, earthquakes, storms, etc.), the sheer amount of information about the emergency flowing through the network can overwhelm decision makers, masking critical events, needs, and trends. Further, all this information is inherently uncertain and dynamic, leading to inconsistent views at different times and places that trigger competing and often incompatible responses. Because emergencies are rare (thankfully), there is little in the way of "standard" responses to fall back on; instead, managers of the response must try to envision the many possible trajectories of how the emergency will play out, and how alternative responses will affect those trajectories. Finally, the tempo at which decisions must be made is daunting, since minor hesitations in reaction can direly affect the loss of life and property.

Multiagent systems hold the promise to help improve the quality and speed of decisions under such conditions. Conceptually, a computational "agent" is a process tasked to achieve goals on behalf of a user, augmenting the user's capabilities and, if necessary, acting in place of a user who is unavailable or focused on other decision tasks. This is analogous to how a (human) theatrical or sports agent is tasked with acting on behalf of a performer/athlete during contract negotiations. Multiagent systems are thus networks of such computational agents, interacting with each other to achieve outcomes that benefit their associated users.

As computational entities, agents have capabilities that can complement the abilities of human users. For example, agents can store and rapidly retrieve vast amounts of information. They can quickly project forward along different hypothetical future trajectories and keep track of alternative "what-if" scenarios. They can perform mundane, routine duties for monitoring and evaluating situations, freeing up their users' attention for problems requiring human insight and perspective. And, working together, agents can jointly search through alternative combinations of actions that their users might concurrently pursue to find good (and in some cases optimal) joint responses.

While these advantages of incorporating multiagent systems into emergency management applications sound compelling, realizing these possible benefits in practice, and especially being able to count on them when lives are on the line, will require advances along a number of fronts. One particular challenge, which is the focus of this paper, is in developing computational techniques for distributed agents to use that will strike an appropriate balance between preserving the autonomy of a user to respond to emergent events while promoting timely, orchestrated actions that accomplish collective goals.

Thus, the position taken in this paper is that multiagent systems, distributed among participants in emergency management operations, can work in the background to improve group performance by automating the process of finding appropriate models for participants to have of each other. This allows participants to focus their attention on the interactions most critical to joint success. To support this position, the remainder of this paper summarizes a few examples of application domains and prototyped technologies that feature the use of multiagent systems to help coordinate their users' activities.

2. Coordination of Coalitions

Independent entities can form a coalition to collectively achieve objectives that they cannot individually accomplish. A fundamental challenge in coalition operations, however, is in smoothly integrating the activities of disparate entities. Each participant in a coalition will have its own perspectives, priorities, and standard operating procedures, and it becomes all too easy for these to collectively lead to inefficient, counterproductive, and sometimes even dangerous joint behaviors. Coordination in a coalition is therefore critical, but difficult because entities might not want to reveal too much about their inner workings, might wish to maximize their independence, and might lack the time and desire to understand each other deeply.

Multiagent techniques can help improve coalition coordination. DARPA's Control of Agent-Based Systems program sponsored the Coalition Agents Experiment (CoAX) [1] earlier this decade, which developed an integrated system of agent technologies to support peacekeeping operations in a fictitious scenario, a simplified version of which is shown in Figure 1. In this scenario, several coalition partners are cooperating as a peacekeeping force in a country called Binni. Among the activities of the coalition forces are maintaining observation posts, delivering humanitarian aid, and enforcing a total exclusion zone (TEZ) to keep enemy combatants apart. Since different countries are responsible for different objectives, potential inefficiencies can occur (such as when troop movements and humanitarian aid deliveries moving along the same route get in each others' way). More critically, failure to coordinate can lead to catastrophic friendly-fire incidents (such as when aircraft enforcing the TEZ fire on partner troops that are moving to observation posts).

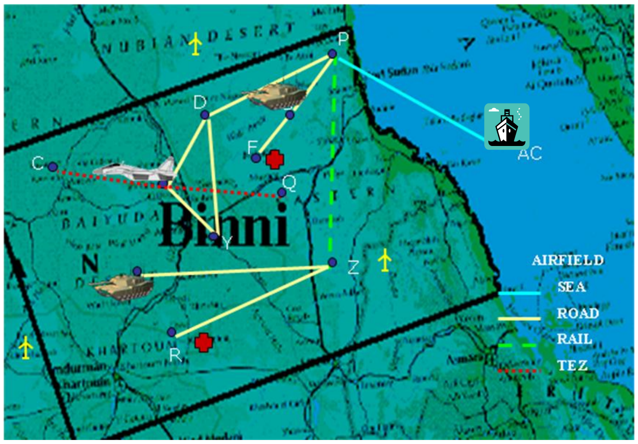

Figure 1. Simple Coalition Coordination Problem

The challenge that the coalition partners face is finding the right information about themselves to project to others to ensure sufficient coordination without revealing sensitive information, without flooding each other with irrelevant details, and without unnecessarily locking themselves into inflexible plans that could become obsolete as domain dynamics evolve. Unfortunately, there is no single static description of local activities that works well in all scenarios, or even between different groups of agents in the same scenario. Instead, participants need to find the right level of detail at which to coordinate their activities in the current circumstances.

If we frame this as a search problem, multiagent technology can be brought to bear. The search is over the space of alternative modeling levels at which agents can coordinate, to find the level that is best suited to their current needs. To create the space of modeling levels, we capitalize on a valuable side-effect of hierarchical planning mechanisms, such as Hierarchical Task Networks (HTNs) [2]. HTNs generate agent plans by iteratively decomposing higher-level tasks into increasingly primitive tasks, until the expansion results in a sequence of primitive actions that is expected to achieve the sought-after goal.

The insight our approach uses is that the intermediate levels of plan abstraction summarize, at varying levels of detail, what the agent's plan is. Thus, if agents can reason about how their abstract actions might interact, they could detect possible reasons to coordinate using smaller abstract plans (rather than sharing detailed plans). Further, if they resolve potential interactions at the abstract plan level, then they can elaborate (and revise) their local plans independently and flexibly as their local circumstances warrant.

The ability to identify possible agent interactions and their resolutions based on more abstract actions depends critically on having sufficient models of what those actions might mean when they are elaborated. Our work has defined a process by which agents can compute summary information for intermediate activities that ensures that possible interactions are never overlooked [3]. With these models, an agent engages in a top-down coordination search. First, it compares its most abstract plan with those of others, and immediately prunes away unrelated agents (which, for many applications where agents have geographic or functional locality, will be most others). For each of the agents with which it might interact, the agent can

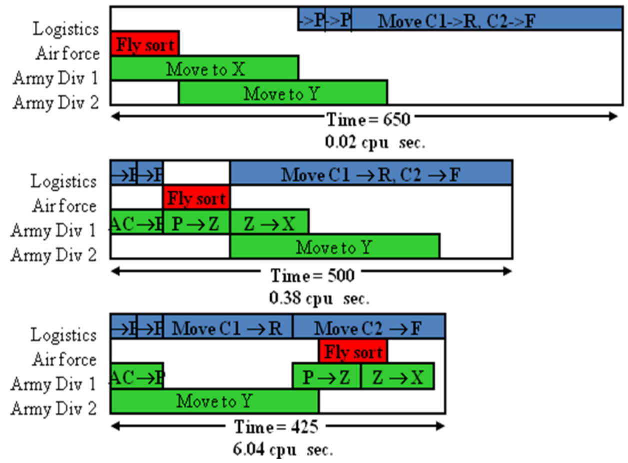

Figure 2. Multiple Alternative Coordination Levels

either resolve the interaction at the abstract level, or exchange plans at the next deeper level of detail to further understand and isolate the interaction. This process repeats separately for each combination of agents until all interactions have been resolved.

Figure 2 portrays a series of three levels of coordination along this search space. The top level involves abstract actions for each of the entities, and resolves interactions very quickly (in 0.02 cpu seconds) by imposing ordering constraints that tend to largely sequence the actions for a makespan of 650. As we progress down to more detailed levels, the agents' plans are (if possible) broken into more primitive actions, allowing interactions to be pinpointed more accurately. The greater the detail is, the shorter the makespan becomes, because greater amounts of concurrency can be safely achieved. However, greater amounts of computation are needed for coordination as the number of agent actions being reasoned about grows.

While this illustration shows the tradeoff between the benefits of better coordination (shorter makespans) and its reasoning costs (rising cpu times), what it does not explicitly show is the impact on local flexibility. As the agents work downwards into more detailed plan decompositions and make promises about what more specifically they will be doing and when, they lock themselves into more stringent commitments that leave them less wiggle room in case something goes wrong. Some flexibility is important.

This is a critical concern in emergency management, where different participants might possess their own capabilities and priorities, and will be reluctant to sacrifice some of their autonomy to work with other groups whose responsibilities might differ. Yet, stovepiped, segregated responders coordinate poorly and slowly, if at all. The trouble is that there is no static level for coordination that fits all situations, and renegotiating relationships in the midst of an emergency distracts from critical activities. It is the position of this paper that adapting technologies like those described for coalition operations could potentially prove valuable, allowing agents operating in the background to weigh the benefits and risks of coordinating at different levels of detail, in order to help organizations converge more efficiently on levels that strike the right balance between integration and autonomy.

3. Coordination of Distributed Teams

The DARPA Coordinators program provides another example of using multiagent systems to support coordination between human entities. In contrast with CoAX, the Coordinators program assumed that the users' goals were fully aligned—they were on a team and their individual successes depended entirely on the team's success. Yet, because the people might be dispersed and facing different local challenges, there is still great advantage in coordinating at an abstract enough level to give individuals latitude for improvisation in changing circumstances.

Because the humans being coordinated were part of a team, in this application it was assumed that the multiagent system would know from the outset what the teamwork interactions were, rather than having to discover them as in CoAX. The Coordinators application also assumed a highly stochastic environment. For example, one scenario involved subteams simultaneously entering several locations suspected of holding a hostage, where those locations might be in very different areas (urban, remote, at sea, etc.). As each subteam moved towards its assigned location, it could be delayed, forcing other subteams to adjust their movements. Further, a subteam's capabilities could degrade (loss of personnel or material) or its objectives change (new orders received). Meanwhile, to avoid detection, radio contact should be minimized.

Not surprisingly, we again focus on the question of how these units should model each other, and in particular how the computational agents embedded with the units should help create, update, manage, and utilize models of others' activities. Designating a single, central controller is infeasible not only because of the inherent risk (single point of failure) and scalability (computational bottleneck) concerns, but also because of the delays that it would impose on agents being able to respond quickly to local events. Instead, agents should exploit periods of connectivity to form and update commitments to each other regarding their interactions, and then individually adapt their execution policies to their situations while continuing to ensure that they adhere to their commitments.

In essence, an agent's commitments represent an abstract model of itself (when it will accomplish tasks that others are counting on) and others (when they will meet its interaction needs). Our research (Figure 3) has investigated techniques that agents can use to tractably decide which commitments to make to each other, and how to maximize local performance while still satisfying commitments [4]. Further, when circumstances conspire such that an agent fails to meet a commitment it has made, alternative courses of commitments can be triggered, essentially implying that agents have contingent policies not only over their planned activities, but also over their commitments to each other.

We have implemented these techniques in agents that model their coordination problems as a form of decentralized POMDP [5]. Our results to date suggest that, for teams of agents who individually have complex tasks and with relatively sparse relationships between different agents' tasks, searching in the space of inter-agent commitments rather than in the detailed joint policy space can lead to considerable speedups, and helps agents retain greater flexibility over their own activities. That teamwork in some emergency management situations has similar characteristics suggests that associating with human responders agents that perform commitment-based coordination could help improve both teamwork and responsiveness.

4. Discussion

In this paper, we have argued that multiagent systems could provide a valuable decision-support infrastructure for managing emergency responses, and have illustrated the use of multiagent systems to help people coordinate their activities both in coalition and in teamwork settings. This only scratches the surface of ideas from the multiagent community that could find use in emergency management. Other ideas include tasking automated agents to monitor features of a situation on the user's behalf [6], negotiating task assignments among teams [7], and using economic principles to optimize the allocation of joint resources across agents [8].

5. REFERENCES

[1] D. N. Allsopp, P. Beautement, J. M. Bradshaw, E. H. Durfee, M. Kirton, C. A. Knoblock, N. Suri, A. Tate, and C. W. Thompson. "Coalition Agents Experiment: Multiagent Cooperation in International Coalitions." *IEEE Intelligent Systems*, 17(3):26-35, May/June 2002.

[2] K. Erol, D. Nau, and J. Hender. "UMCP: A Sound and Complete Planning Procedure for Hierarchical Task-Network Planning." 1996. In *Proc. Int. Conf on AI Planning and Scheduling*, 1994.

[3] B. J. Clement, E. H. Durfee, and A. C. Barrett. "Abstract Reasoning for Planning and Coordination," *J. of Artificial Intelligence Research (JAIR)*, vol. 28, pp. 453-515, 2007.

[4] S. J. Witwicki and E. H. Durfee. "Commitment-Based Service Coordination," *Int. Journal on Agent Oriented Software Engineering*, 3(1):59-87, 2009.

[5] C. V. Goldman and S. Zilberstein. "Decentralized Control of Cooperative Systems: Categorization and Complexity Analysis." *J. Artif. Intell. Res.* (JAIR) 22: 143-174, 2004.

[6] E.H. Durfee, M. J. Huber, M. Kurnow, and J. Lee. "TAIPE: Tactical Assistants for Interaction Planning and Execution." In *Proc. First Int. Conf. on Autonomous Agents*, pages 443-450, February 1997.

[7] H. Li, E. H. Durfee, and K. G. Shin. "Multiagent Planning for Agents with Internal Execution Resource Constraints." In *Proc. Second Int. Joint Conf. on Autonomous Agents and Multiagent Systems*, pages 560-567, July 2003.

[8] D. A. Dolgov and E. H. Durfee. "Resource Allocation Among Agents with MDP-Induced Preferences," *J. of Artificial Intelligence Research*, 27:505-549, 2006.

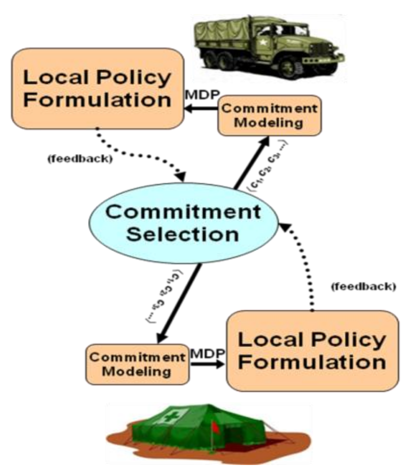

Figure 3: Iterative Commitment Formation Process