

Multi-Agent Systems for the Real World^{*}

(Extended Abstract)

Rajiv Maheswaran, Craig Milo Rogers, Romeo Sanchez and Pedro Szekely
Information Sciences Institute, University of Southern California
Marina Del Rey, CA 90292

Gergely Gati, Kevin Smyth and Chris VanBuskirk
Institute for Software Integrated Systems, Vanderbilt University
Nashville, TN 37235

1. INTRODUCTION

To support real world applications, a multi-agent system must deal with all relevant issues from the environment, the problem, the users and the technological infrastructure. The real world introduces restrictions and noise at all levels. The communication network may provide intermittent connectivity, variable latency and limited bandwidth. Devices have limited computation. The world is uncertain and dynamic. Tasks, resources and the team objective can change. The software may give incorrect advice. Users can make errors and initiate unplanned actions.

Most multi-agent system work focuses on the model-to-algorithm-to-policy aspect of the problem. This assumes that a correct model of the world is given, and the objective is to design the best policy representation and algorithm so that if the policy was executed in the world it would yield the optimal outcome. This work is traditionally evaluated using simulation.

We advocate that the absence of real-world multi-agent systems that help users perform their tasks better is a clear indication that the focus on the model-to-algorithm-to-policy approach has not produced results that can be extended to address all the issues that must be addressed in the real world. Prior work on the PCM approach [1, 2] and the work presented here provide significant evidence that the current focus is problematic. For example, it is difficult to support a real-world application with an approach that requires detailed and accurate models of the real world, when in most cases, one can only get approximate, incomplete and even incorrect models. Also, it is impossible to support a real-world application with an approach that requires excessive amounts of data to be shared among agents in order to compute a good solution. It is crucial to address all the issues simultaneously as they impose constraints on each other. In this paper, we present our experience moving our laboratory system that performed extremely well in simulation into the real world.

^{*}The work presented here is funded by the DARPA COORDINATORS Program under contract FA8750-05-C-0032. The U.S. Government is authorized to reproduce and distribute reports for Governmental purposes notwithstanding any copyright annotation thereon. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of any of the above organizations or any person connected with them. Approved for Public Release, Distribution Unlimited.

Cite as: Multi-Agent Systems for the Real World, (Extended Abstract), Rajiv Maheswaran, Craig Milo Rogers, Romeo Sanchez, Pedro Szekely, *Proc. of 8th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2009)*, Decker, Sichman, Sierra and Castelfranchi (eds.), May, 10–15, 2009, Budapest, Hungary, pp. 1281–1282
Copyright © 2009, International Foundation for Autonomous Agents and Multiagent Systems (www.ifaamas.org), All rights reserved.

2. TECHNICAL CHALLENGES

In this section, we discuss how the real world introduces challenges within the various processes discussed above.

The Modeling Challenge: Information provided about real world domains either are sparse or extremely noisy. For example, in a disaster-rescue simulation field exercise, the estimates of travel times between various locations were calculated using a “reasonable” constant speed on the GPS distance. However, traffic patterns at the times of the exercise were not predictable. In one experiment, a truck blocked off access to the only warehouse for several minutes. Thus, any model that used bounds or distributions based on these estimates would be giving their algorithms very noisy assumptions. Any approach based on planning far in the future would compound these timing errors.

The Communication-Update Challenge: One of the key technical challenges is figuring out how an observation should affect the models of *other* agents. The crux of the problem is what to *communicate* after the appropriate *update* has taken place. The problem is that the decision of how to modify models should not be made on local information. Distribution (local observation) and subjectivity (partial knowledge of team objectives) coupled with real world limitations on communication bandwidth, latency make this one of the most difficult challenges for a system designer.

The Algorithm Challenge: Given a model, an algorithm must produce a policy within the real world limitations of bounded computation and memory. These policies must be produced in timely manner. Specifically, an update or communication of an update can signify a shift in the state of the world that requires immediate action. In the field exercise, this occurs after every survey. Long delays in producing a policy can lead to wasting the time of human agents, delaying the team’s future actions and damaging the team objective, e.g., letting an injured die. A designer must choose algorithms that work within both bounded rationality limits and timeliness requirements of the domain.

3. EVALUATION

The DARPA COORDINATORS program carried out an extensive evaluation of two multi-agent coordination approaches in a field exercise. The goal of the evaluation was to measure the effectiveness of teams assisted by agents compared with a team without such assistance. The field exercise was a physical analogue of a disaster rescue scenario.

Sites represent locations that suffered damage following a disaster, and have facilities such as a gas main, power main, water main, clinic, hospital or a warehouse containing repair parts. Sites have to be surveyed in order to discover the nature of the damage to utilities and to discover the number of injured that must be saved. Points

are given for restoring utilities and rescuing the injured. Injured and utilities at different sites are worth different number of points. The team objective was to maximize the number of points obtained in two hours.

Many constraints exist for repairing utilities and for rescuing injured. For example, the power and gas at a site must be turned off before any utility or injury survey can be performed; the utility mains must be surveyed and repaired for the services to become operational at other sites; all services at the clinics must be restored in order for the clinic to become operational for healing injured; participants have different capabilities and different utility problems require different capabilities; repairs may require multiple subtasks with various temporal relationships.

In the field exercise, there were 12 sites scattered over a five square mile area. The rescue team had 8 people on the field and 2 commanders. Participants were given multi-band radios to communicate among themselves and with the commanders.

Radio Team: the radio team used multi-channel radios to coordinate among team members. They had a command center staffed with two people who had a map of the area and a white board to record status information. The other participants were in the field carrying out the activities of the exercise. The radio team received a description of the exercise the day before the exercise. This description included information about the physical layout, skill assignments, estimates for damage and injured, location of warehouses, and the points scored for the repairs and injury rescues at each site. The radio team participants determined the strategy themselves.

Coordinator Teams: these teams had radios and also carried tablet computers running multi-agent software that offered them assistance on who should do what, when and where. The coordinator team participants were subject to the same rules as the radio team. The coordinator teams also had a command center staffed by two researchers. Their task was not to command, but to monitor the status of the software and intervene when it malfunctioned due to bugs or user errors that the software was not designed to handle. The coordinator teams received the same scenario description as the radio team, and also had 24 hours to compose and initial plan and configure the software for the exercise.

Evaluation: The formal evaluation involved two scenarios. The radio team performed each scenario once while the coordinator teams performed each exercise twice under two conditions: (1) *Cell modems* where the computers that participants carried communicated using cellular modems, and thus were completely autonomous. (2) *Table top* where all the computers were placed in a conference room and networked together using a wired ethernet. Two people played the role of each participant. The *rear* person at the conference room operated the computer and the *forward* person performed the activities in the field. The rear person announced the instructions of the coordinator over the radio. The forward person performed the activities and communicated the results back to the rear person, who entered the results on the computer, and then relayed the next set of instructions.

Results The score of each exercise was the sum of the points received for restoring services at the sites and rescuing injured before the 2 hour time limit. Figure 1 shows the results for the radio team and the our team. The results for the second coordinator team were not made public.

In both Exercises 1 and 2, the cell modem condition yielded the lowest scores. The system was unable to overcome the large latencies and reduced bandwidth of the cellular network, often resulting in users waiting for 5 minutes to get advice from the system. The cell modem experiment in Exercise 2 was suspended because the

	Exercise 1	Exercise 2
Radio Team	1,800	3,195
PCM (cell modem)	1,500	—
PCM (table top)	2,180	2,935

Table 1: Evaluation results.

system failed to cope with the communication challenges of the slow, unreliable cellular network.

The most interesting result of the evaluation is that it is so difficult to outscore humans in a complex planning and scheduling problem. The PCM approach outscored the humans once. The alternative multi-agent approach did not come close to outperforming the humans even once.

4. DISCUSSION

The evaluation results provide strong evidence to show that planning and scheduling with resources and locations under uncertainty in dynamic distributed domains is extremely difficult. The search space is enormous and we don't have effective means of narrowing the search. In the distributed setting, the problem becomes even more intractable as it requires agents to communicate information about the status of their activities leading to excessive communication traffic.

In contrast, humans were able to operate effectively communicating very little information. The radio team used a similar strategy to ours: multiple teams, each which had an itinerary. The radio team was far more capable of adapting their strategy during the exercise, allowing agents to jump around in the itinerary as necessary, where we followed itineraries strictly.

We developed sound high-level strategies, i.e., itineraries, for both Exercise 1 and Exercise 2. In Exercise 1, the system effectively worked out the tactical aspects of the strategy and on the table top led the users to a better score. In Exercise 2, we lost because the system made a tactical blunder. In that scenario, only two people had gas skills, and the system assigned them to transport injured to the clinic when all the other users had finished other activities and moved on to the next site.

This exercise provides good evidence of the importance of human-in-the-loop systems. The main reason that we were not significantly outperformed by the radio team is that we were able to input a human-developed strategy, tailored to the specific problem at hand, into the system. The system took care of the tactical aspects of the plan. The system determined the timing, chose who within a sub-team should do what and who should move to the next site in the itinerary. The other coordinator team had no ability for users to enter a strategy, and the system tried to solve the problem with no human assistance.

5. REFERENCES

- [1] R. Maheswaran and P. Szekely. Criticality metrics for distributed plan and schedule management. In *Proceedings of the 18th International Conference on Automated Planning and Scheduling*, Sydney, Australia, September 2008.
- [2] R. T. Maheswaran, P. Szekely, M. Becker, S. Fitzpatrick, G. Gati, J. Jin, R. Neches, N. Noori, C. Rogers, R. Sanchez, K. Smyth, and C. VanBuskirk. Predictability & criticality metrics for coordination in complex environments. In *Proceedings of the 7th International Joint Conference on Autonomous Agents and Multiagent Systems*, Estoril, Portugal, May 2008.