Electric Elves: What Went Wrong and Why

Milind Tambe, Emma Bowring,
Jonathan P. Pearce,
Pradeep Varakantham
Computer Science Dept.
University of Southern California
Los Angeles, CA 90089
{tambe,bowring,jppearce,varakant}@usc.edu

Paul Scerri
Robotics Institute
Carnegie Mellon University
Pittsburgh, PA 15213
pscerri@cs.cmu.edu

David V. Pynadath
Information Sciences Institute
University of Southern California
Marina del Rey, CA 90292
pynadath@isi.edu

Abstract

Software personal assistants continue to be a topic of significant research interest. This paper outlines some of the important lessons learned from a successfully-deployed team of personal assistant agents (Electric Elves) in an office environment. These lessons have important implications for similar on-going research projects.

The Electric Elves project was a team of almost a dozen personal assistant agents which were continually active for seven months. Each elf (agent) represented one person and assisted in daily activities in an actual office environment. This project led to several important observations about privacy, adjustable autonomy, and social norms in office environments. This paper outlines some of the key lessons learned and, more importantly, outlines our continued research to address some of the concerns raised.

Introduction

The topic of software personal assistants, particularly for office environments, is of continued and growing research interest (Scerri *et al.* 2002; Maheswaran *et al.* 2004; Modi and Veloso 2005; CALO 2003; Pynadath and Tambe 2003). The goal is to provide software agent assistants for individuals in an office as well as software agents that represent shared office resources. The resulting set of agents coordinate as a team to facilitate routine office activities.

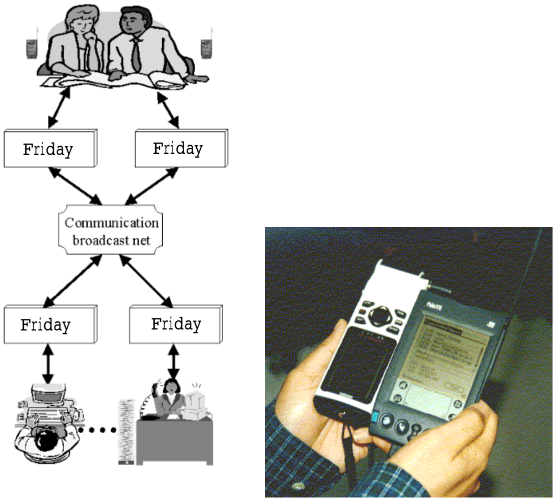
This paper outlines some key lessons learned during the successful deployment of a team of a dozen agents, called Electric Elves (E-Elves), which ran continually from June 2000 to December 2000 at the Information Sciences Institute (ISI) at the University of Southern California (USC)(Scerri *et al.* 2002; Chalupsky *et al.* 2001; Pynadath and Tambe 2003; 2001; Pynadath *et al.* 2000). Each elf (agent) acted as an assistant to one person and aided in the daily activities of an actual office environment. Originally, the E-Elves project was designed to focus on team coordination among software agents. However, while team coordination remained an interesting challenge, several other unanticipated research issues came to the fore. Among these new issues were adjustable autonomy, i.e. agents dynamically adjusting their own level of autonomy, privacy and social norms in office environments.

This paper outlines both the lessons learned during the E-Elves project and our continued research to address the issues raised. Several publications outline the primary technical contributions of E-Elves and research inspired by E-Elves in detail. However, the goal of this paper is to highlight some of what went wrong in the E-Elves project and provide a broad overview of technical advances in the areas of concern without providing specific technical details.

Description of Electric Elves

The Electric Elves (E-Elves) project deployed an agent organization at USC/ISI to support daily activities in a human organization(Pynadath and Tambe 2003; Chalupsky *et al.* 2001). Dozens of routine tasks are required to ensure coherence in a human organization's activities, e.g., monitoring the status of activities, gathering information relevant to the organization and keeping everyone in the organization informed. Teams of software agents can aid humans in accomplishing these tasks, facilitating the organization's coherent functioning, while reducing the burden on humans.

The overall design of the E-Elves is shown in Figure 1(a). Each proxy is called Friday (after Robinson Crusoe's manservant Friday) and acts on behalf of its user in the agent team. The basic design of the Friday proxies is discussed in detail in (Pynadath and Tambe 2003; Tambe *et al.* 2000) (where they are referred to as TEAMCORE proxies). Friday can perform a variety of tasks for its user. If a user is delayed to a meeting, Friday can reschedule the meeting, informing other Fridays, who in turn inform their users. If there is a research presentation slot open, Friday may respond to the invitation to present on behalf of its user. Friday can also order its user's meals (see Figure 2(a)) and facilitate informal meetings by posting the user's location on a Web page. Friday communicates with users via user workstations and using wireless devices, such as personal digital assistants (PALM VIIs) and WAP-enabled mobile phones. Figure 1(b) shows a PALM VII connected to a Global Positioning Service (GPS) device, for tracking users' locations and enabling wireless communication between Friday and a user. Each Friday's team behavior is based on a teamwork model called STEAM(Tambe 1997). STEAM encodes and enforces the constraints among roles that are required for the success of the joint activity, e.g., meeting attendees should arrive at a meeting simultaneously. When an important role



(a) (b)

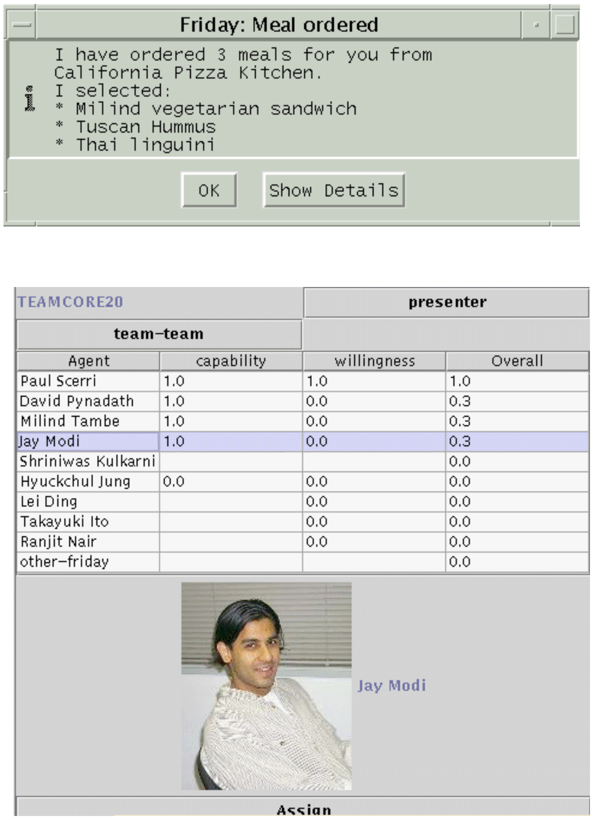
Figure 1: (a) Overall E-Elves architecture, showing Friday agents interacting with users. (b) Palm VII for communicating with users and GPS device for detecting their location.

within the team (e.g. role of a presenter for a research meeting) opens up, the team needs to find the best person to fill that role. To achieve this, the team auctions off the role, taking into consideration complex combinations of factors and assigning the best-suited agent or user. Friday can bid on behalf of its user, indicating whether its user is capable and/or willing to fill a particular role. Figure 2(b) shows a tool that allows users to view auctions in progress and intervene if they so desire. In the auction shown, Jay Modi's Friday has bid that Jay is capable of giving the presentation, but is unwilling to do so. Paul Scerri's agent has the highest bid and was eventually allocated the role.

Adjustable Autonomy

Adjustable autonomy (AA) is clearly important to the E-Elves because, despite the range of sensing devices, Friday has considerable uncertainty about the user's intentions and even location, hence Friday will not always be capable of making good decisions. On the other hand, while the user can make good decisions, Friday cannot continually ask the user for input, because it wastes the user's valuable time.

We illustrate the AA problem by focusing on the key example of meeting rescheduling in E-Elves: A central task for the E-Elves is ensuring the simultaneous arrival of attendees at a meeting. If any attendee arrives late, or not at all, the time of all the attendees is wasted. On the other hand, delaying a meeting is disruptive to users' schedules. Friday acts as proxy for its user so its responsibility is to ensure that its user arrives at the meeting at the same time as other users. Clearly, the user will often be better able to determine whether he/she needs the meeting to be delayed. However, if the agent transfers control to the user for the decision, it must guard against miscoordination while waiting for the user's



```
I have ordered 3 meals for you from
California Pizza Kitchen.
I selected:
* Milind vegetarian sandwich
* Tuscan Hummus
* Thai linguini
```

Agent	capability	willingness	Overall
Paul Scerri	1.0	1.0	1.0
David Pynadath	1.0	0.0	0.3
Milind Tambe	1.0	0.0	0.3
Jay Modi	1.0	0.0	0.3
Shriniwas Kulkarni			0.0
Hyuckchul Jung	0.0	0.0	0.0
Lei Ding		0.0	0.0
Takayuki Ito		0.0	0.0
Ranjit Nair		0.0	0.0
other-friday			0.0

Figure 2: (a) TOP: Friday asking the user for input regarding ordering a meal. (b)BOTTOM: Electric Elves auction tool.

response, especially if the response is not forthcoming, e.g., if the user is in another meeting. Some decisions are potentially costly, e.g., rescheduling a meeting to the following day, so an agent should avoid taking them autonomously. To buy more time for the user to make a decision, an agent has the option of delaying the meeting, i.e., changing coordination constraints. Overall the agent has three options: make an autonomous decision; transfer control; or change coordination constraints. The autonomy reasoning must select from these actions while balancing the various competing influences.

Lessons from Electric Elves

Our first attempt to address AA in Electric Elves was to resolve the transfer-of-control decision by learning from user input; in particular by using decision-tree learning based on C4.5. In training mode, Friday recorded values of a dozen carefully selected attributes and the user's preferred action (identified by asking the user) whenever it had to make a decision. Friday used the data to learn a decision tree that encoded various rules. For example, it learned a rule: IF *two person meeting with important person* AND *user not at department at meeting time* THEN *delay the meeting 15 minutes*. During training Friday also asked if the user wanted such decisions taken autonomously in the future. From these responses, Friday used C4.5 to learn a second decision tree which encoded its AA reasoning.

Initial tests with the C4.5 approach were promising (Pynadath and Tambe 2003), but a key problem soon became apparent. When Friday encountered a decision for which it had learned to transfer control to the user, it would wait indefinitely for the user to make the decision, even though this inaction could lead to miscoordination with teammates if the user did not respond or attend the meeting. To address this problem a fixed time limit (five minutes) was added and if the user did not respond within the time limit, Friday took an autonomous action. Although performance improved, when the resulting system was deployed 24/7, it led to some dramatic failures, including:

- Example 1: Tambe's (a user) Friday autonomously cancelled a meeting with the division director because Friday over-generalized from training examples.
- Example 2: Pynadath's (another user) Friday incorrectly cancelled the group's weekly research meeting when a time-out forced the choice of an autonomous action when Pynadath did not respond.
- Example 3: A Friday delayed a meeting almost 50 times, each time by 5 minutes. It was correctly applying a learned rule but ignoring the nuisance to the rest of the meeting participants.
- Example 4: Tambe's Friday automatically volunteered him for a presentation, but he was actually unwilling. Again Friday had over-generalized from a few examples and when a timeout occurred had taken an undesirable autonomous action.

From the growing list of failures, it became clear that the C4.5 approach faced some significant problems. Indeed, AA in a team context requires more careful reasoning about the costs and benefits of acting autonomously and transferring control and needs to better deal with contingencies. In particular, an agent needs to: avoid taking risky decisions (like example 1) by taking a lower risk delaying action to buy the user more time to respond; deal with failures of the user to quickly respond (examples 2 and 4); and plan ahead to avoid taking costly sequences of actions that could be replaced by a single less costly action (example 3). In theory, using C4.5, Friday might have eventually been able to learn rules that would successfully balance costs, deal with uncertainty and handle all the special cases but a very large amount of training data would be required, even for this relatively simple decision. Given our experience, it was decided that a more careful approach, that explicitly reasoned about important factors was required for AA reasoning in a multi-agent context.

On-going research

To address the early failures in AA, we wanted a mechanism that met three important requirements. First, it should allow us to explicitly represent and reason about different types of costs as well as uncertainty, e.g., costs of miscoordination vs. costs of taking an erroneous action. Second, it should allow lookahead to plan a systematic transfer of decision-making control and provide a response that is better in the longer term (for situations such as a non-responsive user). Finally, it should allow us to encode significant quantities of initial domain knowledge, particularly costs and uncertainty, so that the agent does not have to learn everything from scratch (as was required with C4.5).

Markov Decision Processes (MDPs) fit the above requirements and so, in a second incarnation of E-Elves, were invoked for each decision that Friday made: rescheduling meetings, delaying meetings, volunteering a user for presentation or ordering meals. Although MDPs were able to support sequential decision making in the presence of transitional uncertainty (uncertainty in the outcomes of actions), they were hampered by not being able to handle observational uncertainty (uncertainty in sensing). Specifically, Friday's "sensing" was very coarse and while Friday might follow an appropriate course of action when its observations were correct, when they were incorrect its actions were very poor. For example, a user being in their office was "sensed" by checking for keyboard activity but if they were reading papers Friday would assume they were out and act accordingly – often autonomously.

In a project inspired by E-Elves, we took the natural next step to address this issue by using partially observable MDPs (or POMDPs) to model observational uncertainty and find appropriate courses of action with respect to this observational uncertainty. However, existing techniques for solving POMDPs either provide loose quality guarantees on solutions (approximate algorithms) or are computationally very expensive (exact algorithms). Our recent research has developed efficient exact algorithms for POMDPs, deployed in service of adjustable autonomy, by exploiting the notions of progress or physical limitations in the environment. The key insight was that given an initial (possibly uncertain) set

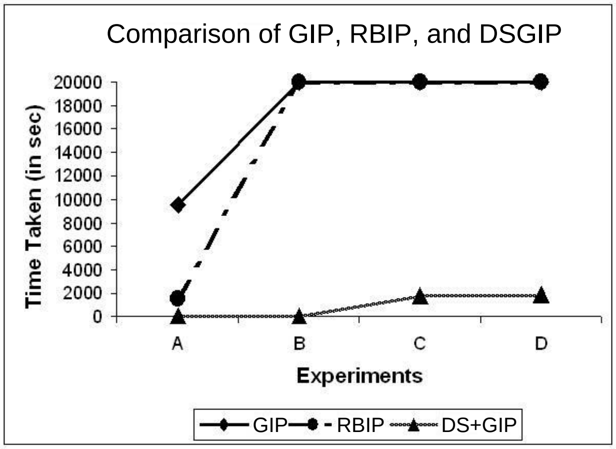


Figure 3: Enhancements provide orders of magnitude speedup over RBIP and GIP

of starting states, the agent needs to be prepared to act only in a limited range of belief states; most other belief states are simply unreachable given the dynamics of the monitored process so no action needs to be generated for such belief states. These bounds on the belief probabilities are obtained using Lagrangian techniques in polynomial time (Varakantham *et al.* 2005).

We tested this enhanced algorithm against two of the fastest exact algorithms: GIP (Generalized Incremental Pruning) and RBIP (Region Based Incremental Pruning). Our enhancements in fact provide orders of magnitude speedup over RBIP and GIP in problems taken from the meeting re-scheduling of Electric Elves, as illustrated in Figure 3. In the Figure, the x-axis shows four separate problem instances, and y-axis shows the run-time in seconds. (Since the problem runs were cutoff at 20000 seconds, the lines for GIP and RBIP are seen to flatten out at 20000 seconds.) DS-GIP is our enhanced algorithm and it is seen to be at least an order of magnitude faster than the other algorithms.

Another issue that arose during the MDP implementation of E-Elves was that both MDPs and POMDPs rely on knowing the probability of events occurring in the environment. For example, the MDP for meeting rescheduling needed to know the probability that a message posted to a Palm Pilot while the user was away from the office would be answered within five minutes. Clearly, these probabilities varied from user to user and hence it was natural to apply learning to adjust these parameters. While the learning itself was effective, the fact that Friday did not necessarily behave the same way each day could be disconcerting to the users – even if the new behavior might actually be "better". The problem was that Friday would change its behavior without warning, after users had adjusted to its (imperfect) behavior. Later research (Pynadath and Tambe 2001) addressed this by allowing users to add hand-constructed inviolable constraints.

Privacy

Just as with adjustable autonomy, privacy was another area of research which was not initially considered important in Electric Elves. Unfortunately, while several privacy related problems became apparent, no systematic solutions were developed during the course of the project. We will describe some of the problematic instances of privacy loss and then some recent steps to quantitatively measure privacy loss that have been inspired by the E-Elves insights.

Lessons from Electric Elves

We begin with a few arenas where privacy issues were immediately brought to the forefront. First, a key part of E-Elves was to assist users in locating other users to facilitate collaborative activities, e.g. knowing that a user is in his/her office would help determine if it is worth walking down to that user's office to engage in discussions. This was especially relevant in our domain since the Information Sciences Institute and main USC campus are across town from each other. Unfortunately, making a user's GPS location available to other project members at all times, even if GPS capabilities were switched off at home, was a very significant invasion of privacy. This led to a too transparent tracking of people's locations. For instance, it was possible to see that a user was delayed for a meeting not because he was stuck in traffic as he suggested but rather because he was eating breakfast at a small cafe.

Second, even when such obviously intrusive location monitoring was switched off and the E-Elves only indicated whether or not a user was in his/her office, privacy loss still occurred. For instance, one user wished to work uninterrupted to finish up a proposal. To simulate being away, he switched off the lights, locked the door and did not respond to knocks or phone calls. To his surprise, a colleague sent him an email, saying that he knew he was in the office because his elf was still transmitting the fact that he was in his office to others.

Third, E-Elves monitored users' patterns of daily activities. This included statistics on users actions related to various meetings, i.e. whether a user was delayed to a meeting, whether he/she attended a meeting and whether the user cancelled the meetings. These detailed statistics were another source of privacy loss when they were made available to other users – in this case, to a student who was interested in running machine learning on the data. The student noticed and pointed out to a senior researcher that, when his meetings were with students, he was always late by 5 minutes, while, on the other hand, he was punctual for his meetings with other senior researchers.

Fourth, one of the parameters used in determining meeting importance was the importance attached to each of the people in the meeting. An agent used this information to determine the actions to take with respect to a meeting, e.g. canceling a meeting with someone very important in the organization was to be avoided. Unfortunately, such information about user importance was clearly very private and caused a minor controversy when it was accidentally leaked.

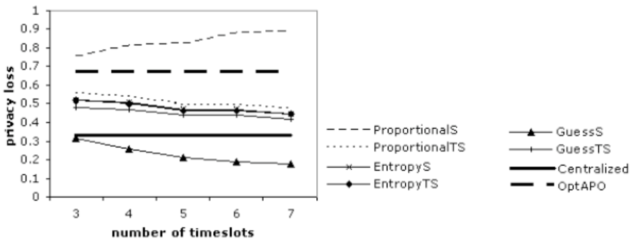


Figure 4: Privacy loss for the SynchBB algorithm using six different VPS metrics

On-going research

Our subsequent research on privacy has focused primarily on the last issue, that of private information being leaked during negotiations between team members. These negotiations often took the form of distributed constraint optimization problems (DCOP)(Modi *et al.* 2005; Mailler and Lesser 2004), in which cooperative agents exchanged messages in order to optimize a global objective function to which each agent contributes. For example, agents may try to optimize a global schedule of meetings by setting their individual schedules. The objective function would incur penalties if attendees of a meeting scheduled it at different times, or if an agent had more than one meeting scheduled at the same time.

Many algorithms exist for solving such problems. However, it was not clear which algorithms preserved more privacy than others, or more fundamentally, what metrics should be used for measuring the privacy loss of each algorithm. While researchers had begun to propose metrics for analysis of privacy loss in multiagent algorithms for distributed optimization problems, a general quantitative framework to compare these existing metrics for privacy loss or to identify dimensions along which to construct new metrics was lacking. To address this question, we introduced VPS (Valuations of Possible States)(Maheswaran *et al.* 2005), a general quantitative framework to express, analyze and compare existing metrics of privacy loss. Based on a state-space model, VPS was shown to capture various existing measures of privacy created for specific domains of distributed constraint satisfaction and optimization problems. Using VPS, we were able to analyze the privacy loss of several algorithms in a simulated meeting scheduling domain according to many different metrics.

Figure 4 from (Maheswaran *et al.* 2005) shows an analysis of privacy loss for the SynchBB algorithm across six different VPS metrics (ProportionalS, ProportionalTS, GuessS, GuessTS, EntropyS and EntropyTS) for a particular meeting scheduling scenario of three agents, averaged over 25 experimental runs in which agents' personal timeslot preference were randomly generated. Also shown on the graph is the privacy loss for the OptAPO algorithm (Mailler and Lesser 2004) and for a centralized solver; both of these were shown to have the same privacy loss regardless of the VPS metric used. The x-axis shows the number of timeslots when meetings could be scheduled in the overall problem, and the y-axis shows the systemwide privacy loss, expressed as the mean of the privacy losses of each agent in the system, where 0 means an agent has lost no privacy to any other agent and 1 means an agent has lost all privacy to all other agents. The graph shows that, according to four of the six metrics, SynchBB's privacy loss lies in between that of centralized and OptAPO, and, interestingly, the effect of increasing the number of timeslots in the system causes privacy loss to increase according to one metric, but decrease according to another.

The key result illustrated in Figure 4 is that distribution in DCOPs does not automatically guarantee improved privacy when compared to a centralized approach, at least as seen from the algorithms tested here — an important result given that privacy is a key motivation for deploying DCOP algorithms in software personal assistants. Thus, DCOP algorithms must more carefully address privacy concerns.

Social norms

Another area which provided unexpected research issues was social norms. Day-to-day operation with E-Elves exposed several important research issues that we have not yet specifically pursued.

Lessons from Electric Elves

Agents in office environments must follow the social norms of the human society within which the agents function. For example, agents may need to politely lie on behalf of their users in order to protect their privacy. If the user is available but does not wish to meet with a colleague, the agent should not transmit the user's location and thus indirectly indicate that the user is unwilling to meet with his/her colleague. Even more crucially, the agent should not indicate to the colleague that meeting with that colleague is considered unimportant. Rather, indicating that the user is unavailable for other reasons is preferable.

Another interesting phenomenon was that users would manipulate the E-Elves to allow themselves to violate social norms without risking being seen to violate norms. The most illustrative example of this was the auction for presenter at regular group meetings. This was a role that users typically did not *want* to perform, because it required preparing a presentation, but also did not want to appear to *refuse*. Several users manipulated the E-Elves role allocation auction to allow themselves to meet both of these conflicting goals. One method was to let Friday respond to the auction autonomously, knowing that the controlling MDP was conservative and assigned a very high cost to incorrectly accepting the role on the user's behalf. A more subtle technique was to fill up one's calander with many meetings because Friday would take into account how busy the person was. Unfortunately, Friday was not sophisticated enough to distinguish between "Project Meeting" and "Lunch" or "Basketball". In both of these cases, the refusal would be attributed to the agent, rather than directly to the user. Another source of manipulation came in when a user had recently presented, since the auction would not assign them the role again immediately. Thus shortly after presenting users could manually submit affirmative bids safe in the knowledge their bid would not be accepted while still getting credit from the rest of the team for their enthusiasm. The important lesson here is that not only must personal assistants not violate norms but they should also minimize opportunities for individuals to hide behind the technology to violate norms.

Summary

This paper outlines some of the important lessons learned from a successfully-deployed team of personal assistant agents (Electric Elves) in an office environment. This project led to several important observations about privacy, adjustable autonomy, and social norms for agents deployed in office environments. This paper outlines some of the key lessons learned and, more importantly, outlines our continued research to address some of the concerns raised. These lessons have important implications for similar on-going research projects.

Acknowledgements

This material is based upon work supported by DARPA, through the Department of the Interior, NBC, Acquisition Services Division, under Contract No. NBCHD030010.

References

CALO: Cognitive Agent that Learns and Organizes, 2003. http://www.ai.sri.com/project/CALO, http://calo.sri.com.

H. Chalupsky, Y. Gil, C. Knoblock, K. Lerman, J. Oh, D. Pynadath, T. Russ, and M. Tambe. Electric elves: Applying agent technology to support human organizations. In *International Conference on Innovative Applications of Artificial Intelligence*, 2001.

R. T. Maheswaran, M. Tambe, E. Bowring, J. P. Pearce, and P. Varakantham. Taking DCOP to the real world: efficient complete solutions for distributed multi-event scheduling. In *AAMAS*, 2004.

R. T. Maheswaran, J. P. Pearce, P. Varakantham, E. Bowring, and M. Tambe. Valuations of possible states (vps): A unifying quantitative framework for analysis of privacy loss in collaboration. In *Proceedings of the Fourth International Joint Conference on Autonomous Agents and Multi Agent Systems (AAMAS 2004)*, New York, NY, July 2005.

R. Mailler and V. Lesser. Solving distributed constraint optimization problems using cooperative mediation. In *AAMAS*, 2004.

P. J. Modi and M. Veloso. Bumping strategies for the multiagent agreement problem. In *AAMAS*, 2005.

P. J. Modi, W. Shen, M. Tambe, and M. Yokoo. Adopt: Asynchronous distributed constraint optimization with quality guarantees. *Artificial Intelligence*, 161(1-2):149–180, 2005.

David V. Pynadath and Milind Tambe. Revisiting asimov's first law: A response to the call to arms. In *Proceedings of Workshop on Agent Theories, Architectures and Languages (ATAL-01)*, 2001.

David V. Pynadath and Milind Tambe. Automated teamwork among heterogeneous software agents and humans. *Journal of Autonomous Agents and Multi-Agent Systems (JAAMAS)*, 7:71–100, 2003.

David V. Pynadath, Milind Tambe, Hans Chalupsky, Yigal Arens, et al. Electric elves: Immersing an agent organization in a human organization. In *Proceedings of the AAAI Fall Symposium on Socially Intelligent Agents*, 2000.

P. Scerri, D. Pynadath, and M. Tambe. Towards adjustable autonomy for the real-world. *Journal of Artificial Intelligence Research*, 17:171–228, 2002.

M. Tambe, D. Pynadath, and N. Chauvat. Building dynamic agent organizations in cyberspace. *IEEE Internet Computing*, 4(2):65–73, 2000.

M. Tambe. Towards flexible teamwork. *Journal of Artificial Intelligence Research*, 7:83–124, 1997.

P. Varakantham, R. Maheswaran, and M. Tambe. Exploiting belief bounds: Practical POMDPs for personal assistant agents. In *AAMAS*, 2005.