AI Magazine Volume 29 Number 2 (2008) (© AAAI)

Electric Elves: What Went Wrong and Why

Milind Tambe, Emma Bowring,
Jonathan P. Pearce, Pradeep Varakantham,
Paul Scerri, and David V. Pynadath

■ *Software personal assistants continue to be a topic of significant research interest. This article outlines some of the important lessons learned from a successfully deployed team of personal assistant agents (Electric Elves) in an office environment. In the Electric Elves project, a team of almost a dozen personal assistant agents 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. In addition to outlining some of the key lessons learned we outline our continued research to address some of the concerns raised.*

The topic of software personal assistants, particularly for office environments, is of continued and growing research interest (Scerri, Pynadath, and Tambe 2002; Maheswaran et al. 2004; Modi and Veloso 2005; Pynadath and Tambe 2003).[1] 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 article 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, Pynadath, and Tambe 2002; Chalupsky et al. 2002; 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 (agents dynamically adjusting their own level of autonomy), as well as privacy and social norms in office environments. Several earlier publications outline the primary technical contributions of E-Elves and research inspired by E-Elves in detail. However, the goal of this article 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 project deployed an agent organization at USC/ISI to support daily activities in a human organization (Pynadath and Tambe 2003, Chalupsky et al. 2002). Dozens of routine tasks are required to ensure coherence in a human organization's activities, for example, 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. 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) and Tambe, Pynadath, and Chauvat (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 invi-

ISSN 0738-4602

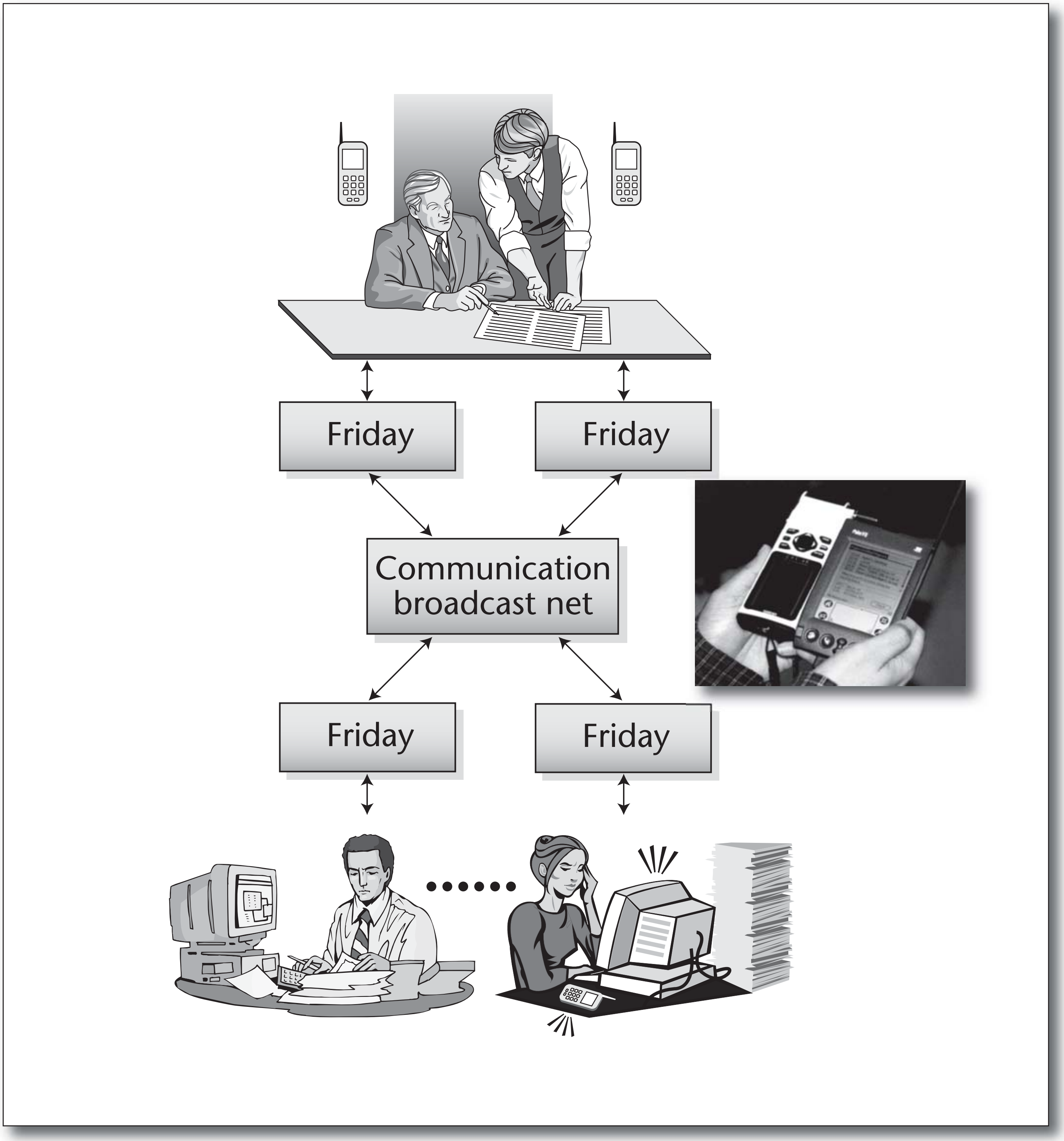

Figure 1. Overall E-Elves Architecture, Showing Friday Agents Interacting with Users.

tation to present on behalf of its user. Friday can also order its user's meals (see figure 2) and facilitate informal meetings by posting the user's location on a web page. Friday communicates with users through user workstations and use of wireless devices such as personal digital assistants (Palm VIIs) and WAP-enabled mobile phones, which, when connected to a global positioning service (GPS) device, track users' locations and enable wireless communication between Friday and a user.

Each Friday's team behavior is based on a team-

Friday: Meal ordered

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

Figure 2: Friday Asking the User for Input.

work model called STEAM (Tambe 1997). STEAM encodes and enforces the constraints among roles that are required for the success of the joint activity. For example, meeting attendees should arrive at a meeting simultaneously. When an important role within the team (such as the 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.

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 response, especially if the response is not forthcoming, such as if the user is in another meeting. Some decisions are potentially costly (for example, 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 (that is, 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 E-Elves was to learn 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 (the one it had learned to be the user's preferred action). This led to improved performance, and the problem uncovered in initial tests appeared to have been addressed. Unfortunately, when the E-Elves were first deployed 24/7, there were some dramatic failures. In example 1, Friday autonomously cancelled a meeting with the division director because Friday overgeneralized from training examples. In example 2, Friday incorrectly cancelled the group's weekly research meeting when a timeout forced the choice of an autonomous action when Pynadath did not respond. In example 3, 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. Finally, in example 4, Friday automatically volunteered Tambe for a presentation, but he was actually unwilling to participate. Again Friday had overgeneralized 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, we decided to pursue an alternative approach that explicitly considered costs and uncertainties.

Ongoing 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, such as costs of miscoordination versus 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 nonresponsive 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 these requirements and so, in a second incarnation of E-Elves, MDPs were invoked for each decision that Friday made: rescheduling meetings, delaying meetings, volunteering a user for presentation, or ordering meals (Scerri, Pynadath, and Tambe 2002). 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. 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 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. 2007; Varakantham, Maheswaran, and Tambe 2005).

We tested this enhanced algorithm against two of the fastest exact algorithms: generalized incremental pruning (GIP) and region-based incremental pruning (RBIP). Our enhancements in fact provide orders of magnitude speedup over RBIP and GIP in problems taken from the meeting rescheduling of Electric Elves, as illustrated in figure 3. In the figure, the x-axis shows four separate problem instances, and the y-axis shows the run time in seconds (the problem runs were cut off at 20,000 seconds). DSGIP 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 is possibly of special relevance to personalization in software assistants, above and beyond E-Elves. In particular, both MDPs and POMDPs rely on knowing

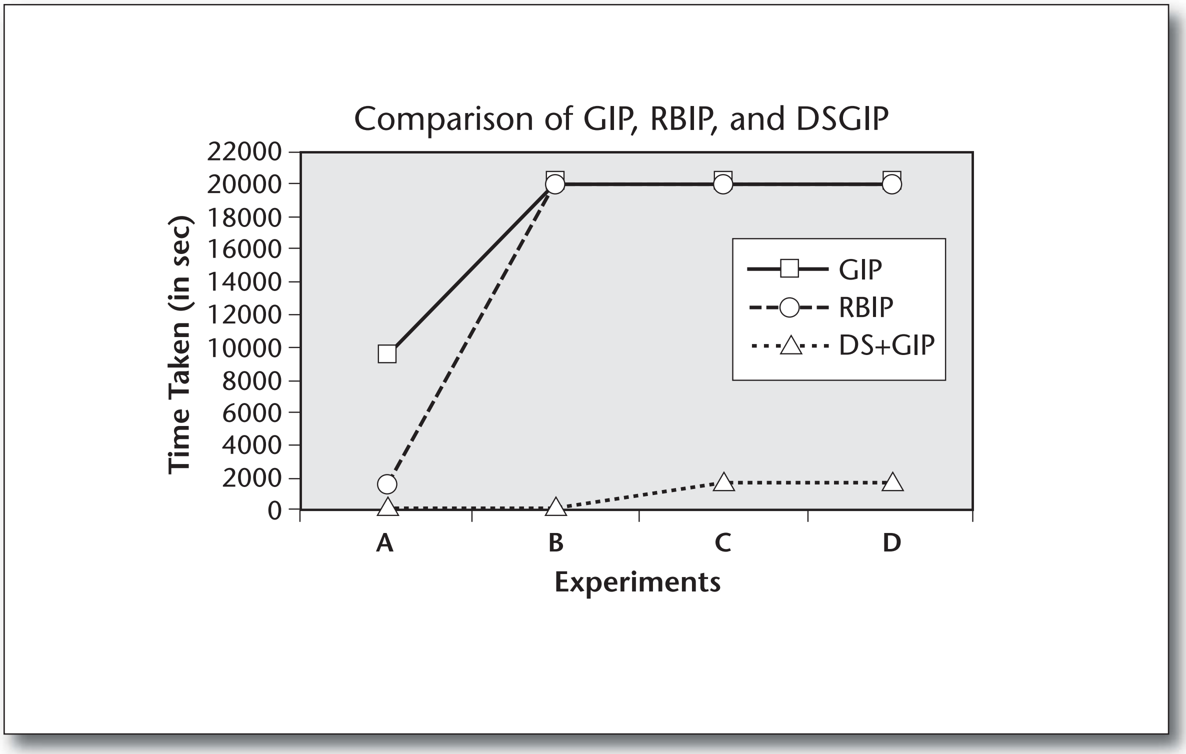

Figure 3: Speedup over RBIP and GIP.

the probability of events occurring in the environment. 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 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 work (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 that 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, such as knowing that a user is in his or her office, which would help determine if it is worth walking down to that person's office to engage in discussions. This was especially relevant in our domain because ISI and the 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 outside work hours, was a very significant invasion of privacy. This led to a too transparent tracking of people's locations. For instance, when user Tambe had an early morning 8:30 a.m. meeting, and he was delayed, he blamed it on the Los Angeles freeway traffic. However, one of the meeting attendees had access to Tambe's GPS data. So he could remark that the delay in this meeting was not because of traffic as was suggested but rather because Tambe was eating breakfast at a small cafe next to ISI.

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 or her office, privacy loss still occurred. For instance, one standard technique for Tambe to avoid getting interrupted was to hide in the office and simulate being away. While finishing a particularly important proposal, he simulated being

Moving Walls

Marcel Schoppers

In the early 1980s, I used Stan Rosenschein's functional language REX to servo Flakey's two wheels to set points of speed and direction that were functions of sonar readings. This let Flakey drive along hallways with no dead reckoning or planning whatsoever. It seemed miraculous at the time; a situated automaton that knew things without needing any models. However, I thought of it as (sensor-driven) feedback control, versus (plan driven, eyes shut) feed-forward control.

I then used Mike Georgeff's procedural reasoning system (PRS) to make Flakey not only drive but navigate an office building. In some respects this project succeeded: the robot's "domain knowledge" was nothing more than a static connection graph—no distances to drive, no widths of halls or doorways, no a priori obstacles—such information was acquired en route from sensory input. In other respects, however, progress was unsatisfying. The robot would frequently get stuck facing a wall, interpret it as an obstacle, ask it to move, and then wait forever. The robot was just as helpless as if it had made a dead-reckoning error.

The fault was that the robot expected a doorway where none existed and perceived the wall as a (presumably) movable obstacle on its path. I soon saw that my program worked well only where it employed sensory feedback and that the PRS procedures were another form of feed-forward control, with built-in (though branching) expectations overruling sensing.

This realization led to the conception of *universal plans:* feedback control through Boolean state-spaces, viewing plans as control laws, and planners as reaction-choosers based on a weak model of action effects. From there, what actually happens is up to Nature, and yet "reaction plans" reliably achieve their goals because of their robustness and persistence.

Marcel Schoppers is a senior flight software engineer working at the NASA Jet Propulsion Laboratory, Section 316, in Pasadena, California.

away by switching off his office lights, locking the door and not responding to knocks on the door or telephone calls from colleagues. However, to his surprise, a colleague sent him an e-mail, saying that the colleague knew Tambe was in the office because his elf was still transmitting that fact to others.

Third, E-Elves monitored users' patterns of daily activities, including statistics on users' actions related to various meetings, such as whether a user was delayed to a meeting, whether he or 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; for example, canceling a meeting with someone very important in the organization was to be avoided. Such information about user importance was supposed to be private but was not considered particularly controversial. Unfortunately, when information about meeting importance from someone else's elf was accidentally leaked to Ph.D. students, a minor controversy ensued. Were Ph.D. students less important than other researchers at ISI? Part of the complicated explanation provided was not that meetings with Ph.D. students were any less important, but rather the fact that meeting importance rolled multiple factors into one factor: not only the importance of the meeting but also the difficulty of scheduling a meeting and the urgency of the meeting.

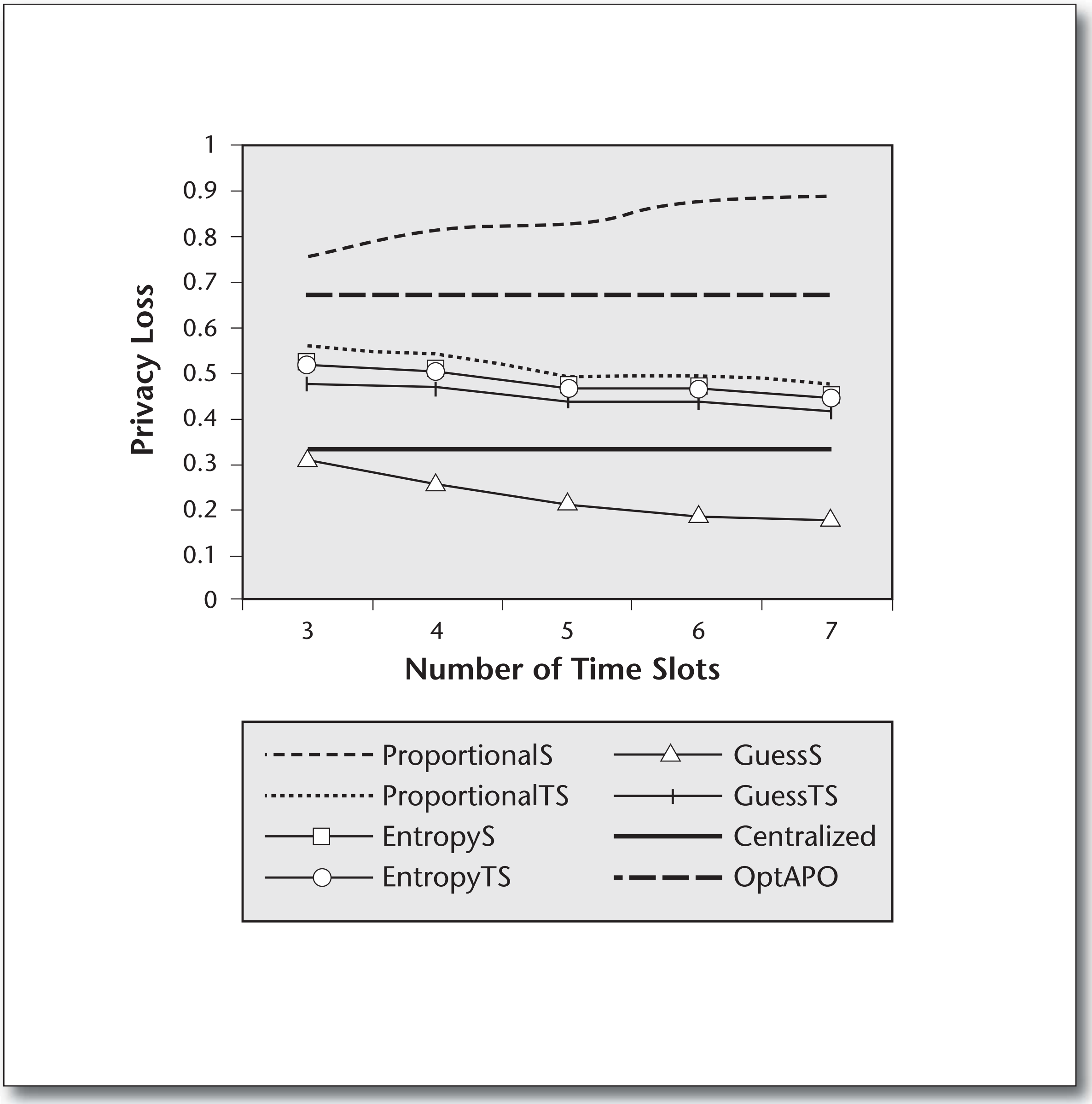

Figure 4. Privacy Loss for the SynchBB Algorithm Using Six Different VPS Metrics.

Ongoing 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.

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 valuations of possible states (VPS) (Maheswaran et al. 2006), a general quantitative framework to express, analyze, and compare existing metrics of privacy loss. With VPS, we quantify an agent's privacy loss to others in a multiagent system using as a basis the possible states the agent can be in. In particular, this quantification is based on an agent's valuation on the other agents' estimates about (a probability distribution over) its own possible states. For example, the agent may value poorly the fact that other agents are almost certain about the agent's possible state.

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. (2006) 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 time-slot preferences 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 time slots 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 time slots 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. Later work (Greenstadt, Pearce, and Tambe 2006) showed that several other DCOP algorithms (for example, Adopt [Modi et al. 2005]) did perform better than the centralized approach with respect to privacy. However, it is clear that algorithms for DCOP must address privacy concerns carefully and cannot assume that distribution alone provides privacy.

Social Norms

Another area that 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. In particular, 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 colleagues. Even more crucially, the agent should not indicate to the colleague that meeting with that colleague is considered unimportant—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 that was actually used 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 that was used 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 a "project meeting" and "lunch" or "basketball." In all 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 the user the role again immediately. Thus shortly after presenting, users could manually submit affirmative bids safe in the knowledge that their bid would not be accepted while they would still get credit from the rest of the team for their enthusiasm. (These techniques came to light only after the project ended!)

One important lesson here is that personal assistants must not violate norms. However, another is that personal assistants in this case were designed for group efficiency and, as such, face an incentive compatibility problem: users may have their own personal aims as a result of which they work around the personal assistants. Rather than insisting on group efficiency, it may be more useful to allow users to manage this incompatibility in a more socially acceptable manner.

Summary

This article outlined some of the important lessons learned from a successfully-deployed team of personal assistant agents (Electric Elves) in an office environment. The Electric Elves project led to several important observations about privacy, adjustable autonomy, and social norms for agents deployed in office environments. This article 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 ongoing research projects.

Note

1. Another example is CALO: Cognitive Agent that Learns and Organizes (calo.sri.com).

Acknowledgments

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

References

Chalupsky, H.; Gil, Y.; Knoblock, C.; Lerman, K.; Oh, J.; Pynadath, D.; Russ, T.; and Tambe, M. 2002. Electric Elves: Agent Technology for Support Human Organizations. *AI Magazine* 23(2): 11–24.

Greenstadt, R.; Pearce, J. P.; and Tambe, M. 2006. Analysis of Privacy Loss in DCOP Algorithms. In *Proceedings of the Twenty-First National Conference on Artificial Intelligence (AAAI)*, 647–653. Menlo Park, CA: AAAI Press.

Maheswaran, R. T.; Pearce, J. P.; Bowring, E.; Varakantham, P.; and Tambe, M. 2006. Privacy Loss in Distributed Constraint Reasoning: A Quantitative Framework for Analysis and Its Applications. *Journal of Autonomous Agents and Multi-Agent Systems* 13(1): 27–60.

Maheswaran, R. T.; Tambe, M.; Bowring, E.; Pearce, J. P.; and Varakantham, P. 2004. Taking DCOP to the Real World: Efficient Complete Solutions for Distributed Multi-Event Scheduling. In *Proceedings of the International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS)*. Los Alamitos, CA: IEEE Computer Society.

Mailler, R., and Lesser, V. 2004. Solving Distributed Constraint Optimization Problems Using Cooperative Mediation. In *Proceedings of the International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS)*. Los Alamitos, CA: IEEE Computer Society.

Modi, P. J., and Veloso, M. 2005. Bumping Strategies for the Multiagent Agreement Problem. In *Proceedings of the International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS)*. New York: Association for Computing Machinery.

Modi, P. J.; Shen, W.; Tambe, M.; and Yokoo, M. 2005. Adopt: Asynchronous Distributed Constraint Optimization With Quality Guarantees. *Artificial Intelligence* 161(1–2): 149–180.

Pynadath, D. V., and Tambe, M. 2001. Revisiting Asimov's First Law: A Response to the Call to Arms. In *Intelligent Agents VIII: The Eighth Workshop on Agent Theories, Architectures and Languages (ATAL-01), Lecture Notes in Computer Science* 2333, 307–320. Berlin: Springer.

Pynadath, D. V., and Tambe, M. 2003. Automated Teamwork among Heterogeneous Software Agents and Humans. *Journal of Autonomous Agents and Multi-Agent Systems* 7(1): 71–100.

Pynadath, D.; Tambe, M.; Arens, Y.; Chalupsky, H.; Gil, Y.; Knoblock, C.; Lee, H.; Lerman, K.; Oh, J.; Ramachandran, S.; Rosenbloom, P.; and Russ, T. 2000. Electric Elves: Immersing an Agent Organization in a Human Organization. In *Socially Intelligent Agents: The Human in the Loop*. Papers from the 2000 Fall Symposium, 150–154. AAAI Technical Report FS-00-04. Menlo Park, CA: AAAI Press.

Scerri, P.; Pynadath, D.; and Tambe, M. 2002. Towards Adjustable Autonomy for the Real-World. *Journal of Artificial Intelligence Research* 17: 171–228.

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

Tambe, M.; Pynadath, D.; and Chauvat, N. 2000. Building Dynamic Agent Organizations in Cyberspace. *IEEE Internet Computing* 4(2): 65–73.

Varakantham, P.; Maheswaran, R.; Gupta, T.; and Tambe, M. 2007. Towards Efficient Computation of Error Bounded Solutions in POMDPs: Expected Value Approximations and Dynamic Disjunctive Beliefs. In *Proceedings of the Twentieth International Joint Conference on Artificial Intelligence* (IJCAI-07). Menlo Park, CA: AAAI Press.

Varakantham, P.; Maheswaran, R.; and Tambe, M. 2005. Exploiting Belief Bounds: Practical POMDPs for Personal Assistant Agents. In *Proceedings of the International Joint Conference on Autonomous Agents and Multiagent Systems*. New York: Association for Computing Machinery.

Milind Tambe is a professor of computer science at the University of Southern California.

Emma Bowring is a faculty member in the School of Engineering and Computer Science at the University of the Pacific, Stockton, California.

Jonathan P. Pearce is working in quantitative finance in New York city.

Pradeep Varakantham is a postdoctoral fellow in the Robotics Institute at Carnegie Mellon University.

Paul Scerri is a systems scientist in the Robotics Institute at Carnegie Mellon University.

David V. Pynadath is a research scientist in the Inteligent Systems division of the USC Information Sciences Institute.