

## Chapter 1

# ELECTRIC ELVES: ADJUSTABLE AUTONOMY IN REAL-WORLD MULTI-AGENT ENVIRONMENTS

David V. Pynadath

Milind Tambe

*University of Southern California Information Sciences Institute*

*4676 Admiralty Way, Marina del Rey, CA 90292*

{pynadath,tambe}@isi.edu

**Abstract** Through *adjustable autonomy* (AA), an agent can dynamically vary the degree to which it acts autonomously, allowing it to exploit human abilities to improve its performance, but without becoming overly dependent and intrusive. AA research is critical for successful deployment of agents to support important human activities. While most previous work has focused on individual agent-human interactions, this paper focuses on *teams* of agents operating in real-world human organizations, as well as the novel AA coordination challenge that arises when one agent's inaction while waiting for a human response can lead to potential miscoordination. Our multi-agent AA framework, based on Markov decision processes, provides an adaptive model of users that reasons about the uncertainty, costs, and constraints of decisions. Our approach to AA has proven essential to the success of our deployed Electric Elves system that assists our research group in rescheduling meetings, choosing presenters, tracking people's locations, and ordering meals.

## 1. Introduction

Software agents support critical human activities in intelligent homes (Lesser et al., 1999), electronic commerce (Collins et al., 2000), long-term space missions (Dorais et al., 1998), etc. Future human organizations will be even more highly agentized, with software agents supporting information gathering, planning, and execution monitoring, as well as having increased control of resources and devices. This agentization will assist organizations of all types, whether military, corporate, or educational. For example, in a research institution, agentization may facilitate meeting organization, paper composition, software development, etc. We envision agent proxies for each person within

an organization. Thus, for instance, if an organization requires a deployment of people and equipment, then agent proxies could volunteer on behalf of the people or resources they represent, while also ensuring that the selected team collectively possesses sufficient resources and capabilities. The proxies could also monitor the progress of the participants and of the mission as a whole, executing corrective actions when necessary.

Applications of agents within human organizations have fostered an increasing interest in *adjustable autonomy* (AA), where agents dynamically adjust their own level of autonomy, harnessing human skills and knowledge as appropriate, without overly burdening the humans. When agents are embedded in large human organizations, they must also coordinate with each other and act jointly in teams. The requirements of teamwork and coordination give rise to novel AA challenges not addressed by previous research, which focuses on interactions between only an individual agent and its human user (Ferguson et al., 1996; Horvitz et al., 1999; Dorais et al., 1998). In particular, the *AA coordination challenge* arises during the transfer of decision-making control. In a team setting, an agent cannot transfer control freely, because as the agent waits for a human response, its teammates expect it to still fulfill its responsibilities to the overall joint task. Thus, the AA coordination challenge requires that an agent weigh possible team miscoordination while waiting for a human response against possible erroneous actions as a result of uninformed decisions.

We have conducted our research on AA using a real-world multi-agent system, *Electric Elves* (E-Elves) (Chalupsky et al., 2001), that we have used since June 1, 2000, at USC/ISI. E-Elves assists a group of 10 users in their daily activities. To address the AA coordination challenge, E-Elves agents use Markov decision processes (MDPs) (Puterman, 1994) to explicitly reason about team coordination via a novel three-step approach. First, before transferring decision-making control, an agent explicitly weighs the cost of waiting for user input and any potential team miscoordination against the cost of erroneous autonomous action. Second, agents do not rigidly commit to transfer-of-control decisions (as is commonly done in previous work), but instead reevaluate decisions as required. Third, an agent can change coordination arrangements, postponing or reordering activities, to “buy time” to lower decision cost/uncertainty. Overall, the agents look ahead at possible sequences of coordination changes, selecting one that maximizes team benefits.

## 2. Electric Elves

As a step towards agentization of large-scale human organizations, the Electric Elves effort at USC/ISI has had an agent team of 15 agents, including 10 proxies (for 10 people), running 24/7 since June 1, 2000, at USC/ISI (Chalupsky et al., 2001). The 5 other agents provide additional functionality for match-

ing users' interests and capabilities and for extracting information from Web sites. Each agent proxy is called Friday (from Robinson Crusoe's servant Friday) and acts on behalf of its user in the agent team. If a user is delayed to a meeting, Friday can reschedule the meeting, informing other Fridays, who in turn inform their human 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 and track the user's location. Friday communicates with users using wireless devices, such as Palm Pilots and WAP-enabled mobile phones, and via user workstations. We have used Friday's location reasoning to construct a People Locator that publishes the whereabouts of members of our research group on a Web page. This automatically updated information provides a cheap means for increasing social awareness (similar to previous work in the field (Tollmar et al., 1996)).

AA is of critical importance in Friday agents. Clearly, the more autonomous Friday is, the more time it saves its user. However, Friday has the potential to make costly mistakes when acting autonomously (e.g., volunteering an unwilling user for a presentation). Thus, each Friday must make intelligent decisions about when to consult its user and when to act autonomously. Furthermore, Friday faces significant, unavoidable uncertainty (e.g., if a user is not at the meeting location at meeting time, does s/he plan to attend?).

In addition to uncertainty and cost, the E-Elves domain raises the AA coordination challenge. Suppose that, when faced with uncertainty, a Friday agent consults its user (e.g., to check whether the user plans to attend a meeting), but the user, caught in traffic, fails to respond. While waiting for a response, Friday may miscoordinate with its teammates (other Friday agents), since it fails to inform them whether the user will attend the meeting. This, in turn means that other meeting attendees (humans) waste their time waiting. Conversely, if, to maintain coordination, Friday tells the other Fridays that its user will not attend the meeting, but the user does indeed plan to attend, the human team suffers a potentially serious cost from receiving this incorrect information. Friday must instead make a decision that makes the best tradeoff possible between the possible costs of inaction and the possible costs of incorrect action.

### **3. Decision-Tree Approach to AA**

Our first attempt at AA in E-Elves was inspired by CAP (Mitchell et al., 1994), an agent system for helping a user schedule meetings. Like CAP, Friday learned user preferences using C4.5 decision-tree learning (Quinlan, 1993). Although initial tests were promising (Tambe et al., 2000), when we deployed the resulting system 24/7, it led to some dramatic failures, including:

- 1 Tambe's Friday incorrectly, autonomously cancelled a meeting with the division director. C4.5 over-generalized from training examples.
- 2 Pynadath's Friday incorrectly cancelled a meeting. A time-out forced the choice of an (incorrect) autonomous action when Pynadath did not respond.



Figure 1.1. Dialog for meetings

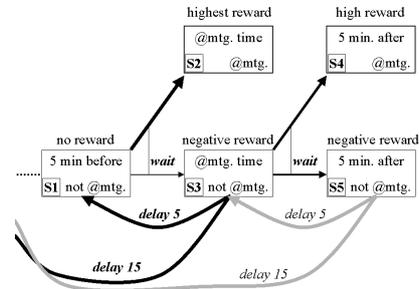


Figure 1.2. A small portion of simplified version of the delay MDP

- 3 A Friday delayed a meeting almost 50 times, each time by 5 minutes, ignoring the nuisance to the rest of the meeting participants.
- 4 Tambe's proxy automatically volunteered him for a presentation, though he was actually unwilling. Again, C4.5 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 approach faced some fundamental problems. The first problem was the AA coordination challenge. Learning from user input, when combined with timeouts, failed to address the challenge, since the agent sometimes had to take autonomous actions although it was ill-prepared to do so (examples 2 and 4). Second, the approach did not consider the team cost of erroneous autonomous actions (examples 1 and 2). Effective agent AA needs explicit reasoning and careful tradeoffs when dealing with the different individual and team costs and uncertainties. Third, decision-tree learning lacked the lookahead ability to plan actions that may work better over the longer term. For instance, in example 3, each five-minute delay is appropriate *in isolation*, but the rules did not consider the ramifications of one action on successive actions. Planning could have resulted in a one-hour delay instead of many five-minute delays. Planning and consideration of cost could also lead to an agent taking the low-cost action of a short meeting delay while it consults the user regarding the higher-cost cancel action (example 1).

#### 4. MDPs for Adjustable Autonomy

MDPs were a natural choice for addressing the issues identified in the previous section: reasoning about the costs of actions, handling uncertainty, planning for future outcomes, and encoding domain knowledge. The delay MDP, typical of MDPs in Friday, represents a class of MDPs covering all types of meetings for which the agent may take rescheduling actions. For each meeting, an agent can autonomously perform any of the 10 actions shown in the

dialog of Figure 1.1. It can also wait, i.e., sit idly without doing anything, or can reduce its autonomy and ask its user for input.

The delay MDP reasoning is based on a world state representation, the most salient features of which are the user's location and the time. Figure 1.2 shows a portion of the state space, showing only the location and time features, as well as some of the state transitions (a transition labeled "delay  $n$ " corresponds to the action "delay by  $n$  minutes"). Each state also has a feature representing the number of previous times the meeting has been delayed and a feature capturing what the agent has told the other Fridays about the user's attendance. There are a total of 768 possible states for each individual meeting.

The delay MDP's reward function has a maximum in the state where the user is at the meeting location when the meeting starts, giving the agent incentive to delay meetings when its user's late arrival is possible. However, the agent could choose arbitrarily large delays, virtually ensuring the user is at the meeting when it starts, but forcing other attendees to rearrange their schedules. This team cost is considered by incorporating a negative reward, with magnitude proportional to the number of delays so far and the number of attendees, into the delay reward function. However, explicitly delaying a meeting may benefit the team, since without a delay, the other attendees may waste time waiting for the agent's user to arrive. Therefore, the delay MDP's reward function includes a component that is negative in states after the start of the meeting if the user is absent, but positive otherwise. The reward function includes other components as well and is described in more detail elsewhere (Scerri et al., 2001).

The delay MDP's state transitions are associated with the probability that a given user movement (e.g., from office to meeting location) will occur in a given time interval. Figure 1.2 shows multiple transitions due to a 'wait' action, with the relative thickness of the arrows reflecting their relative probability. The "ask" action, through which the agent gives up autonomy and queries the user, has two possible outcomes. First, the user may not respond at all, in which case, the agent is performing the equivalent of a "wait" action. Second, the user may respond, with one of the 10 responses from Figure 1.1. A communication model (Tambe et al., 2000) provides the probability of receiving a user's response in a given time step. The cost of the "ask" action is derived from the cost of interrupting the user (e.g., a dialog box on the user's workstation is cheaper than sending a page to the user's cellular phone). We compute the expected value of user input by summing over the value of each possible response, weighted by its likelihood.

Given the states, actions, probabilities, and rewards of the MDP, Friday uses the standard value iteration algorithm to compute an optimal policy, specifying, for each and every state, the action that maximizes the agent's expected utility (Puterman, 1994). One possible policy, generated for a subclass of possible meetings, specifies "ask" and then "wait" in state **S1** of Figure 1.2, i.e., the

agent gives up some autonomy. If the world reaches state **S3**, the policy again specifies “wait”, so the agent continues acting without autonomy. However, if the agent then reaches state **S5**, the policy chooses “delay 15”, which the agent then executes autonomously. However, the exact policy generated by the MDP will depend on the exact probabilities and costs used. The delay MDP thus achieves the first step of Section 1’s three-step approach to the AA coordination challenge: balancing individual and team rewards, costs, etc.

The second step of our approach requires that agents avoid rigidly committing to transfer-of-control decisions, possibly changing its previous autonomy decisions. The MDP representation supports this by generating an autonomy *policy* rather than an autonomy *decision*. The policy specifies optimal actions for each state, so the agent can respond to any state changes by following the policy’s specified action for the new state (as illustrated by the agent’s retaking autonomy in state **S5** by the policy discussed in the previous section). In this respect, the agent’s AA is an ongoing process, as the agent acts according to a policy throughout the entire sequence of states it finds itself in.

The third step of our approach arises because an agent may need to act autonomously to avoid miscoordination, yet it may face significant uncertainty and risk when doing so. In such cases, an agent can carefully plan a change in coordination (e.g., delaying actions in the meeting scenario) by looking ahead at the future costs of team miscoordination and those of erroneous actions. The delay MDP is especially suitable for producing such a plan because it generates policies after looking ahead at the potential outcomes. For instance, the delay MDP supports reasoning that a short delay buys time for a user to respond, reducing the uncertainty surrounding a costly decision, albeit at a small cost.

Furthermore, the lookahead in MDPs can find effective long-term solutions. As already mentioned, the cost of rescheduling increases as more and more such repair actions occur. Thus, even if the user is very likely to arrive at the meeting in the next 5 minutes, the uncertainty associated with that particular state transition may be sufficient, when coupled with the cost of subsequent delays if the user does not arrive, for the delay MDP policy to specify an initial 15-minute delay (rather than risk three 5-minute delays).

## 5. Evaluation of Electric Elves

We have used the E-Elves system within our research group at USC/ISI, 24 hours/day, 7 days/week, since June 1, 2000 (occasionally interrupted for bug fixes and enhancements). The fact that E-Elves users were (and still are) willing to use the system over such a long period and in a capacity so critical to their daily lives is a testament to its effectiveness. Our MDP-based approach to AA has provided much value to the E-Elves users, as attested to by the 689 meetings that the agent proxies have monitored over the first six months of

execution. In 213 of those meetings, an autonomous rescheduling occurred, indicating a substantial savings of user effort. Equally importantly, humans are also often intervening, leading to 152 cases of user-prompted rescheduling, indicating the critical importance of AA in Friday agents.

The general effectiveness of E-Elves is shown by several observations. Since the E-Elves deployment, the group members have exchanged very few email messages to announce meeting delays. Instead, Fridays autonomously inform users of delays, thus reducing the overhead of waiting for delayed members. Second, the overhead of sending emails to recruit and announce a presenter for research meetings is now assumed by agent-run auctions. Third, the People Locator is commonly used to avoid the overhead of trying to manually track users down. Fourth, mobile devices keep us informed remotely of changes in our schedules, while also enabling us to remotely delay meetings, volunteer for presentations, order meals, etc. We have begun relying on Friday so heavily to order lunch that one local Subway restaurant owner even suggested marketing to agents: *“More and more computers are getting to order food, so we might have to think about marketing to them!!”*

Most importantly, over the entire span of the E-Elves’ operation, the agents have *never* repeated any of the catastrophic mistakes that Section 3 enumerated in its discussion of our preliminary decision-tree implementation. For instance, the agents do not commit error 4 from Section 3 because of the domain knowledge encoded in the bid-for-role MDP that specifies a very high cost for erroneously volunteering the user for a presentation. Likewise, the agents never committed errors 1 or 2. The policy described in Section 4 illustrates how the agents would first ask the user and then try delaying the meeting, before taking any final cancellation actions. The MDP’s lookahead capability also prevents the agents from committing error 3, since they can see that making one large delay is preferable, in the long run, to potentially executing several small delays. Although the current agents do occasionally make mistakes, these errors are typically on the order of asking the user for input a few minutes earlier than may be necessary, etc. Thus, the agents’ decisions have been reasonable, though not always optimal. Unfortunately, the inherent subjectivity in user feedback makes a determination of optimality difficult.

## 6. Conclusion

Gaining a fundamental understanding of AA is critical if we are to deploy multi-agent systems in support of critical human activities in real-world settings. Indeed, living and working with the E-Elves has convinced us that AA is a critical part of any human collaboration software. Because of the negative result from our initial C4.5-based approach, we realized that such real-world, multi-agent environments as E-Elves introduce novel challenges in AA that

previous work has not addressed. For resolving the *AA coordination challenge*, our E-Elves agents explicitly reason about the costs of team miscoordination, they flexibly transfer autonomy rather than rigidly committing to initial decisions, and they may change the coordination rather than taking risky actions in uncertain states. We have implemented our ideas in the E-Elves system using MDPs, and our AA implementation now plays a central role in the successful 24/7 deployment of E-Elves in our group. Its success in the diverse tasks of that domain demonstrates the promise that our framework holds for the wide range of multi-agent domains for which AA is critical.

## Acknowledgments

This research was supported by DARPA award No. F30602-98-2-0108 (Control of Agent-Based Systems) and managed by ARFL/Rome Research Site.

## References

- Chalupsky, H., Gil, Y., Knoblock, C. A., Lerman, K., Oh, J., Pynadath, D. V., Russ, T. A., and Tambe, M. (2001). Electric elves: Applying agent technology to support human organizations. In *Proc. of the IAAI. Conf.*
- Collins, J., Bilot, C., Gini, M., and Mobasher, B. (2000). Mixed-init. dec.-supp. in agent-based auto. contracting. In *Proc. of the Conf. on Auto. Agents.*
- Dorais, G. A., Bonasso, R. P., Kortenkamp, D., Pell, B., and Schreckenghost, D. (1998). Adjustable autonomy for human-centered autonomous systems on mars. In *Proc. of the Intn'l Conf. of the Mars Soc..*
- Ferguson, G., Allen, J., and Miller, B. (1996). TRAINS-95 : Towards a mixed init. plann. asst. In *Proc. of the Conf. on Art. Intell. Plann. Sys.*, pp. 70–77.
- Horvitz, E., Jacobs, A., and Hovel, D. (1999). Attention-sensitive alerting. In *Proc. of the Conf. on Uncertainty and Art. Intell.*, pp. 305–313.
- Lesser, V., Atighetchi, M., Benyo, B., Horling, B., Raja, A., Vincent, R., Wagner, T., Xuan, P., and Zhang, S. X. (1999). A multi-agent system for intelligent environment control. In *Proc. of the Conf. on Auto. Agents.*
- Mitchell, T., Caruana, R., Freitag, D., McDermott, J., and Zabowski, D. (1994). Exp. with a learning personal asst. *Comm. of the ACM*, 37(7):81–91.
- Puterman, M. L. (1994). *Markov Decision Processes*. John Wiley & Sons.
- Quinlan, J. R. (1993). *C4.5: Progs. for Mach. Learn.* Morgan Kaufmann.
- Scerri, P., Pynadath, D. V., and Tambe, M. (2001). Adjustable autonomy in real-world multi-agent environments. In *Proc. of the Conf. on Auto. Agents.*
- Tambe, M., Pynadath, D. V., Chauvat, N., Das, A., and Kaminka, G. A. (2000). Adaptive agent integration architectures for heterogeneous team members. In *Proc. of the Intn'l Conf. on MultiAgent Sys.*, pp. 301–308.
- Tollmar, K., Sandor, O., and Schömer, A. (1996). Supp. soc. awareness: @Work design & experience. In *Proc. of the ACM Conf. on CSCW*, pp. 298–307.