Adjustable Autonomy: A Response

Milind Tambe, David Pynadath, Paul Scerri

Information Sciences Institute, University of Southern California
4676 Admiralty Way, Marina del Rey, CA 90292
tambe@isi.edu

Gaining a fundamental understanding of adjustable autonomy (AA) is critical if we are to deploy multi-agent systems in support of critical human activities. Indeed, our recent work with intelligent agents in the "Electric Elves" (E-Elves) system has convinced us that AA is a critical part of any human collaboration software. In the following, we first briefly describe E-Elves, then discuss AA issues in E-Elves.

Electric Elves: A Deployed Multi-agent System

The past few years have seen a revolution in the field of software agents, with agents now proliferating in human organizations, helping individuals in information gathering, activity scheduling, managing email, etc. The E-Elves effort at USC/ISI is now taking the next step: dynamic teaming of all such different heterogeneous agents, as well as proxy agents for humans, to serve not just individuals, but to facilitate the functioning of entire organizations. The ultimate goal of our work is to build agent teams that assist in all organization activities, enabling organizations to act coherently, to robustly attain their mission goals and to react swiftly to crises, e.g., helping a disaster rescue organization to coordinate movement of personnel and equipment to the site of a disaster. The results of this work could potentially be relevant to all organizations.

As a step towards this goal, we have had an agent team of 15 agents, including 10 proxies (for 10 people) running 24/7 for the past four months at USC/ISI. Each proxy is called Friday (from Robinson Crusoe's Friday), and it acts on the behalf of its user in the agent team. Thus, if a user is delayed to a meeting, then Friday will reschedule that meeting, by informing other Fridays, which in turn will inform the humans users. If there is a research presentation slot open, Friday may volunteer or decline the invitation for that slot. In addition, Friday can also order a user's meals — a user can say "order my usual" and Friday will select a nearby restaurant such as *California Pizza Kitchen* and send over a fax to order the meal from the user's usual favorites. Friday communicates with a user using different types of mobile wireless devices, such as PALM VIIs and WAP enabled mobile phones. By connecting a PALMVII to a GPS, Friday can also track our locations using wireless transmission.

Each Friday is based on a teamwork model called STEAM[3], which helps it communicate and coordinate with other Fridays. One interesting new development wrt STEAM is that roles in teamwork are now auctioned off. In particular, some meetings have a presenter role. Given a topic of presentation, Friday bids on behalf of its user, indicating if its user is capable and/or willing for that topic. Here, a Friday bids autonomously on capability by looking up user's capability in a capability database, but its willingness decision is not autonomous. The highest bidder wins the auction and gets the preseter role.

Adjustable Autonomy in Electric Elves

AA is of critical importance in Friday agents. Clearly, the more decisions that Friday makes autonomously, the more time its user saves. Yet, given the high uncertainty in Friday's beliefs about its user's state, it could potentially make very costly mistakes while acting autonomously, e.g., it may order an expensive dinner when the user is not hungry, or volunteer a busy user for a presentation. Thus, each Friday must make intelligent decisions about when to consult its user and when to act autonomously.

One key problem here is that a Friday agent faces significant uncertainty in its autonomous decision, e.g., if a user is not at the meeting location at meeting time, does he/she plan to attend? To address such uncertainty, our initial attempt at AA in E-Elves was inspired by CAP[1], the well-known agent system for advising a human user on scheduling meetings. As with CAP, Friday learned user preferences using C4.5[2] decision-tree learning, although Friday's focus was on rescheduling meetings. Thus, in the training mode, Friday recorded values of a dozen carefully selected attributes and also the user's preferred action (by querying him/her using a dialog box as shown in Figure 1). This recorded data and user response was used to learn a decision tree, e.g., *if* the user has a meeting with his/her advisor, but the user is not at ISI at meeting time, *then* delay the meeting 15 minutes. Simultaneously, Friday queried the user if s/he wanted Friday to take the decision autonomously or not; C4.5 was again used to learn a second decision tree from these responses.

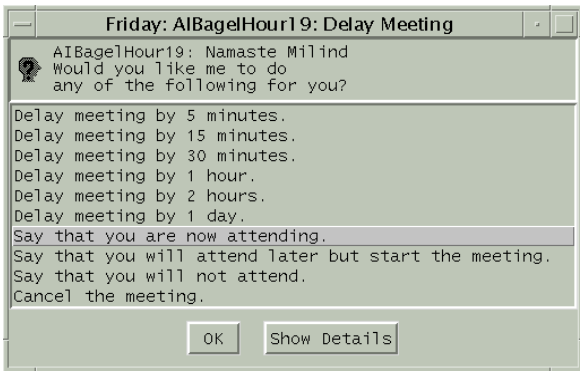

Fig. 1. Dialog boxes used in Electric Elves.

Initial tests based on the above setup were successful, as reported in [4]. Soon thereafter however, one key problem became apparent: a user would suggest Friday to not take some specific decision autonomously, but then s/he would not be available to provide any input. Thus, a Friday would end up waiting for user input, and miscoordinate with its team. To address this *team miscoordination* problem, timeouts were introduced: if a user did not respond within a time-limit, Friday used its own decision tree to take autonomous action. In our initial tests, the results still looked promising. Unfortunately, when the resulting system was deployed for real, it led to some dramatic failures:

1. Tambe's (a user) Friday incorrectly cancelled a meeting with his division's director. C4.5 had overgeneralized, incorrectly taking an autonomous action from the initial set of training examples.
2. Pynadath's (another user) Friday incorrectly cancelled the group's weekly research meeting. The time-out forced an incorrect autonomous action when Pynadath was unavailable to respond in time.
3. One of the Fridays delayed a meeting almost 50 times, each in 5 minute increments. The agent was applying its learned rule to cause a small delay each time, but ignoring the nuisance to the rest of the meeting participants.
4. Tambe's proxy automatically volunteered him for a presentation, even though he was not willing. Again, C4.5 had overgeneralized from a few examples and with timeout, taken an undesirable autonomous action.

From the growing list of failures, it became increasingly clear that our original approach faced some fundamental problems. The first problem is clearly that agents must balance the possibility of team miscoordination against effective team action. Learning from user input combined with timeouts, failed to address this challenge: the agent was sometimes forced to take autonomous actions when it was ill-prepared, causing problems as seen in example 2 and 4. Second, C4.5 was not considering the cost to the team due to erroneous autonomous actions, e.g., an erroneous cancellation, as seen in example 1 and 2. Third, decision-tree learning lacked the ability to look-ahead, to plan actions that would work better in the longer term. For instance, in example 3, each 5 minute delay is appropriate for its corresponding state *in isolation*, but the C4.5 rules did not take into account the consequences of one action on future actions. Such planning could have preferred a one hour delay instead of several 5 minute delays.

Thus, one major challenge in AA, based on our experience with C4.5 is guaranteeing *safe learning* in AA. In particular, agents may often learn in the presence of noisy data, e.g., they may be unable to observe that a user is attending a meeting on time. Yet, the increased need for autonomous action in teams may lead agents to act despite such faulty learning, making highly inaccurate decisions and causing drastic team failures. One argument here is that if agents wait long enough to collect a lot more data, they would overcome such problems; but in rich domains, it would be difficult to first gather the required amount of training data in any reasonable amount of time. Thus, we need a safety mechanism to protect the agent team from temporary distortions in learning.

References

1. Tom Mitchell, Rich Caruana, Dayne Freitag, John McDermott, and David Zabowski. Experience with a learning personal assistant. *Communications of the ACM*, 37(7):81–91, July 1994.
2. J. R. Quinlan. *C4.5: Programs for machine learning*. Morgan Kaufmann, San Mateo, CA, 1993.
3. M. Tambe. Towards flexible teamwork. *Journal of Artificial Intelligence Research (JAIR)*, 7:83–124, 1997.
4. Milind Tambe, David V. Pynadath, Nicolas Chauvat, Abhimanyu Das, and Gal A. Kaminka. Adaptive agent integration architectures for heterogeneous team members. In *Proceedings of the International Conference on MultiAgent Systems*, pages 301–308, 2000.