Adjustable Autonomy Challenges in Personal Assistant Agents: A Position Paper

Rajiv T. Maheswaran[1], Milind Tambe[1], Pradeep Varakantham[1], and Karen Myers[2]

[1] University of Southern California
Salvatori Computer Science Center
Los Angeles, CA 90089
and
[2] SRI International
333 Ravenswood Avenue
Menlo Park, CA 94025

Abstract. The successful integration and acceptance of many multi-agent systems into daily lives crucially depends on the ability to develop effective policies for adjustable autonomy. Adjustable autonomy encompasses the strategies by which an agent selects the appropriate entity (itself, a human user, or another agent) to make a decision at key moments when an action is required. We present two formulations that address this issue: user-based and agent-based autonomy. Furthermore, we discuss the current and future implications on systems composed of personal assistant agents, where autonomy issues are of vital interest.

1 Introduction

Increasingly, researchers have focused on deploying multi-agent systems to support humans in critical activities, revolutionizing the way a variety of cooperative tasks are carried out, at home, at the office, in a large organization, or in the field. For example, multi-agent teams could support rapid response to natural (or man-made) disasters, such as earthquakes or assist coordination of autonomous spacecraft. One promising application of interest to many organizations is the development of personal assistant agents [2, 3, 10]. These entities would be capable of performing a variety of tasks such as scheduling meetings, gathering information or communicating on behalf of its user. Unburdened by routine or mundane tasks, humans would free up time for more productive pursuits.

To be able to perform their roles as assistants, agents will need to be endowed with a measure of autonomy. This raises many issues as agents may not have the information or ability to carry out a required action or the human user may not want agents to make certain decisions. These concerns have led to an emergence of the study of *adjustable autonomy* , i.e. agents dynamically varying their own autonomy, transferring decision making control to other entities (typically human users) in key situations [9, 12]. One personal assistant agent project where adjustable autonomy issues are being addressed is CALO [2]. The goal is to create an agent that will observe and learn while interacting with users, other agents and the environment enabling it to handle a vast set of interrelated decision-making tasks that remain unaddressed by today's technology. This next-generation personal assistant agent and many similar multi-agent systems will critically depend on the ability of today's research community to answer the challenges of adjustable autonomy.

Arguably, the fundamental research question is determining whether and when transfers of control should occur from the agent to other entities. Answering this question is very important for the development of personal assistant agents (along with other multi-agent systems with similar autonomy issues). An effective autonomous agent must be able to obtain the trust of a user through it's action policies. These policies must

balance the ability to make competent and timely decisions without treading into domains where the user wants sole control.

In this position paper, we present two approaches in dealing with issues of transfer of control in adjustable autonomy: user-based and agent-based adjustable autonomy. We also discuss future directions of research and how they will be applied to a next-generation personal assistant agent.

2 User-based autonomy

User-based adjustable autonomy is driven by the need to support user control over the activities of an agent at runtime. In situations where activities are routine and decisions are straightforward, a human may be content to delegate all problem-solving responsibility to an agent. However, in situations where missteps could have undesirable consequences, the human needs the ability to control the problem-solving process to ensure that appropriate actions are taken. The ideal in such situations is not to devolve all responsibility to the human. Rather, the human should be able to guide the agent in much the same way that a human manager would direct a subordinate on tasks that were beyond his or her capabilities. Thus, what is required is a form of supervised autonomy for the agent [1].

The requirement for user-driven adjustable autonomy stems from two main issues: capability and personalization.

- *Capability*: For many applications, it will be difficult to develop agents whose problem-solving capabilities match or even approach those of humans. Agents can still play a valuable role in such applications by performing routine tasks that do not require user intervention and by working under user supervision on more complex tasks. For example, an office assistant agent may be capable of scheduling meetings that do not involve changes to the users current commitments, but require human guidance if commitments need to be revised.

- *Personalization*: Many decisions do not have a clear "best" answer, as they depend on multiple factors that can interact in complex ways. An individual user may have his own preference as to how to respond in different situations, making it impossible to precode appropriate responses in an off-the-shelf agent. Additionally, those preferences may change over time. While automated learning techniques present one possible mechanism for addressing customization, they are a long way from providing the sophisticated adaptability needed for agent customization. Furthermore, they are inherently limited to projecting preferences based on past decisions.

The central issue for user-based adjustable autonomy agent is the design of mechanisms by which a user can dynamically modify the scope of autonomy for an agent. Such mechanisms should be natural, easy to use, and sufficiently expressive to enable fine-grained specifications of autonomy levels.

One approach to user-based adjustable autonomy is oriented around the notion of policies [9]. A policy can be considered a declarative statement that explicitly bounds the activities that an agent is allowed to perform without user involvement. Policies can be asserted and retracted throughout the scope of an agents operation, thus providing the means to tailor autonomy dynamically in accord with a users changing perspective on what he or she feels comfortable delegating to the agent. This comfort level may change as a result of the user acquiring more confidence in the agents ability to perform certain tasks, or the need for the user to focus his problem-solving skills on more important matters.

Policies could be defined from the perspective of either explicitly enabling or restricting agent activities. We are interested in domains where agents will generally need to operate with high degrees of autonomy.

For this reason, we assume a permissive environment: unless stated otherwise, agents are allowed to operate independently of human interaction. In this context, policies are used to limit the scope of activities that can be performed autonomously. Activities outside that set may still be performable, but require some form of interaction with the user.

Our policies assume a Belief-Desire-Intention (BDI) model of agency [11] in which an agent has a predefined library of parameterized plans that can be applied to achieve assigned tasks or respond to detected events. Within this framework, we consider two classes of policies: permission requirements for action execution and consultation requirements for decision making.

- *Permission Requirements*: Permission requirements declare conditions under which an agent must obtain authorization from the human supervisor before performing activities. For example, consider the directive *Obtain permission before scheduling any meetings after 5:00*.

- *Consultation Requirements*: Consultation requirements designate a class of decisions that should be deferred to the human supervisor. These decisions can relate to either the selection of a value for parameter instantiation (e.g., *Let me choose airline flights*) or the selection of a plan for a goal (e.g., *Consult me when deciding how to respond to requests to cancel staff meetings*).

These adjustable autonomy policies are part of a more general framework for agent guidance that enables high-level user management of agent activities [9]. An alternate class of policies, called strategy preference guidance, supports the specification of recommendations on how an agent should accomplish tasks. These preferences could indicate specific plans to employ restrictions on plans that should not be employed, as well as constraints on how plan parameters can be instantiated. For example, the directive *Use Expedia to find options for flying to Washington next week* expresses a preference over approaches to finding flight information for planning a particular trip. On the other hand, the directive *Dont schedule project-related meetings for Monday mornings* restricts the choice for instantiating parameters that denote project meeting times.

Given that strategy preference guidance can be used to restrict the set of problem-solving activities of an agent, it can be viewed as a mechanism for limiting agent autonomy. However, this form of guidance does not support the explicit transfer of decision-making control to the user, as is the case with most work in the area of adjustable autonomy. Rather, the users influence on the decision-making process is implicit in the guidance itself.

These policies are formulated in terms of high-level characterizations of classes of activities, goals, and decisions over which an agent has autonomy. The language used to encode policies makes use of a logical framework that builds on both the underlying agent domain theory, and a domain metatheory. The domain metatheory is an abstract characterization of the agent domain theory that specifies important semantic attributes of plans, planning parameters, and instances. For instance, it could be used to identify parameters that play certain roles within agent plans (e.g., parameters for project meeting times), or to distinguish alternative procedures according to properties such as cost or speed. This abstraction allows a user to express policies in compact, semantically meaningful terms without having to acquire detailed knowledge of the agents internal workings or constituent knowledge.

One nice feature of this policy-based approach to adjustable autonomy is that it has minimal set-up costs, requiring only the formulation of the domain metatheory for defining policies. As argued in [8], the metatheory should be a natural by-product of a principled approach to domain modeling. Enforcement of the policies for adjustable autonomy can be done via a simple filtering mechanism that overlays standard BDI executor models. This filtering mechanism adds little to the computation cost of agent execution, as it requires simple matching of policies to plan and task properties.

3 Agent-based autonomy

In settings with agent-based adjustable autonomy, domains of authority are not prescribed by the user. Rather, an agent explicitly reasons about whether and when to transfer decision-making control to another entity. If control is to be transferred, an agent must choose the appropriate entity to which to yield responsibility. A policy to transfer control for a decision or action needs to balance the likely benefits of giving control to a particular agent or human with the potential costs of doing so, thus the key challenge is to balance two potentially conflicting goals. On one hand, to ensure that the highest-quality decisions are made, an agent can transfer control to a human user (or another agent) whenever that entity has superior decision-making expertise. On the other hand, interrupting a user has high costs and the user may be unable to make and communicate a decision, thus such transfers-of-control should be minimized.

The work so far in agent-based AA has examined several different techniques that attempt to balance these two conflicting goals and thus address the transfer-of-control problem. For example, one technique suggests that decision-making control should be transferred if the expected utility of doing so is higher than the expected utility of making an autonomous decision [7]. A second technique uses uncertainty as the sole rationale for deciding who should have control, forcing the agent to relinquish control to the user whenever uncertainty is high [6]. Yet other techniques transfer control to a user if an erroneous autonomous decision could cause significant harm [4] or if the agent lacks the capability to make the decision [5].

Previous work has investigated various approaches to addressing this problem but has often focused on individual agent-human interactions, in service of single tasks. Unfortunately, domains requiring collaboration between teams of agents and humans reveals at least two key shortcomings of these previous approaches. First, these approaches use rigid one-shot transfers of control that can result in unacceptable coordination failures in multi-agent settings. Second, they ignore costs (e.g., in terms of time delays or effects on actions) to an agent's team due to such transfers of control.

To remedy these problems, we base a novel approach to adjustable autonomy on the notion of *transfer-of-control strategy*. A transfer-of-control strategy consists of a conditional sequence of two types of actions: (i) actions to transfer decision-making control (e.g., from the agent to the user or vice versa) and (ii) actions to change an agent's pre-specified coordination constraints with team members, aimed at minimizing miscoordination costs. The goal is for high quality individual decisions to be made with minimal disruption to the coordination of the team. Strategies are operationalized using Markov decision processes (MDPs) which select the optimal strategy given an uncertain environment and costs to individuals and teams. A general reward function and state representation for an MDP have been developed, to help enable implementation such strategies [12].

An agent strategy can be an ordering, composed of authority owners and constraint changes, which outlines a particular sequence of responsibility states interconnected by temporal limits. For instance, a strategy denoted ADH implies that an agent (A) initially attempts to make an autonomous decision. If the agent makes the decision autonomously the strategy execution ends there. However, there is a chance that it is unable to make the decision in a timely manner, perhaps because its computational resources are busy with higher priority tasks or a high quality decision cannot be made due to lack of information. To avoid miscoordination, the agent executes a D action that changes the coordination constraints on the activity. For example, a D action could be to inform other agents that the coordinated action will be delayed, thus incurring a cost of inconvenience to others but buying more time to make the decision. If the agent still cannot make the decision, it will eventually transfer decision-making control to the human (H) and wait for a response.

Transfer-of-control strategies provide a flexible approach to adjustable autonomy in complex systems with many actors. By enabling multiple transfers-of-control between two (or more) entities, rather than rigidly committing to one entity (i. e., A or H), a strategy attempts to provide the highest quality decision, while

avoiding coordination failures. Thus, a key AA problem is to select the right strategy, i.e., one that provides the benefit of high quality decisions without risking significant costs such as interrupting the user or miscoordination with the team. Furthermore, an agent must select the right strategy despite significant uncertainty. Markov decision processes (MDPs) (Puterman, 1994) are a natural choice for implementing such reasoning because they explicitly represent costs, benefits and uncertainty as well as doing look ahead to examine the potential consequences of sequences of actions.

This agent-based autonomy approach has been tested in the context of a real-world multi-agent system, called Electric Elves (E-Elves) [3, 10] that was used for over six months at the University of Southern California and the Information Sciences Institute. Individual user proxy agents called Friday (from Robinson Crusoe's servant Friday) act in a team to assist with rescheduling meetings, ordering meals, finding presenters and other day-to-day activities. Figure 1 describes the interactions and architecture of the E-Elves project.

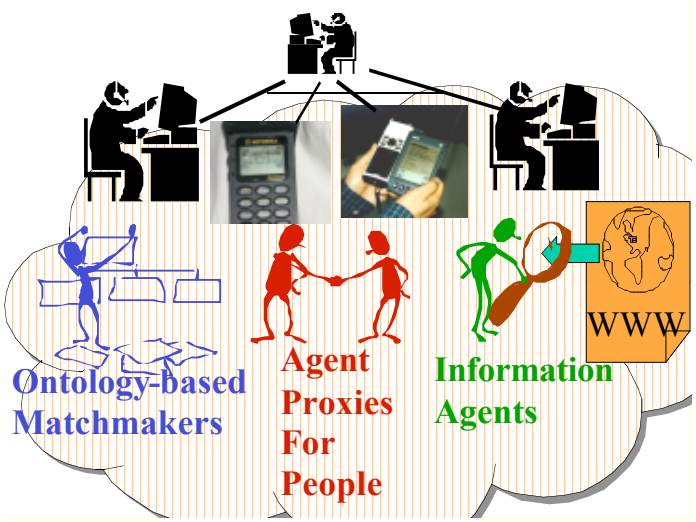

Fig. 1. Model for Interaction and Architecture of Electric Elves Project

MDP-based AA reasoning was used around the clock in the E-Elves, making many thousands of autonomy decisions. Despite the unpredictability of the user's behavior and the agent's limited sensing abilities, the MDP consistently made sensible AA decisions. Moreover, the agent often performed several transfers-of-control to cope with contingencies such as a user not responding.

4 Future of adjustable autonomy

The greatest challenges in adjustable autonomy involve bridging the gaps in the user-based and agent-based methodologies to create agents that can leverage the benefits of both systems. An ideal system would be able to include the situational sensitivity and personalization capabilities of the user-based autonomy while incorporating the autonomous adaptability, modeling of uncertainty in decision making and constraint modification aspects of the agent-based model. The result should be an agent customizable with low set-up cost, yet be able to factor in multi-agent issues when considering autonomy issues. A key to developing such a system will be resolving a fundamental formulation in the BDI and MDP modeling. This can be approached by applying hybrid schemes that weave both structures into their policy space. Another method may be to discover mapping from BDI plans to decision theoretic structures that yield congruent plans.

One lesson learned when actually deploying the system was that sometimes users wished to influence the AA reasoning, e.g., to ensure that control was transferred to them in particular circumstances. To enable users to influence the AA reasoning, safety constraints were introduced that allowed users to prevent agents from taking particular actions or ensuring that they do take particular actions. These safety constraints provide guarantees on the behavior of the AA reasoning, making the basic approach more generally applicable and, in particular, making it more applicable to domains where mistakes have serious consequences. Creating structures and models for safety constraints can be considered a step towards more effective adjustable autonomy systems that infuse personalization into agents.

Another area for exploration are situations where the underlying state is not known explicitly. One can envision scenarios where the current information available to a personal assistant agent does not allow it to deduce a single state from which to apply a policy. One way to approach this problem might be to assign a most probable state given every possible information set *a priori*, and apply a plan for the resulting probable state. This would become extremely cumbersome for many problem settings where the underlying state space or the information sets are large. Another approach is to apply Partially Observable Markov Decision Processes (POMDPs), into the models for autonomy decisions, allowing for uncertainty in both observation and action.

Acknowledgements
This work was sponsored by a subcontract from SRI International based on the DARPA CALO project.

References

[1] K.S. Barber and C.E. Martin. Dynamic adaptive autonomy in multi-agent systems: Representation and justification. *International Journal of Pattern Recognition and Artificial Intelligence*, 15(3), 2001.

[2] CALO, 2003. http://www.ai.sri.com/project/CALO.

[3] H. Chaulpsky, Y. Gil, C. Knoblock, J. Oh, K. Lerman Dnd D. Pynadath, T. Russ, and M. Tambe. Electric elves: Applying agent technology to support human organizations. In *International Conference on Innovative Applications of AI*, pages 51–58, 2001.

[4] G. Dorais, R. Kortenkamp, B. Pell, and D. Schreckenghost. Adjustable autonomy for human-centered autonomous systems on mars. In *proceedings of the First International Conference of the Mars Society*, pages 397–420, 1998.

[5] G. Ferguson, J. Allen, and B. Miller. Towards a mixed-initiative planning assistant. In *proceedings of the Third conference on Artificial Intelligence Planning Systems*, pages 70–77, 1996.

[6] J. Gunderson and W. Martin. Effects of uncertainty on variable autonomy in maintenance robots. In *Workshop on Autonomy Control Software*, pages 26–34, 1999.

[7] E. Horvitz, A. Jacobs, and D. Hovel. Attention-sensitive alerting. In *proceedings of Conference on Uncertainty and Artificial Intelligence*, pages 305–313, Stockholm, Sweden, 1999.

[8] K.L. Myers. Domain metatheories: Enabling user-centric planning. In *In proceedings of AAAI Workshop on Representational Issues for Real-World Planning Systems*, 2000.

[9] K.L. Myers and D.N. Morley. *Agent Autonomy*, chapter The TRAC Framework for Agent Directability. Kluwer Academic Publishers, 2002.

[10] D. Pynadath, M. Tambe, Y. Arens, H. Chalupsky, Y. Gil, C. Knoblock, H. Lee, J. Oh, K. Lerman, S. Kamachandran, P. Rosenbloom, and T. Russ. Electric elves: Immersing an agent organization in a human organization. In *Proceedings of the AAAI Fall symposium on socially intelligent agents - the human in the loop*, 2000.

[11] A.S. Rao and M.P. Georgeff. Bdi agents: From theory to practice. In *proceedings of International Conference on Multi-Agent Systems*, San Francisco, CA, 1995.

[12] Paul Scerri, David V. Pynadath, and Milind Tambe. Towards adjustable autonomy for the real world. *Journal of Artificial Intelligence Research*, 17, 2003.