Why the elf acted autonomously: Towards a theory of adjustable autonomy

Paul Scerri, David V. Pynadath, Milind Tambe
Information Sciences Institute and Computer Science Department
University of Southern California
4676 Admiralty Way, Marina del Rey, CA 90292
scerri@isi.edu, pynadath@isi.edu, tambe@isi.edu

ABSTRACT

Adjustable autonomy refers to agents' dynamically varying their own autonomy, transferring decision making control to other entities (typically human users) in key situations. Determining whether and when such transfer of control must occur is arguably the fundamental research question in adjustable autonomy. Practical systems have made significant in-roads in answering this question and in providing high-level guidelines for transfer of control decisions. For instance, [11] report that Markov decision processes were successfully used in transfer of control decisions in a real-world multiagent system, but that use of C4.5 led to failures. Yet, an underlying theory of transfer of control, that would explain such successes or failures is missing. To take a step in building this theory, we introduce the notion of a *transfer-of-control strategy*, which potentially involves several transfer of control actions. A mathematical model based on this notion allows both analysis of previously reported implementations and guidance for the design of new implementations. The practical benefits of this model are illustrated in a dramatic simplification of an existing adjustable autonomy system.

Categories and Subject Descriptors

I.2.11 [**Artificial Intelligence**]: Distributed Artificial Intelligence—*Intelligent Agents*

General Terms

Theory

1. INTRODUCTION

A growing number of applications require that agents have *adjustable autonomy*, i.e., that agents *dynamically adjust their own level of autonomy based on the situation* [4]. At the heart of adjustable autonomy (AA) is the question of whether and when agents should make autonomous decisions and when they should transfer decision-making control to other entities (e.g., human users).

Initial answers to this question, outlining specific transfer-of-control techniques and their effectiveness have already appeared in the literature[11, 3, 7]. For instance, one key report is by this paper's authors[11] and focused on AA in a real-world multiagent system called *Electric Elves*, that is deployed 24/7[1]. It is reported that the use of C4.5[9] led to dramatic failures in transfer-of-control decisions, but that the use of Markov decision processes (MDPs) led to satisfactory decisions. Similarly, Horvitz[7] presents a decision-theoretic approach to deciding whether to transfer control from an agent assistant to a user. Fleming[2] presents a technique based on thresholds of learned rules to decide if agents should transfer control to a user.

While such reports are useful, without an underlying, *domain-independent* model of AA, we cannot predict whether an approach that has success in one domain can translate that success to other domains. For instance, could Horvitz's[7] technique for transfer of control have worked in Electric Elves? Should we forever banish C4.5 from AA systems, given its failure in Electric Elves? Must all AA applications use complex MDPs instead? There have been informal answers given to these questions, e.g., that the context of multiagent teams influences the AA approach[11], but no domain-independent model or formal answer to the questions.

To take a step towards such a model of AA, this paper introduces the notion of a *transfer-of-control strategy*. Previous AA research, when focusing on the transfer-of-control of an individual decision from an agent to another (e.g., a human user), has framed the problem in terms of two basic choices: either transfer control to a human or take autonomous action (i.e., do not transfer control). The notion of a transfer-of-control strategy implies that these are just two of the many transfer-of-control strategies available to an agent. In particular, a transfer-of-control strategy is a planned sequence of transfer-of-control actions, including both those that actually transfer control and those that simply buy more time to get input. The agent executes such a strategy by performing the actions in sequence, transferring control to the specified entity and buying time as required, until some point in time when the entity currently in control exercises that control and makes the decision. For instance, an AH strategy implies that an agent A, initially attempts autonomous actions given a problem. In this case, if it is unable to make the decision it passes control to the human (H) for the decision. Thus, there is uncertainty about which entity will make that decision and when it will do so. The central AA problem is to find a transfer-of-control strategy that maximizes the expected utility (EU) of the decision.

Section 3 presents our model, which we have constructed to capture the factors important to AA reasoning identified in the literature: the relative decision-making quality of the entities, the probability that a given entity will respond in a timely manner, the costs incurred by waiting for a response, and any intermediate actions that the agent can take to limit those costs. Our model uses

AAMAS'02, July 15-19, 2002, Bologna, Italy.

decision-theoretic techniques to determine the EU of a particular transfer of control strategy within a specific domain. We can use the model to explain not only the successes and failures of specific approaches in Electric Elves, but also to other systems, developed and reported on by other researchers.

Our model also supports other key results useful in constructing future AA systems. Most importantly, we prove that no transfer-of-control strategy dominates all others across all application domains. Therefore, AA developers must determine which strategy or strategies are right for their application. To facilitate such a search through the strategy space, we have developed an algorithm that does a systematic search through the strategy space, using our model's ability to evaluate candidate strategy's EU. We empirically demonstrate the validity of this strategy by replacing the existing complex MDP-based approach with the single strategy, and *replicate previously published results*.

2. AA SUCCESSES AND FAILURES

A key reported application of AA is the Electric Elves project[1], which involves personal assistant agents helping with the day-to-day activities of a human organization. Occasionally, the agents need to decide whether and when to transfer control to their users or whether to act. Users are not always available to answer agent queries; hence, the agents require transfer-of-control strategies to handle this contingency. We focus on two specific decisions that need to be made to ensure the smooth running of the organization: whether a user will attend a meeting on time and whether to close an auction for an open role in the organization. Two approaches to AA have been applied in the Electric Elves: one using C4.5 and another using MDPs. We briefly summarize these approaches here, but the full details are available in previous reports[11].

The first implementation used two sets of learned C4.5 rules: one for deciding which action to take, and another set for deciding whether to do a transfer of control[11]. For example, an autonomy rule learned for one user was "if the department head is not at the meeting and it is a Monday, keep control and make a decision". If the autonomy rules decided that the agent should transfer control to a user and the user failed to respond in a fixed amount of time (five minutes), then the agent would take the autonomous action suggested by its first set of rules. A second implementation used MDPs to do both decision making and transfer-of-control reasoning. The MDP explicitly modeled the consequences of asking the user for input and not receiving a prompt response, and it gave the agent much more flexibility in choosing transfer-of-control actions and timing. Despite initially promising behavior, the C4.5 rules made some catastrophic mistakes, including autonomously cancelling a meeting with the division director, while the MDP implementation made no serious errors despite months of real-world use.

Many factors were thought to influence the Electric Elves results. For example, previous reports suggested that the involvement of a *team* was a critical point[11]. The dynamics of the domain and the effect that waiting had on other users clearly played a part as well. However, previous reports could not identify which of the factors was actually the key to the downfall of the C4.5 approach and how the MDP dealt with that factor. Without such an explanation, it is difficult to generalize the conclusions.

Among other practical AA systems, Goodrich et al.[5], in work on tele-operated robots, have looked at the effect of *user neglect* on robot performance. More specifically, they looked at the performance of four control systems (differing in the robot's autonomy) while varying the length of time the user "neglected" the robot. Other work on AA and mixed-initiative planning has focused on detailed comparisons of the EU of user and agent decision making as the sole rationale for deciding who should have control[7, 3]. While intuitive reasons are given for the success of these approaches there is no general theory to support the claims.

3. TRANSFER OF CONTROL STRATEGIES

In this section, we present a model of AA using decision-theoretic techniques. We first present the general framework and then give a specific instantiation which illustrates some of the characteristics of the problem.

We begin with some definitions. An agent, A, is responsible for making a decision, d. There are n entities, $e_1 \ldots e_n$, who can potentially make the decision. These entities can be human users, other agents, or the agent itself. The agent has some mechanism for transferring decision-making control to any of the entities. The expected quality of decisions made by each of the entities, $\mathbf{EQ} = \{EQ^d_{e_i}(t) : \mathcal{R} \rightarrow \mathcal{R}\}_{i=1}^{n}$, is known, though perhaps not exactly. $\mathbf{P} = \{P_\top(t) : \mathcal{R} \rightarrow \mathcal{R}\}$ represent continuous probability distributions over the time that the entity in control will respond with a decision of quality $EQ^d_e(t)$.

Section 2 hinted that delaying a decision is an important consideration for AA. We model the cost of delaying a decision until time t as $\{\mathcal{W} : t \rightarrow R\}$. The set of possible wait-cost functions is $\mathbf{W}$. We assume $\mathcal{W}(t)$ is non-decreasing and that there is some point in time, $\lhd$, when the costs of waiting stop accumulating (i.e., $\forall t \geq \lhd, \forall \mathcal{W} \in \mathbf{W}, \mathcal{W}(t) = \mathcal{W}(\lhd)$).

Finally, the agent has some mechanism by which it can take some action, with cost $\mathcal{D}_{cost}$, with the result of reducing the rate at which wait costs accumulate. We call such an action a *deadline delaying action* and denote it $\mathcal{D}$. For example, a $\mathcal{D}$ action might be as simple as informing the party waiting for the decision that there has been a delay, or more complex, such as reordering tasks. We model the value of the $\mathcal{D}$ by letting $\mathcal{W}$ be a function of $t - \mathcal{D}_{value}$ (rather than t) after the $\mathcal{D}$ action.

We define the set $\mathbf{S}$ to be all possible transfer-of-control strategies available to an agent. The problem for the agent can then be defined as:

Definition 3.1 *For a decision d, the agent must select $s \in \mathbf{S}$ such that $\forall s' \in \mathbf{S}, s' \neq s, EU^d_s t \geq EU^d_{s'} t$*

We define a simple shorthand for referring to particular transfer-of-control strategies by simply writing the order that entities receive control or, alternatively, $\mathcal{D}$s are executed. For example, $a_{fred} a_{barney} \mathcal{D} a_{barney}$ is shorthand for a strategy where the agent gives control to the agent $fred$, then gives it to the agent $barney$, then does a $\mathcal{D}$, and finally gives control back indefinitely to $barney$. Notice that the shorthand does not record the timing of the transfers of control. In the following discussion we assume that the agent itself can always make the decision itself, instantly.

To calculate the EU of an arbitrary strategy, we multiply the probability of response at each instant of time with the expected utility of receiving a response at that instant, and then sum the products. Hence, for an arbitrary continuous probability distribution:

$$EU = \int_0^\infty P_\top(t) EU^d_{e_c}(t) \; .dt \tag{1}$$

where e_c represents the entity currently in decision-making control.

Since we are primarily interested in the effects of delayed response, we can decompose the expected utility of a decision at a certain instant, $EU^d_{e_c}(t)$, into two terms. The first term captures the quality of the decision, independent of delay costs, and the second captures the costs of delay, i.e.,: $EU^d_{e_c} t = EQ^d_e(t) - \mathcal{W}(t)$. A $\mathcal{D}$ action affects the future cost of waiting. For example, the wait

cost after performing a $\mathcal{D}$ at $t = \Delta$ at cost $\mathcal{D}_{cost}$ is : $\mathcal{W}(t|\mathcal{D}) = \mathcal{W}(\Delta) - \mathcal{W}(\Delta - \mathcal{D}_{value}) + \mathcal{W}(t' - \mathcal{D}_{value}) + \mathcal{D}_{cost}$.

To calculate the EU of a strategy, we need to ensure that the probability of response function and the wait-cost calculation reflect the control situation at that point in the strategy. For example, if the user has control at time t, $P_\top(t)$ should reflect the user's probability of responding at t. To do this simply, we can break the integral from Equation 1 into separate terms, with each term representing one segment of the strategy, e.g., for a strategy eA there would be one term for when e has control and another for when A has control.

Using this basic technique, we can now write down the equations for some general transfer-of-control strategies. Equations 3-7 are the general EU equations for the AA strategies A, e, eA and $e\mathcal{D}eA$ respectively. We create the equations by writing down the integral for each of the segments of the strategy, as described above. T is the time when the agent takes control from the user, and Δ is the time at which the $\mathcal{D}$ occurs. One can write down the equations for more complex strategies in the same way.

So far, we have presented general equations for some strategies. To make things more concrete, we instantiate the general model with specific functions. We look specifically at the case where the agent has only one entity to call on (i.e., the user U), the response probability is Markovian, and the wait costs increase exponentially with time until some deadline, $\lhd$. More specifically, for $\mathcal{W}(t)$, we use the following function:

$$\mathcal{W}(t) = \begin{cases} \omega \exp^{\omega t} & t \leq \lhd \\ \omega \exp^{\omega \lhd} & \text{otherwise} \end{cases} \quad (2)$$

and for the probability of response we use: $P_\top(t) = \rho \exp^{-\rho t}$.

The entities' decision-making quality is constant over time, i.e., $EQ^d_A(t) = \alpha$ and for $EQ^d_U(t) = \beta$. For convenience, let $\delta = \rho - \omega$. Table 2 shows the resulting instantiated equations for the strategies in Table 1. Figures 1(a) and (b) show graphically how the EU of the eA strategy varies along different axes. Notice how the EU depends on the transfer time as much as it does on β. Figure 1(d) shows the value of a $\mathcal{D}$ (explained later).

Figure 1(c) compares the EU of the $e\mathcal{D}eA$ and e strategies. Notice that in some parts of the graph strategy $e\mathcal{D}eA$ has higher EU, while in others strategy e has higher EU. In, general, the more complex the transfer-of-control strategy (i.e., the more transfers of control it makes), the flatter the EU graph when plotted against wait cost and response probability. More complex strategies perform relatively worse when probability of response is high and/or the cost of waiting it low. Conversely, more complex strategies perform relatively *much* better when the wait costs are high and the probability of response is low, confirming our contention that complex strategies are in fact useful.

$$EU^d_A = EQ^d_A(0) - \mathcal{W}(0) \quad (3)$$

$$EU^d_e = \int_0^{\lhd} P_\top(t) \times (EQ^d_e(t) - \mathcal{W}(t)).dt + \int_{\lhd}^{\infty} P_\top(t) \times (EQ^d_e(t) - \mathcal{W}(D)).dt \quad (4)$$

$$EU^d_{eA} = \int_0^{T} P_\top(t) \times (EQ^d_e(t) - \mathcal{W}(t).dt) + \int_T^{\infty} P_\top(t).dt \times (EQ^d_a(T) - \mathcal{W}(T)) \quad (5)$$

$$EU^d_{U\mathcal{D}eA} = \int_0^{\Delta} P_\top(t)(EQ^d_e(t) - \mathcal{W}(t)).dt + \int_\Delta^{T} P_\top(t)(EQ^d_e(t) - \mathcal{W}(\Delta) + \mathcal{W}(\Delta - \mathcal{D}_{value}) - \mathcal{W}(t - \mathcal{D}_{value}) - \mathcal{D}_{cost}).dt + \int_T^{\infty} P_\top(t)(EQ^d_A(t) - \mathcal{W}(\Delta) + \mathcal{W}(\Delta - \mathcal{D}_{value}) - \mathcal{W}(T - \mathcal{D}_{value}) - \mathcal{D}_{cost}).dt \quad (6)$$

Table 1: General AA EU equations for simple transfer of control strategies.

4. MANY USEFUL STRATEGIES

The rationale for introducing complex strategies is to give the agent more flexibility which in turn leads to higher EU. However, as we saw in the previous section, it is not always the case that more complex strategies have higher EU. This section presents three Lemmas that show when certain types of strategy are optimal. The Lemmas narrow the field of potentially appropriate strategies for a particular application.

An important assumption underlying transfer-of-control strategies is that several transfers-of-control will sometimes be required. However, Lemma 1 shows that sometimes, even when wait costs continue to accrue, a strategy with a single transfer-of-control might be optimal. In particular, if $\mathcal{W}(t)$ is non-decreasing, $EQ^d_U(t) = \beta$ and $EQ^d_A(t) = \alpha$:

LEMMA 1: *If $s \in \mathbf{S}$ is a strategy ending with U, and s' is sA, then $EU^d_{s'} > EU^d_s$ iff $\forall e \in E, \exists t < \lhd$ such that* $\int_t^{\lhd} P_\top(t')\mathcal{W}(t').dt' - \mathcal{W}(t) > EQ^d_U(t) - EQ^d_A(t)$

Thus, in cases where, even if all wait costs accrue while waiting for a decision, getting a user's input is better than making an autonomous decision, then a strategy that leaves control in the hands of the user indefinitely is better than one that hands over to the agent at some point. Hence, in an application with a single user and agent, we can immediately exclude half the possible strategies.

A $\mathcal{D}$ can be a very useful action since it buys time to increase the chance of getting a user response. One might expect a $\mathcal{D}$ to be useful whenever wait costs accrue reasonably fast, $\mathcal{D}_{cost}$ is not too high, and the user's decisions are far superior to the agent's. However, it turns out to be more subtle than that. We calculate the expected value of a $\mathcal{D}$ by comparing the EU of a strategy with and without the $\mathcal{D}$ (excluding the cost). The $\mathcal{D}$ is useful if and only if its expected value is greater than its cost. Given the specific model instance given in Section 3 we have:

LEMMA 2: *if $s \in \mathbf{S}$ has no $\mathcal{D}$ and s' is s with a $\mathcal{D}$ added then* $EU^d_{s'}t > EU^d_s t$ **iff** $-\delta\omega\rho \exp^{-\delta\Delta}(1 - \exp^{-\omega \mathcal{D}_{value}}) > \mathcal{D}_{cost}$

Figure 1(d) shows that the value of the $\mathcal{D}$ is highest when the probability of response is neither too low or too high. When the probability of response is low, the user is unlikely to respond, even given the extra time, and hence, the agent will have incurred $\mathcal{D}_{cost}$ with no benefit. A $\mathcal{D}$ has low value when the probability of response is high, because the user will likely respond shortly after the $\mathcal{D}$, meaning that it has little effect (the effect of the $\mathcal{D}$ is on the wait costs *afterward*). So, the usefulness of a $\mathcal{D}$ is far more restricted than it would initially seem. While we have shown this

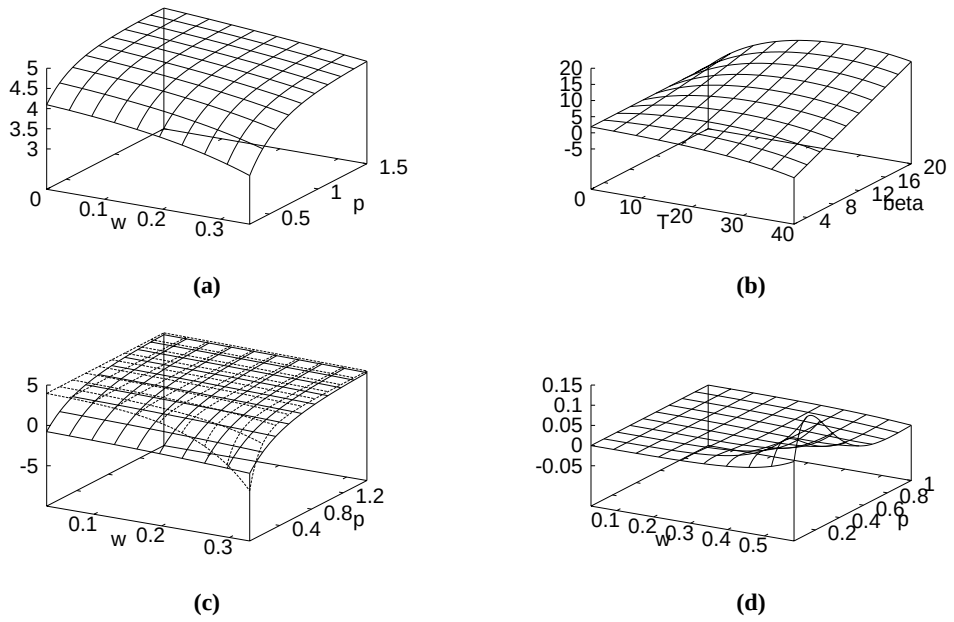

Figure 1: Equation 8 plotted against (a) ω (rate of wait cost increase parameter) and ρ (probability of response parameter) and (b) T (time to transfer control back to agent) and β (quality of user's decision making). (c) Comparing strategies $e\mathcal{D}eA$ and e(dotted line is e). (d) The value of a $\mathcal{D}$.

$$EU_e^d t = \quad \exp^{-\lhd\delta} \times\omega(\tfrac{\rho}{\delta}-1) - \tfrac{\rho\omega}{\delta} + \beta \tag{7}$$

$$EU_{eA}^d t = \tag{8}$$
$$\omega \exp^{-T\delta}(\tfrac{\rho}{\delta}-1) + \exp^{-\rho T}(\alpha-\beta) - \tfrac{\rho\omega}{\delta} + \beta$$

$$EU_{e\mathcal{D}eA}^d t = \tag{9}$$
$$\tfrac{\rho\omega}{\delta}(exp^{-\Delta\delta}-1) + \beta(1-\exp^{-\Delta\rho}) +$$
$$\tfrac{\rho\omega\exp^{-\omega\mathcal{D}_{value}}}{\delta}(\exp^{-T\delta}-\exp^{-\Delta\delta}) +$$
$$(\mathcal{D}_{cost}-\beta)(\exp^{-\rho T}-\exp^{-\rho\Delta}) +$$
$$\omega\exp^{\Delta\omega}(\exp^{-\omega\mathcal{D}_{value}}-1)(\exp^{-\rho\Delta}-\exp^{-\rho T}) -$$
$$\exp^{-\rho T}(\mathcal{D}_{cost}-\alpha+\omega(\exp^{\omega\Delta} -$$
$$\exp^{\omega(\Delta-\mathcal{D}_{value})}+\exp^{\omega(T-\mathcal{D}_{value})}))$$

Table 2: Instantiated AA EU equations for simple transfer of control strategies.

for only a specific instance of the model, it is also true in general. In fact, since the specific model has exponential wait costs, in models where wait costs grow more slowly, the value of the $\mathcal{D}$ is likely to be even less. The value of the $\mathcal{D}$ turns out to depend not only on when it is done, but also on when control is given back to the agent afterward. This is a good illustration of the need for planning transfer-of-control strategies—whether or not a $\mathcal{D}$ is useful at a certain point in time depends heavily on when a later transfer of control will take place.

In cases where one $\mathcal{D}$ is of value, it is reasonable to ask if five $\mathcal{D}$s would be of considerably more value. Usually, this is not the case, for a variety of reasons. In fact:

LEMMA 3: $\forall K \in \mathbf{N}, \exists \mathcal{W}(\mathbf{t}) \in \mathbf{W}, \exists \mathbf{P}_\top(\mathbf{t}) \in \mathbf{P}$, $\exists \mathbf{EQ_e^d}(\mathbf{t}) \in \mathbf{EQ}$ *such that K is the optimal number of $\mathcal{D}$s*

The proof builds on Lemma 2. Consider a situation where the cost of a $\mathcal{D}$ was a function of the number of $\mathcal{D}$s to date. For example, in the Electric Elves' meeting case, the cost of delaying a meeting for the third time is higher than the cost of the first delay. Given that the cost of the Kth $\mathcal{D}$ is $f(K)$, the test for the usefulness of the Kth $\mathcal{D}$ is:

$$f(K) < \tag{10}$$
$$\omega(\exp^{-\mathcal{D}_{value}\omega}-1) \times (\tfrac{p}{\delta}\exp^{-\delta T} - \tfrac{\omega}{\delta}\exp^{-\delta\Delta} - \exp^{\omega\Delta-\rho T})$$

Depending on the nature of $f(K)$, Equation 10 might be made to hold for any number of $\mathcal{D}$s. As with Lemma 2, although the details of the proof are specific to the model, the conclusion is general.

There are other reasons, too, why more $\mathcal{D}$s might not necessarily be better. Figure 1(d) shows that the value of the $\mathcal{D}$ depends on the wait cost. Doing a $\mathcal{D}$ reduces the accruing wait costs, hence making another $\mathcal{D}$ less valuable (unless no decision is made and costs start accruing quickly again.) Notice also that $\mathcal{D}$s become redundant at the time when wait costs stop accruing (i.e,. the deadline), since they no longer provide any benefit after this time.

4.1 Discussion

Lemmas 1-3 show that the relative EU of transfer-of-control strategies depends on factors like the total possible wait cost and the value of a $\mathcal{D}$. We can conclude from this that it is not possible to find a general optimal strategy for all parameters. This is an important result for developers, since it tells them that they need to carefully consider which strategies to use. In the following, we discuss three other conclusions that can be drawn from the model.

Why Plan Transfers-of-Control?

The EU of strategy e might be lower than A even if human decision making is superior, since high wait costs are incurred in strategy e when the user takes time to respond. However, strategy $e\,A$ might be better than both, allowing the agent to safely give control to the user, *provided* it plans ahead and considers complex strategies. In applications where personal preference is an important aspect of a decision, giving the user an opportunity to respond, even if an autonomous decision is eventually made, may increase user acceptance of the system.

How Important is Decision Quality?

Costs due to delays in the user's decision can eventually overwhelm even large differences in the expected quality of agent's and user's decisions. In Figure 1(c) the expected quality of the user's decision is four times that of the agent's, yet, in some situations, strategies where the agent will eventually act still have higher EU to those where the user has control indefinitely. This has interesting implications for AA approaches that focus on detailed calculations of the relative quality of user and agent decision making, e.g., [7, 3]. If even large differences in expected decision-making quality are not sufficient to dictate which strategy is optimal, then perhaps more fine tuning of the decision-making quality calculation may not be critically important.

Why is Timing so Important?

The EU of a strategy is very sensitive to the timing of transfers-of-control. It is relatively easy to understand why the timing of a transfer-of-control back to the agent is important — if too early, the opportunity for a better user decision is lost; if too late, and high costs have already been incurred. The reason for the sensitivity of the $\mathcal{D}$ action timing is slightly more subtle. While the cost of the $\mathcal{D}$ is constant, its value depends on how long after the $\mathcal{D}$ that a decision is made, as well as the wait costs accrued during that period. In particular, if a decision is made straight after a $\mathcal{D}$ there is no value to the $\mathcal{D}$ since there are no saved wait costs (but there is incurred $\mathcal{D}_{cost}$). The model shows that $\mathcal{D}$ actions are most useful when wait costs are accruing sufficiently fast and probability of response is sufficiently low.

5. FINDING OPTIMAL STRATEGIES

The model above shows how to calculate the EU of a strategy but does not provide a method for finding an optimal strategy. In this section we present an algorithm that finds the optimal strategy with up to K transfers of control. The algorithm is a basic branch and bound search that starts with the simplest strategies then adds new segments to create more complex strategies. New strategies are created by appending a transfer of control to another entity or a $\mathcal{D}$ (and control back to the same entity) to an existing strategy (unless the strategy length is $\geq K$). New strategies added as branches of the strategy from which they were created.

In the worst case, the algorithm does an exhaustive search of all possible strategies under length K. However, two simple heuristics are used to exclude many of the strategies from search. Both heuristics will cut a strategy, s, if it or its children can be no better than other strategies that will be checked. In particular, the heuristics exclude strategies where the optimal time for any transfer-of-control is either infinite or the same as the optimal time of another transfer-of-control. In such cases, the optimal timing of the transfers-of-control effectively excludes one of the transfer-of-control actions.

The algorithm was implemented and run over random configurations of entities and wait costs. There were between three and six entities, with one being the agent. The functions used were those shown in the model instantiation in Section 3. The response probability parameter and $EQ_e^d(t)$ of each of the entities was selected at random at the start of each trial, as was the wait cost parameter. Figure 2 shows how the number of strategies checked increases with K and shows the algorithm to be both efficient and scale well as K is increased.

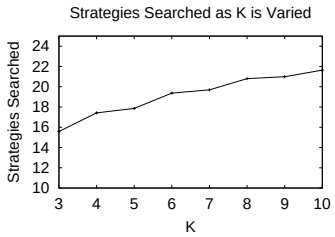

Figure 2: The average number of strategies searched over 1927 trials for different maximum strategy lengths (K).

6. WHY WAS THE ELF AUTONOMOUS?

The instantiated functions given in Section 3 are a coarse approximation of a number of domains, including the Electric Elves. Hence we can use the instantiated model to make predications about the AA reasoning required for Electric Elves. In particular, we focus on the decision whether the user will attend a meeting on time. A key feature of the Electric Elves is that the user is mobile. As she moves around the environment, her probability of responding to requests for decisions changes drastically, e.g., she is most likely to respond when at her workstation. When the agent communicates via a workstation dialog box, the user will respond, on average, in five minutes. However, when the agent communicates via a Palm pilot the average user response time is an hour. Users generally take longer to decide whether they want to present at a research meeting, taking approximately two days on average. So, the function $P_{\top}(t)$ should have an average value of 5 minutes when the user in her office, an average of one hour when the user is contacted via a Palm pilot and an average of two days when the decision is whether to present at a research meeting. We assume that the user's decision-making $EQ_U^d(t)$ is high with respect to the agent's, $EQ_A^d(t)$. When dealing with more important decisions, the cost of the agent's errors is higher and, hence, its decision quality lower. Thus, in some cases, the decision-making quality of the user and the agent will be similar, i.e., $EQ_d^U(t) \approx EQ_d^A(t)$; while in other cases, there will be an order of magnitude difference, i.e., $EQ_d^U(t) \approx 10 * EQ_d^A(t)$. The wait cost function, $\mathcal{W}(t)$, will be much larger when more other users are waiting for the decision to be made. Finally, the cost of delays, i.e., $\mathcal{D}_{cost}$, can vary by about an order of magnitude. For example, the cost of rescheduling meetings varies greatly, e.g., the cost of rescheduling small informal meetings with colleagues is far less than rescheduling a full lecture room at 3 PM Friday.

Above we have shown how the parameters vary from decision to decision. For a specific decision, we can calculate exact EU values for different strategies and draw the following conclusions (Table 6.2 in Section 6.2 presents a quantitative illustration of these predictions):

- The strategy e should not be used, since for all combinations of user location and meeting importance the EU of this strategy is very low.
- Multiple strategies are required, since for different user locations and meeting importance different strategies are optimal.
- Since quite different strategies are required when the user is in different locations, the AA reasoning will need to change strategies when the user changes location.

- No strategy has a reasonable EU for all possible parameter instantiations, hence always using the same strategy will occasionally cause dramatic failures.
- For most decisions, strategies will end with the agent taking a decision, since strategies ending with the user in control generally have very low EU.

These predictions provide important guidance about a successful solution for AA in the E-Elves. In the remainder of this section, we use those predictions to explain the success and failure of the two approaches used in the application.

6.1 Using C4.5 to Learn AA

The initial implementation of AA using C4.5 made some catastrophic mistakes. At the time we identified several reasons for the mistakes including not having enough training data and not taking into account the uncertainty and potential costs involved in an agent's decision. While these reasons are correct the more interesting questions are "Why is so much training data required?" and "Why is uncertainty so important?" With the model we are in a position to answer these more fundamental questions.

The C4.5 implementation learned one strategy and stuck with that strategy, even when the situation's parameters were quite different from those under which it learned the strategy. The model predicts that such an approach would fail and this was precisely what was observed. Enough of the training data fell into a particular an area of the ω-ρ space that the learning focused on the appropriate rule for that area of the space, ignoring other possible strategies. The rule that caused the autonomous cancellation of a meeting with the division director illustrates this point. The rule said that on Mondays strategy A should be followed. This was (likely) due to training data that showed the user was unlikely to respond in a timely manner on Mondays. If the training data also showed the expected quality of agent and user decision making to be reasonably close then the strategy A is the correct strategy. However, in the unusual situation of a meeting with the division director the high cost of error associated with cancelling an important meeting lowers the expected quality of the agents decision. According to the model this means a strategy which gives more opportunity for user response should have been employed but the fixed C4.5 rule could not reason about this and ended up making a serious error.

According to the model predictions, the C4.5 approach of timing out after five minutes and returning control to the agent (either for a $\mathcal{D}$ or a decision on attendance) might be reasonable in some situations, very suboptimal in others. This is another explanation for C4.5's enigmatic behavior.

6.2 MDPs for AA

Figure 3 shows a frequency distribution of the number of actions taken per meeting. The number of actions taken for a meeting corresponds to the length of the strategy followed. The graph shows both that the MDP followed complex strategies in the real world and that it followed *different* strategies at different times. The model predicted this would be required in a successful solution, it was not present in C4.5 which failed and is present here, for the successful solution.

Using the instantiated model from Section 3 and the specific parameters given above we can calculate the EU of various strategies and compare them with the strategies given by the MDP. In its current form, the model cannot directly deal with parameters that change during strategy execution. In order to do a meaningful comparison between the model and the MDP's results, we focus on only those cases when the user's location does not change (i.e., where the probability of response stays constant). The EU values computed by the model and the strategy selected by the MDP are shown in Table 6.2. We ignore any $\mathcal{D}$'s that the MDP performs before the user is asked they are not currently represented in the model. Except for a slight discrepency in the first case the match is exact. Notice, that the strategy e should never be used and is never chosen by the MDP. Thus, despite the model being a considerable simplification of the MDP the correspondence between the results is very close. The particular parameters are not unique in providing a matching model; in fact, many other parameter sets with roughly the same relative values led to matches as well. Moreover, general properties of the policies that were predicted by the model were borne out exactly. In particular, recall that the model predicted different strategies would be required, that strategy e would not be used, and that generally strategies ending in A would be best — all properties of the MDP policies.

Location	A	e	$e\,A$	$e\,\mathcal{D}e\,A$	MDP
Small meeting, active participant					
office	14.8	-277	41.9	42.05	$\mathcal{D}\mathcal{D}e\,\mathcal{D}A$
not @ dept.	14.8	-6E7	31.4	28.0	$\mathcal{D}\mathcal{D}e\,A$
@ meet loc.	14.8	-2E5	39.2	39.1	$e\,A$
Large meeting, passive participant					
office	14.6	-7E12	30.74	30.65	$\mathcal{D}\mathcal{D}e\,A$
not @ dept.	14.6	-2E17	14.6	7.7	$\mathcal{D}\mathcal{D}e\,A$
@ meet loc.	14.5	-7E14	25.1	23.5	$e\,A$

Table 3: EU values for the simple strategies as calculated from the model. The last column shows the strategy actually followed by the MDP.

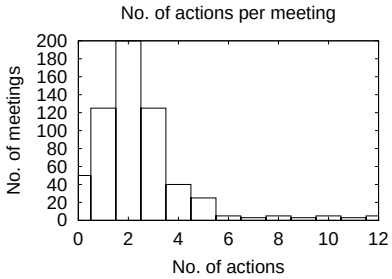

Figure 3: Graph showing the frequency distribution of the number of steps taken in an AA strategy for the meeting scenario.

6.3 Auction Closing MDP

The model predicts that if parameters do not vary greatly from instance to instance of a particular decision then it is sufficient to find a single optimal strategy and simply follow that strategy every time the decision is encountered. The auction closing AA is an instance of this for the Electric Elves. From a team perspective, the behavior of the team when auctioning off an open role, is similiar for every auction, i.e., the wait cost is the same and the pattern of incoming bids is reasonably consistent, hence from the team perspective the probability of response is constant (average response time was two days). Using the algorithm described in Section 5 we determined that the strategy $e\,A$ was optimal and that the transfer-of-control time should vary according to the expected quality of the agent's decision (which is based on the percentage of bids received so far). We replaced the very complex, computationally expenisive MDP code, that originally made the decision, with very simple

code implementing the fixed strategy. Using log files recorded during the actual auctions reported in [11], we experimentally verified that both the MDP and the eA strategy produced the same result. Table 6.3 shows the percentage of available auction time remaining (e.g., if the auction was opened four days before the role should be performed, closing the auction one day before would correspond to 25%) when the MDP version and the eA version of the code closed the auction. The number of bids is used to estimate the agent's expected decision quality. The times at which the MDP would close the auction and the times at which the eA strategy would close the auction are very similar, certainly within a few hours. However, the result is not precisely the same for the MDP and strategy implementations, because the MDP implementation was more reactive to incoming bids.

Date	No. Bids	MDP	eA
7/20/00	9	25%	26%
7/27/00	7	14%	20%
8/3/00	8	29%	23%

Table 4: Auction results. The "MDP" column shows the percentage of available auction time remaining when the MDP closed the auction. The "eA" column shows the percentage of available auction time remaining when the strategy eAwould have closed the auction.

7. EXPLAINING EARLIER WORK

Goodrich et al.[5] report on tele-operated teams of robots, where both the user's high-level reasoning and the robots' low-level skills are required to achieve some task. Within this domain, they have examined the effect of *user neglect* on robot performance. Four control systems were tested on the robot, each giving a different amount of autonomy to the robot, and the performance was measured as user neglect was varied. Figure 4 shows the performance (y-axis) of the four control policies as the amount of user neglect was increased (x-axis). The experiments showed that higher robot autonomy allowed the operator to "neglect" the robot more without as serious an impact on its performance. The idea of user neglect is similar to our idea of entities taking time to make decisions; in this case, if the user "neglects" the robot, the joint task takes longer to perform. Thus, the work can be straightforwardly mapped to our model and the notion of transfer-of-control strategies can be used to qualitatively predict the same behavior as was observed in practice, even though Goodrich et al. did not use the notion of strategies.

The lowest autonomy control policy used by Goodrich et al. was a pure tele-operation one. Since the robot cannot resort to its own decision making, we represent this control policy with a strategy U (where U denotes the user). The second control policy allows the user to specify *waypoints* and on-board intelligence works out the details of getting to the waypoints. Since the robot has no high-level decision-making ability, we again use a strategy U to model its behavior. However, since the coordination between the robot and user is more abstract, the wait cost function is less severe. Also the human is giving less detailed guidance than in the fully tele-operated case (which is not as good according to [5]), hence we use a lower value for the expected quality of the user decision. The next control policy allows the robot to choose its own waypoints given that the user inputs *regions of interest*. The robot can also accept waypoints from the user. The ability for the robot to calculate waypoints is modeled as a $\mathcal{D}$, since it effectively delays the need for the user to provide input. We model this control policy as the strategy $U\mathcal{D}U$. The final control policy is full autonomy, i.e., A. Robot decision making is inferior to that of the user, hence the

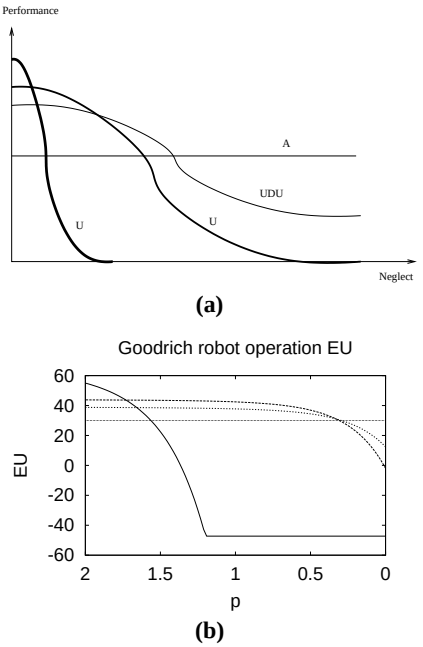

Figure 4: Goodrich at al's various control strategies plotted against neglect. (a) Experimental results. Thinner lines represent control systems with more intelligence and autonomy. (b) Results theoretically derived from model of strategies presented in this article.

robot's decision quality is less than the user's. The graphs of the four strategies, plotted against the probability of response parameter (getting smaller to the right, to match "neglect" in the Goodrich et al graph) is shown in Figure 4. Notice that the shape of the graph theoretically derived from our model (Figure 4(b)) is qualitatively the same as the shape of the experimentally derived graph (Figure 4(a)). Hence, the model qualitatively predicted *from theory* the same performance as was found from experimentation.

A common assumption in earlier AA work has been that if any entity is asked for a decision it will make that decision promptly, hence strategies handling the contingency of a lack of response have not been required. For example, Hortvitz's[7] work using decision theory is aimed at developing general, theoretical models for AA reasoning for a user at a workstation. A prototype system, called LookOut, for helping users manage their calendars has been implemented to test these ideas[7]. In the typical case for LookOut, the agent has three options: to take some action, not to take the action, or to engage in dialog. The central factor influencing the decision is whether the user has a particular goal that the action would aid, i.e., if the user has the goal, then the action is useful, but if he/she does not have the goal, the action is disruptive. Choosing to act or not to act corresponds to pursuing strategy A.[1] Choosing to seek user input corresponds to strategy U. Figure 5(a) shows a graph of the different options plotted against the probability the user has the goal (corresponds to Figure 6 in [7]). The agent's expected decision quality, $EQ^d_A(t)$ is derived from Equation 2 in [7]. (In other words, Lookout performs more detailed calculations of expected decision quality.) Our model then predicts the same selection of strategies as LookOut does, i.e., choosing strategy A when

[1]We consider choosing not to act an autonomous decision, hence categorize it in the same way as autonomous action

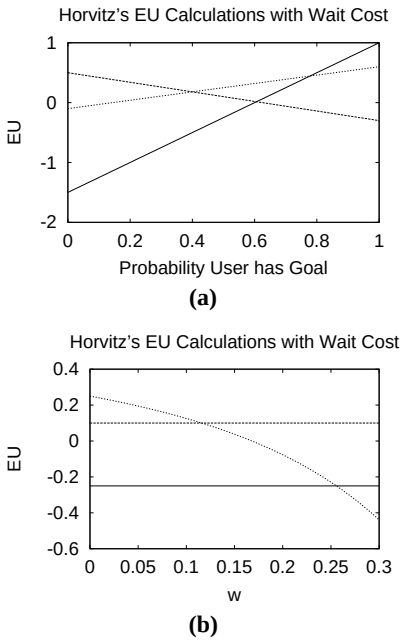

Figure 5: EU of different agent options. The solid (darkest) line shows the EU taking an autonomous action, the dashed (medium dark) line shows the EU of autonomously deciding not to act and the dotted line shows the EU of transferring control to the user. (a) Plotted against the probability of user having goal, no wait cost. (b) plotted against wait cost, fixed probability of user having goal.

$EQ^d_A(t)$ is low, U otherwise (assuming that only those two strategies are available). However, our model further predicts something that LookOut did not consider, i.e., that if the rate at which wait costs accrue becomes non-negligible then the choice is not as simple. Figure 5(b) shows how the EU of the two strategies changes as the rate of wait costs accrue is increased. The fact that the optimal strategy varies with wait cost suggests that LookOut's approach would not immediately be appropriate for a domain where wait costs were non-negligible, e.g., it would need to be modified in many multi-agent settings.

8. SUMMARY AND RELATED WORK

In this paper we have introduced the idea of transfer-of-control strategies for AA. We presented a decision theoretic model of transfer of control strategies which allowed us to explain reported results and can also guide the design of algorithms for AA reasoning. We presented an algorithm which can quickly find optimal strategies. The algorithm, leveraging the model, enabled us to dramatically simplify the complexity of an existing implementation.

Other work has addressed the effect of timeliness on the performance of a system. Hexmoor has looked at how an agent can reason about autonomy when time is limited[6]. Since the agent must select some actions under more time pressure than others, the complexity of the AA reasoning must vary. The agent selects its method of reasoning (ranging from following built-in dispositions, to careful expected-utility analysis) to determine its level of autonomy based on the amount of time available. However, Hexmoor does not address possible transfer of control strategies.

Research in more general AI areas, such as *anytime algorithms*[12], *meta-reasoning*[10], and contingency planning[8], addresses issues similar to those addressed by this work. AA can be viewed in the same framework as meta-reasoning by viewing entities at computations. To our knowledge an analogy of multiple transfers-of-control has not been dealt with in these related areas, primarily because they make different assumptions than are made here.

Acknowledgments

This research was supported by DARPA award no. F30602-98-2-0108. We thank our colleagues, especially, Craig Knoblock, Yolanda Gil, Hans Chalupsky and Tom Russ for collaborating on the Electric Elves project. We would also like to thank Jim Blythe, Gaurav Sukhatme and George Becky for useful comments on the paper.

9. REFERENCES

[1] 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 *Int. Conf. on Innovative Applications of AI*, 2001.

[2] M. Fleming and R. Cohen. Towards a methodology for designing and evaluating mixed-initiative AI systems. In *Proceedings of AAAI Workshop on mixed initiative intelligence*, 1999.

[3] M. Fleming and R. Cohen. A utility-based theory of initiative in mixed initiative systems. In *The IJCAI-01 Workshop on Autonomy, Delegation, and Control: Interacting with Autonomous Agents*, 2001.

[4] Call for Papers. AAAI spring symposium on adjustable autonomy. www.aaai.org, 1999.

[5] M. Goodrich, D. Olsen, J. Crandall, and T. Palmer. Experiments in adjustable autonomy. In *Proceedings of IJCAI Workshop on Autonomy, Delegation and Control: Interacting with Intelligent Agents*, 2001.

[6] H. Hexmoor. A cognitive model of situated autonomy. In *Proceedings of PRICAI-2000, Workshop on Teams with Adjustable Autonomy*, 2000.

[7] Eric Horvitz. Principles of mixed-initiative user interfaces. In *Proceedings of ACM SIGCHI Conference on Human Factors in Computing Systems (CHI'99)*, 1999.

[8] L. Pryor and G. Collins. Planning for contingency: a decision based approach. *J. of Artificial Intelligence Research*, 4:81–120, 1996.

[9] J. Quinlan. *C4.5: Programs for machine learning*. Morgan Kaufmann, 1993.

[10] Stuart J. Russell and Eric Wefald. Principles of metareasoning. In *KR'89: Principles of Knowledge Representation and Reasoning*, Morgan Kaufmann 1989.

[11] P. Scerri, D. Pynadath, and M. Tambe. Adjustable autonomy in real-world multi-agent environments. In *Proceedings of the Fifth Int. Conf. on Autonomous Agents (Agents'01)*, 2001.

[12] S. Zilberstein. Using anytime algorithms in intelligent systems. *AI Magazine*, 17(3):73–83, 1996.