RIAACT: A robust approach to adjustable autonomy for human-multiagent teams

(Short Paper)

Nathan Schurr
Aptima, Inc.
12 Gill St. Suite 1400
Woburn, MA 01801
nschurr@aptima.com

Janusz Marecki
University of Southern California
Los Angeles, CA
marecki@usc.edu

Milind Tambe
University of Southern California
Los Angeles, CA
tambe@usc.edu

ABSTRACT

When human-multiagent teams act in real-time uncertain domains, adjustable autonomy (dynamic transferring of decisions between human and agents) raises three key challenges. First, the human and agents may differ significantly in their worldviews, leading to inconsistencies in their decisions. Second, these human-multiagent teams must operate and plan in real-time with deadlines with uncertain duration of human actions. Thirdly, adjustable autonomy in teams is an inherently distributed and complex problem that cannot be solved optimally and completely online. To address these challenges, our paper presents a solution for Resolving Inconsistencies in Adjustable Autonomy in Continuous Time (RIAACT). RIAACT incorporates models of the resolution of inconsistencies, continuous time planning techniques, and hybrid method to address coordination complexity. These contributions have been realized in a disaster response simulation system.

1. INTRODUCTION

Adjustable autonomy, which is the dynamic transfer of control over decisions between humans and agents [7], is critical in human-multiagent teams. It has been applied in domains ranging from disaster response[8] to multi-robot control [9]. In situations where agents lack the global perspective or general knowledge to attack a problem, or the capability to make key decisions, adjustable autonomy enables agents to access a human participant's superior decisions while ensuring that humans are not bothered for routine decisions.

This paper focuses on *time-critical adjustable autonomy*, which is adjustable autonomy in highly uncertain, deadline-driven domains, where the domain complexity necessarily implies that humans may sometimes provide incorrect input to agents. In such domains, the human may have a global perspective on the problem, but it may be impossible to provide the human with a timely accurate local perspective of individual agents in the team. An example of this is seen when adjustable autonomy was used in disaster response simulations [8]. Incorporating human advice degraded the team performance, at times, and it was shown that an agent team cannot blindly accept or blindly reject human input.

Previous work in adjustable autonomy [7, 10] has failed to address these issues in time-critical domains. Previous work has relied on techniques such as Markov Decision Problem (MDP) and Partially Observable MDP (POMDP) for planning interactions with humans [7, 10]. While successful in domains such as office environments [7], they fail when facing time-critical adjustable autonomy. First, adjustable autonomy planning has, so far, assumed the infallibility of human decisions, whereas these realistic domains demand resolution of inconsistencies between human and agent decisions. Second, previous work has utilized discrete-time planning approaches, which are highly problematic given highly uncertain action durations and deadlines. For example, the task of resolving the inconsistency between a human and an agent takes an uncertain amount of time. Given deadlines, the key challenge is whether at a given time to attempt a resolution. Discrete time planning with coarse-grained time intervals may lead to significantly lower quality in such planning for adjustable autonomy because it may miss a critical opportunity. Planning with very fine grained intervals unfortunately causes a state space explosion, grinding the MDPs/POMDPs down to slow speeds.

We have developed a new approach that addresses these challenges called RIAACT (Resolving Inconsistencies with Adjustable Autonomy in Continuous Time). First, RIAACT extends existing adjustable autonomy policies to overcome inconsistencies between the human and the agents. This allows the agents to avoid a potentially poor input from the human. The aim of this paper is an overarching framework that stands above any particular inconsistency resolution method that is chosen between an agent and a human. RIAACT provides plans that determine how long to allow a human to ponder over a decision, whether to resolve any inconsistency that may arise if the human provides a decision. Secondly, RIAACT leverages recent work in Time-Dependent Markov Decision Problems (TMDPs) [4, 5]. Thus, by exploiting the fastest current TMDP solution technique, we have illustrated the feasibility of applying this TMDP methodology to the adjustable autonomy problem. The result is a continuous time policy that allows for actions to be prescribed at arbitrary points in time, without the state space explosion that results from solving with fixed discrete intervals.

Thirdly, to address the challenge of coordinating the interaction of a team of agents with a human, RIAACT uses a hybrid approach [6], using TMDPs for planning interaction with the human, but relying on non-decision-theoretic approaches (e.g. relying on BDI-logic inspired teamwork), thus significantly reducing the computational burden by not using distributed MDPs. RIAACT's goal is to incorporate these techniques into a practical solution for human-multiagent teams. We illustrate RIAACT's benefits with experiments in a complex disaster response simulation.

Cite as: RIAACT: A robust approach to adjustable autonomy for human-multiagent teams (Short Paper), Nathan Schurr, Janusz Marecki and Milind Tambe, *Proc. of 7th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2008)*, Padgham, Parkes, Müller and Parsons (eds.),May,12-16.,2008,Estoril,Portugal,pp.1429-1432.

2. BACKGROUND AND RELATED WORK

2.1 Adjustable Autonomy

Early work in mixed-initiative and adjustable autonomy interactions suffered from two key limitations: (i) it only allowed for one-shot autonomy decisions that were problematic given uncertain human response times in time-critical domains or (ii) it allowed for sequential transfer of control between humans and agents, but would not scale up to our domains of interest. We elaborate on some of the weaknesses of this prior works.

To remedy that sequential interactions [7, 10, 3, 9] that allow for back-and-forth transfer of control have been proposed. However, these techniques assume that time is discretized, and as a result, to ensure high accuracy decisions, have to deal with large state spaces that this discretization entails. Consequently, these techniques only scale up to tiny domains with time horizons limited to few time ticks — a restriction that is not acceptable for in the Disaster Rescue domain. On the other techniques for planning with continuous time that have been proposed [1, 4, 5] do not discretize time and as such scale up to larger time horizons, but have traditionally not been used in context of human-multiagent teams.

2.2 Time-Dependent Decision Making

Very often, agents that act in real environments have to deal with uncertain durations of their actions. The semi-markovian decision model allows for action durations to be sampled from a given distribution. However, the policy of a semi-markovian decision model is not dependent on time, but only state and as a result, reasoning about deadlines is problematic. To remedy that the Time-dependent Markov Decision Process (TMDP) model was introduced in [1]. The TMDP's solution to handle continuous time is to associate with the discrete state a continuous function of the state value over time. These functions, for different actions executed from the discrete state, can then be compared and an optimal policy for each point in time can be extracted from them.

Recently, there has been a significant progress on solving TMDPs [1, 4, 5]. The primary challenge that any TMDP solver must address is how to perform value iteration over an infinite number of states because the time dimension is continuous. Consequently, each TMDP solution technique must trade off between the algorithm run time and the quality of the solution. We have chosen to utilize the Continuous Phase (CPH) solver [5] as it has been shown to be the fastest of the TMDP solvers available. Thus, the TMDP model matches the requirement posed by adjustable autonomy problems: it allows for back-and-forth transfer of control and returns time dependent policies, yet scales up to realistic domains since it does not discretize time.

3. RIAACT

RIAACT has been designed to address the challenges that arise from this time-critical adjustable autonomy problem. The focus of RIAACT is an overarching framework that will determine adjustable autonomy policy in a time-constrained (deadline) environment where actions take an uncertain amount of time to execute. The planner provides a policy that shows which action to take a distributed team setting. In order to explain this, we will first recall the TMDP model and then show how it can be applied to adjustable autonomy.

The TMDP model [1] is defined as a tuple $\langle S, A, P, D, R\rangle$ where S is a finite set of discrete states and A is a finite set of actions. P is the discrete transition function, i.e., $P(s, a, s')$ is the probability of transitioning to state $s' \in S$ if action $a \in A$ is executed in state $s' \in S$. Furthermore, each tuple $\langle s, a, s'\rangle$ has a corresponding probability density function of action duration $d_{\langle s,a,s'\rangle} \in D$ such that $d_{\langle s,a,s'\rangle}(t)$ is the probability that the execution of action a from state s to state s' took time t. Finally, R is the time-dependent reward function, i.e., $R(s, a, s', t)$ is the reward for transitioning to state s' from state s via action a completed at time t. The optimal policy π^* for a TMDP then maps all discrete states $s \in S$ and times $t \in [0, \Delta]$ to actions $\pi^*(s, t) \in A$ where $[0, \Delta]$ is the desired execution interval.

3.1 Adjustable Autonomy Using TMDPs

In order to address the challenges brought about by dealing with time-critical adjustable autonomy, we model agent policies using TMDP, and achieve coordination across agents by a hybrid approach described later. The RIAACT TMDP model (Figure 1) improves on the previous techniques [7, 10, 3, 9] in two important aspects: (i) it explicitly captures and resolves decision inconsistencies, (ii) it extracts time from the adjustable autonomy problem description, and hence, can take advantage of efficient TMDP algorithms to solve the planning problem at hand. The RIAACT TMDP model is illustrated in Figure 1. Here, single states now have policies that are functions over time. In addition, each arrow in Figure 1 represents not a constant duration, but an entire action duration distribution that can be any arbitrary distribution. Note, that the model in Figure 1 represents a single team decision, and one of these would be instantiated for each team decision in a hybrid approach discussed later.

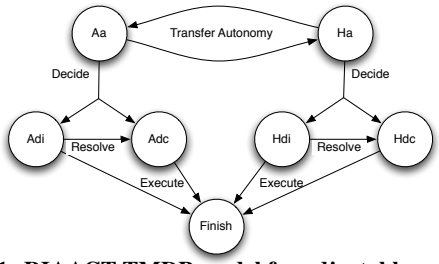


Figure 1: RIAACT TMDP model for adjustable autonomy.

We now describe how RIAACT is represented as a TMDP:

States - Each circle in Figure 1 represents the state that a team decision can be in. To address the challenge of scale while developing an online solution, we have leveraged state abstractions. Each of these state categories can be broken into sub-categories to more accurately model the world, e.g. the state of an inconsistent human decision, Hdi, can be split into several possible inconsistent states, each with their own reward. The RIAACT policy in Figure 1 represents a single team decision in one of the following states: (i) *Agent has autonomy (Aa)* - The agent team has autonomy over the decision. At this point, the agent team can either transfer control to the human or try to make a decision. (ii) *Human has autonomy (Ha)* - Human has the autonomy over the decision. At this point, the human can either transfer control to an agent or make a decision. (iii) *Agent decision inconsistent (Adi)* - This state represents any state in which the agent has made a decision and the human disagrees with that decision. (iv) *Agent decision consistent (Adc)* - This state represents any state in which the agent has made a decision and the human agrees with that decision. (v) *Human decision inconsistent (Hdi)* - This state represents any state in which the human has made a decision and the agent believes that the decision will result in substantial decrease in average reward for the team. (vi) *Human decision consistent (Hdc)* - This state represents any state in which the human has made a decision and the agent believes that the decision will either increase the reward for the team or does not

have enough information to raise an inconsistency about the decision. (vii) *Task finished (Finish)* - This represents the state where the task has been completed and a reward has been earned. The reward can vary based on which decision was executed.

Actions - The arrows in Figure 1 represent the actions that do not take a fixed amount of time — each arrow also has a corresponding function which maps time to probability of completion that action after any time from $[0, \Delta]$. There are four available actions: $Transfer$, $Decide$, $Resolve$, $Execute$. $Transfer$ results in a shift of autonomy between a human and an agent. $Decide$ allows for a decision to be made and results in a transition to either the consistent or inconsistent states (Adc, Adi if agent executed action $Decide$ or Hdc, Hdi if human executed action $Decide$). $Resolve$ is an action that attempts to resolve from an inconsistent state Adi or Hdi to a consistent state Adc or Hdc, which yields higher rewards. To $Execute$ a particular decision results in the implementation of that decision towards the $Finish$ state.

Rewards - The reward for a state is only received if that state is reached before the deadline (time Δ). In previous adjustable autonomy work [8] the decisions made by either party were assumed to have some average quality or reward. In our effort to try and model the diverse perspectives that the agents and humans can have, we extended the model to categorize the decision as either consistent Adc or Hdc or inconsistent Adi or Hdi. It is the case that there can be a wide variety of both consistent and inconsistent team decisions and the model allows for that.

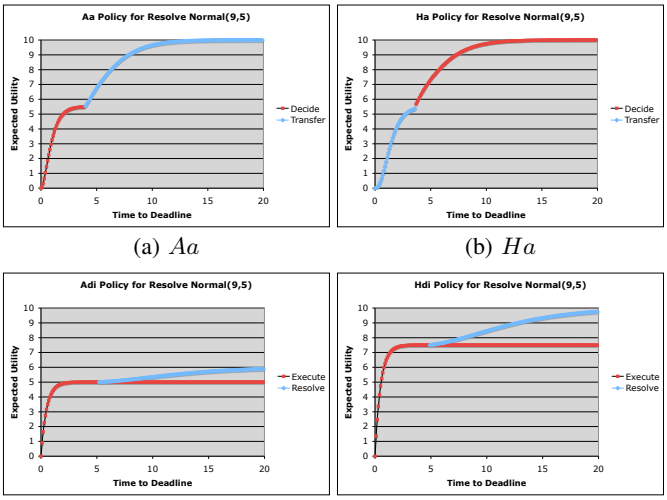
3.2 Hybrid Coordination

In designing our approach for time-critical adjustable autonomy, we treat the RIAACT policy as a team plan, composed of joint actions [2]. Upon generation of the policy, an agent communicates that policy to the rest of the team. The team now has access to the team plan to be executed and the durations of joint actions. This allows us to leverage existing team coordination algorithms such as those based on [2, 8]. For example, suppose that all agents jointly commit to transferring autonomy to the human, and after a certain amount of time a decision is made. If any agent detects an inconsistency, it invokes a joint commitment to the $Resolve$ team action. If an agent detects that this joint commitment is achieved or unachievable (via resolution) then that agent will communicate with the rest of the team. An added benefit of this approach is that multiple agents will not simultaneously commit to resolve, thereby preventing conflicting or redundant resolve team plans. This hybrid approach avoid using computationally expensive distributed MDPs for coordination [6].

4. EXPERIMENTS

We have conducted two sets of experiments to evaluate RIAACT: First, to explore the advantages of its policies over policies returned by previous adjustable autonomy models on a test bed domain and second, to examine RIAACT policies in context of the DEFACTO disaster simulation system [8]. This disaster response scenario includes a human incident commander collaborating with a team of 6 fire engine agents in a large scale disaster with multiple fires. These fires are engulfing the buildings quickly and each have the chance of spreading to adjacent buildings. A decision must be made very quickly about how the team is to divide their limited resources (fire engines) among the fires.

We instantiate the parameters of the RIAACT model as follows: The probability of a consistent decision for the human and the agent is $P(c, H) = P(c, A) = 0.5$. We measure the reward in terms of buildings saved compared to the maximum of 10 building that can catch fire. The reward for an agent decisions is 6 if the decision is consistent with the human and 5 if not, whereas the reward for the human decisions is 6 if the decision is consistent with the agents and 7.5 otherwise. The durations of the $Transfer$ of autonomy action and the $Decide$, $Execute$ and $Resolve$ actions for agents are fast and follow an Exponential distribution with the mean of 0.5. In contrast, the $Decide$ and $Resolve$ actions for the human are slow and follow a Normal Distribution with the mean of 3 and the standard deviation of 1. Throughout all the experiments we focus on the $Resolve$ action as it allows us to demonstrate the unique benefits of RIAACT: resolving inconsistencies and developing a continuous time policy.



(a) Aa (b) Ha

(c) Adi (d) Hdi

Figure 2: RIAACT Model example policy output given that the resolve action duration fits a Normal(9,5).

4.1 Testbed Policy Experiments

For these first experiments, we created a simple testbed domain to construct a policy that included 6 agents, where the $Resolve$ action duration follows a Normal(9,5). The reason for the experiment was to show the benefits in the theoretical model of (i) continuous time, and (ii) the resolve action. The result of the experiment was that each of the benefits are shown and this confirms the usefulness of the RIAACT model in the testbed environment.

Figure 2 shows an example of a policy where the $Resolve$ action duration distribution is a Normal(9,5). The policies for states Adc and Hdc have been omitted from the figure since they show only one action over time to be taken from these consistent decisions, $Execute$. For each general state, the policy shows the optimal action to take and the expected utility of that action as a function over time. Figure 2c and 2d include additional policies, but the policy is the dominant action. On each x-axis is the amount of time left until the deadline and on the y-axis is the expected utility. Thus, if any state is reached, then the time to deadline is referred to and the optimal policy is chosen. For example, if the human has the autonomy (Ha) and the time to deadline is greater than 3.6 seconds, then the optimal action is to attempt a human decision. Otherwise, the optimal action is to transfer that decision over to the agent in order to have the agent make a quicker, but lower average quality decision. Figure 2a shows that the dominant action for the agent has autonomy state, Aa, is to transfer the decision to the human up until 3.9 seconds before the deadline.

Figure 2c and 2d show the times at which the $Resolve$ action is optimal. In order to show the benefit that the $Resolve$ action provides, a new set of experiments was run. The results of this experi-

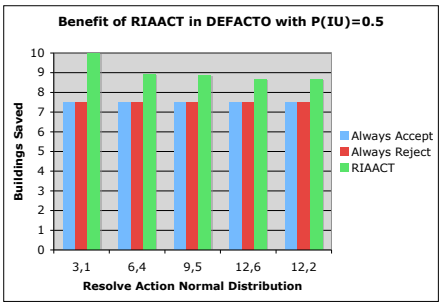


Figure 3: Experiments given a simulated human.

ment can be seen in Figure 2c and 2d. The Execute line represents the policy from previous work, where the inconsistent decision is executed immediately. The Resolve line represents where the policy deviates from $Execute$ if the $Resolve$ action. As seen in both charts, the $Resolve$ action provides a higher expected reward over time. For example, as seen in Figure 2c the policy for Adi is to attempt to resolve an inconsistency if it is detected with at least 14.8 seconds if the $Resolve$.

4.2 DEFACTO Experiments

We have also implemented this in a disaster response simulation system (DEFACTO), which is a complex system that includes several simulators and allows for humans and agents to interact together in real-time [8]. In these experiments, a distributed human-multiagent team works together to try and allocate fire engines to fires in the best way possible. These experiments have been conducted in the DEFACTO simulation in order to test the benefits of the RIAACT policy output. In the scenario that we are using for these experiments, the human had the autonomy and has made a decision. However, this decision is found to be inconsistent (Hdi) and now a RIAACT TMDP policy is computed to determine whether, at this point in continuous time it is beneficial to either $Resolve$ the inconsistency or $Execute$ the inconsistent human decision Hdi.

The experiments included 6 agents and a simulated human. Section 4.1 explained the complete RIAACT policy space for an experimental setting where the $Resolve$ duration was kept as Normal(9,5). In these experiments, we create a different RIAACT policy for each of the following $Resolve$ duration distributions: Normal(3,1), Normal(6,4), Normal(9,5), Normal(12,6), and Normal(12,2). This serves to explore the effects of modeling varying resolve durations and how they effect the policy and eventually the team performance. In each of the experiments, the deadline is the point in time at which fires spread to adjacent buildings and becomes uncontrollable, which in the simulation is 8.7 seconds until deadline.

Using the RIAACT policies, we conducted experiments where DEFACTO was run with a simulated human. A simulated human was used to allow for repeated experiments and to achieve statistical significance in the results. Experiments were conducted comparing the performance of the $Resolve$ action following the RIAACT policy, Always Accept policy or the Always Reject policy (see Figure 3). We assumed the probability that the detected inconsistency was useful, $P(IU) = 0.5$. The $Resolve$ action duration is sampled from the varying normal distributions, shown on the x-axis. These are averaged over 50 experimental runs. The y-axis shows performance in terms of amount of buildings saved. The Always Accept policy is the equivalent of previous work in adjustable autonomy where a decision was assumed to be final, whereas the decision was immediately rejected in the Always Reject policy. The RIAACT policy improves over both of these static policies.

Figure 3 also shows that as the $Resolve$ action duration increases, the benefit gained from using RIAACT decreases. This is due to the approaching deadline and the decreased likelihood that the $Resolve$ will be completed in time. Although, the difference in performance for the Normal(12,2) case may be the smallest, the results show statistical significance $P < 0.05$ ($P = 0.0163$).

5. CONCLUSION

In this paper, we have presented an approach to address the challenges that arise in time-critical adjustable autonomy for human-multiagent teams acting in uncertain, deadline-driven domains, called RIAACT. Our goal is to provide robust solutions for human-multiagent teams in these kinds of environments. Our approach makes three contributions to the field in order to address these challenges. First, our adjustable autonomy framework models resolution of inconsistencies between human and agent view, rather than assuming the human to be infallible. Second, agents plan their interactions in continuous time, avoiding a discretized time model, while remaining efficient. Third, we have created a hybrid approach that combines non-decision-theoretic algorithms for coordination with the decision theoretic planning, to avoid the complexities of the distributed problem. We have conducted experiments that both explore the RIAACT policy space and apply these policies to an urban disaster response simulation. These experiments have shown how can RIAACT can provide improved policies that increase human-multiagent team performance.

6. REFERENCES

[1] J. Boyan and M. Littman. Exact solutions to time-dependent MDPs. In *NIPS*, pages 1026–1032, 2000.

[2] P. R. Cohen and H. J. Levesque. Intention is choice with commitment. *Artificial Intelligence*, 42(2–3):213–261, 1990.

[3] M. A. Goodrich, T. W. McLain, J. D. Anderson, J. Sun, and J. W. Crandall. Managing autonomy in robot teams: observations from four experiments. In *SIGART Conference on Human-Robot Interaction, HRI*, 2007.

[4] L. Li and M. Littman. Lazy approximation for solving continuous finite-horizon MDPs. In *AAAI*, pages 1175–1180, 2005.

[5] J. Marecki, S. Koenig, and M. Tambe. A fast analytical algorithm for solving markov decision processes with real-valued resources. In *IJCAI*, January 2007.

[6] R. Nair and M. Tambe. Hybrid bdi-pomdp framework for multiagent teaming. *Journal of Artificial Intelligence Research (JAIR)*, 23:367–420, 2005.

[7] P. Scerri, D. Pynadath, and M. Tambe. Towards adjustable autonomy for the real world. *Journal of Artificial Intelligence Research*, 17:171–228, 2002.

[8] N. Schurr, P. Patil, F. Pighin, and M. Tambe. Using multiagent teams to improve the training of incident commanders. In *AAMAS '06*, NY, USA, 2006. ACM.

[9] B. P. Sellner, F. Heger, L. Hiatt, R. Simmons, and S. Singh. Coordinated multi-agent teams and sliding autonomy for large-scale assembly. *Proceedings of the IEEE - Special Issue on Multi-Robot Systems*, July 2006.

[10] P. Varakantham, R. Maheswaran, and M. Tambe. Exploiting belief bounds: Practical pomdps for personal assistant agents. In *AAMAS*, 2005.