Asimovian Multiagents: Applying Laws of Robotics to Teams of Humans and Agents

Nathan Schurr[1], Pradeep Varakantham[1], Emma Bowring[1], Milind Tambe[1], and Barbara Grosz[2]

[1] Computer Science Department, University of Southern California
Los Angeles, California
{schurr, varakant, bowring, tambe}@usc.edu
[2] Harvard University, Maxwell-Dworkin Laboratory, Room 249
33 Oxford Street, Cambridge, MA 02138
grosz@eecs.harvard.edu

Abstract. In the March 1942 issue of "Astounding Science Fiction", Isaac Asimov for the first time enumerated his *three laws of robotics*. Decades later, researchers in agents and multiagent systems have begun to examine these laws for providing a useful set of guarantees on deployed agent systems. Motivated by unexpected failures or behavior degradations in complex mixed agent-human teams, this paper for the first time focuses on applying Asimov's first two laws to provide behavioral guarantees in such teams. However, operationalizing these laws in the context of such mixed agent-human teams raises three novel issues. First, while the laws were originally written for interaction of an individual robot and an individual human, clearly, our systems must operate in a team context. Second, key notions in these laws (e.g. causing "harm" to humans) are specified in very abstract terms and must be specified in concrete terms in implemented systems. Third, since removed from science-fiction, agents or humans may not have perfect information about the world, they must act based on these laws despite uncertainty of information. Addressing this uncertainty is a key thrust of this paper, and we illustrate that agents must detect and overcome such states of uncertainty while ensuring adherence to Asimov's laws. We illustrate the results of two different domains that each have different approaches to operationalizing Asimov's laws.

1 Introduction

Recent progress in the agents arena is bringing us closer to the reality of multiagent teams and humans working together in large-scale applications [11, 10, 12, 3, 4]. In deploying such multiagent teams and making them acceptable to human teammates, it is crucial to provide the right set of guarantees about their behavior. The unanswered question is then understanding the right set of guarantees to provide in such teams.

In this paper, we focus on Asimov's three laws of robotics from his science-fiction stories that provide us a starting point for such behavior guarantees. We do not claim that these laws are the only or best collection of similar rules. However, the laws outline some of the most fundamental guarantees for agent behaviors, given their emphasis on ensuring that *no harm* comes to humans, on obeying human users, and ensuring protection of an agent. Indeed, these laws have inspired a great deal of work in agents and multiagent systems already [14, 4, 8]. However, in operationalizing these laws in the context of multiagent teams, three novel issues arise. First, the key notions in these laws (e.g. "no harm" to humans) are specified in very abstract terms and must be specified in concrete terms in implemented systems. Second, while the laws were originally written for interaction of an individual robot and an individual human, clearly, our systems must operate in a team context. Third, since, in many realistic domains, agents or humans may not have perfect information about the world, they must act based on these laws despite information uncertainty and must overcome their mutual information mismatch.

Indeed, as mentioned earlier, researchers have in the past advocated the use of such laws to provide guarantees in agent systems [14, 4, 8]. However, previous work only focused on a single law (the first law of safety) and in the process addressed two of the issues mentioned above: defining the notion of harm to humans and applying the laws to teams rather than individual agents. The key novelty of our work is going beyond previous work to consider the second of Asimov's laws, and more importantly in recognizing the fundamental role that uncertainty plays in any faithful implementation of such a law. In particular, Asimov's second law addresses situations where an agent or agent team may or may not obey human orders — it specifies that in situations where (inadvertant) harm may come to other humans, agents may disobey an order. However, in the presence of uncertainty faced either by the agents or the human user about each others' state or state of the world, either the set of agents or the human may not be completely certain of their inferences regarding potential harm to humans. The paper illustrates that in the presence of such uncertainty, agents must strive to gather additional information or provide additional information. Given that the information reduces the uncertainty, agents may only then disobey human orders to avoid harm.

To the best of our knowledge, this paper for the first time provides concrete implementations that address the three key issues outlined above in operationalizing Asimov's laws. Our implementations are focused on two diverse domains, and thus require distinct approaches in addressing these issues. The first domain is that of disaster rescue simulations. Here a human user provides inputs to a team of (semi-)autonomous fire-engies in order to extinguish maximum numbers of fires and minimize damage to property. The real-time nature of this domain precludes use of computationally expensive decision-theoretic techniques, and instead agents rely on heuristic techniques to recognize situations that may (with some probability) cause harm to humans. The second domain is that of a team of software personal assistant deployed in an office environment to assist human users to complete tasks on time. The personal assistants face significant uncertainty since observations about the human users. Here, we use partially observable markov decision problems (POMDPs) to address such uncertainty.

2 Human-Multiagent Systems

Increasingly, agents and agent teams are being viewed as assistants to humans in many critical activities, changing the way things are done at home, at office, or in a large organization. For example, as illustrated in [11], multiagent teams can help coordinate teams of fire fighters in rescue operations during disaster response. Furthermore, they are also being used as helping hand to humans in an office setting for assisting in various activities like scheduling meetings, collecting information, managing projects etc. Such a transformation seems a necessity, because it relieves humans of the routine and mundane tasks and allows them to be more productive and/or successful.

However, making such a transformation introduces many new challenges concerning how the human and agents will interact. The primary challenge that we focus on in this paper is that if humans are to trust agents with important tasks, they are going to want some guarantees on the performance of the agents. This allows the humans to be confident that problematic or dangerous situations won't arise for the humans.

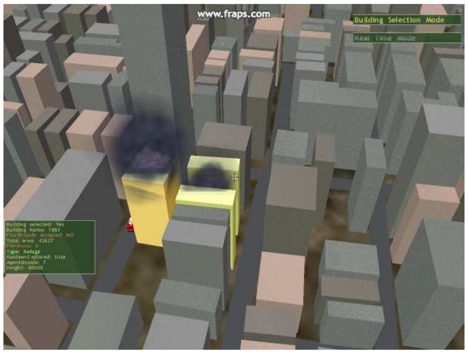
Below, we will introduce two domains that have to address the challenges of including both humans and agents in real world situations. First, we will describe a disaster response simulation where a human user must help a team of fire fighter agents put out all the fires in a downtown area. Second, we present a domain where agents assist workers with assigning of duties in an office environment.

A key aspect of both domains is "adjustable autonomy" which refers to an agent's ability to dynamically change its own autonomy, possibly to transfer control over a decision to a human. Adjustable autonomy makes use of flexible transfer-of-control strategies [9]. A transfer-of-control strategy is a preplanned sequence of actions to transfer control over a decision among multiple entities. For example, an AH strategy implies that an agent (A) attempts a decision and if the agent fails in the decision then the control over the decision is passed to a human (H). An optimal transfer-of-control strategy optimally balances the risks of not getting a high quality decision against the risk of costs incurred due to a delay in getting that decision.

Both of these constructed systems were described in earlier publications [11, 13] and had noted the problematic situations when interacting with humans. However, the diagnosis of such situations and their particular solutions are novel contributions. Because of their differing characteristics, the domains arrive at different approaches to their solutions.

2.1 Disaster Response

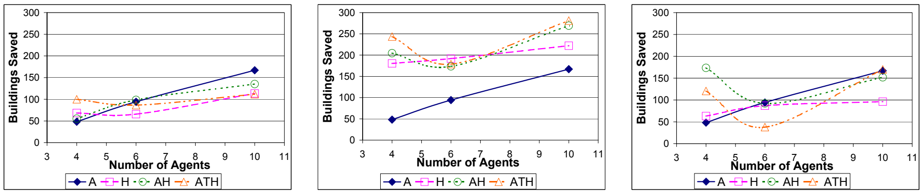
Techniques for augmenting the automation of routine coordination are rapidly reaching a level of effectiveness where they can simulate realistic coordination on



Building selected: Yes
Building Name: 1861
Fire Brigade Assigned: NO
Total area: 43627
Fieryness: 0
Type: Refuge
HasBeenExplored: true
AgentsInside: 7
Height: 40000

Fig. 1. The DEFACTO system displays multiple fires in an urban environment.

the ground for large numbers of emergency response entities (e.g. fire engines, police cars) for the sake of training. Furthermore, it seems inevitable that future disaster response systems will utilize such technology for coordination among different rescue vehicles. We have constructed DEFACTO (Demonstrating Effective Flexible Agent Coordination of Teams through Omnipresence) as a high fidelity system for training and simulating of future disaster response. DEFACTO allows for a human user (fire fighter) to observe a number of fires burning in buildings in an urban environment, and the human user is also allowed to help assign available fire engines to the fires. The DEFACTO system achieves this via three main components: (i) Omnipresent Viewer - intuitive interface (see Figure 1), (ii) Proxy Framework - for team coordination, and (iii) Flexible Interaction - adjustable autonomy between the human user (fire fighter) and the team. More about DEFACTO can be found here [11].



(a) Subject 1

(b) Subject 2

(c) Subject 3

Fig. 2. Performance.

The DEFACTO system's effectiveness was evaluated through experiments comparing the effectiveness of adjustable autonomy strategies over multiple users. In DEFACTO, each fire engine is controlled by a "proxy" agent [11] in order to handle the coordination and execution of adjustable autonomy strategies. Consequently, the proxy agents can try to allocate fire engines to fires in a distributed manner, but can also transfer control to the more capable human user. The user can then allocate engines to the fires that the user has control of.

The results of our experiments are shown in Figure 2, which shows the results of subjects 1, 2, and 3. Each subject was confronted with the task of aiding fire engines in saving a city hit by a disaster. For each subject, we tested three strategies, specifically, H, AH and A_TH; their performance was compared with the completely autonomous A strategy. An H strategy implies that agents completely rely on human inputs for all their decisions. An A strategy is one where agents act with complete autonomy and allocate themselves to task (fires) without human assistance. An AH strategy allows the agent to possibly allocate itself to a task, and if not, then transfer control to the human, whereas the similar A_TH strategy allows the whole agent team to try and allocate a task amongst the team before it will resort to transferring control to a human. Each experiment was conducted with the same initial locations of fires and building damage. For each strategy we tested, we varied the number of fire engines between 4, 6 and 10. Each chart in Figure 2 shows the varying number of fire engines on the x-axis, and the team performance in terms of numbers of building saved on the y-axis. For instance, subject 2 with strategy AH saves 200 building with 4 agents. Each data point on the graph is an average of three runs. Note that the phenomena described below ranges over multiple users, multiple runs, and multiple strategies.

Figure 2 enables us to conclude that: *Following human orders can lead to degradation in agent team performance.* Contrary to expectations and prior results, human involvement does not uniformly improve team performance, as seen by human-involving strategies performing worse than the A strategy in some cases. For instance, for subject 3, AH strategy provides higher team performance than A for 4 agents, yet at 6 agents human influence is clearly not beneficial (AH performs worse than A). Furthermore, for subject 1, following human orders leads to lower performance with 10 agents, with AH or A_TH, than with a fully autonomous strategy (A). We also note that the strategies including the humans and agents (AH and A_TH) for 6 agents show a noticeable decrease in performance for subjects 2 and 3 (see Figure 2) when compared to 4 agents. Since the performance of the fully autonomous strategy increases along with the increasing number of agents, we conclude that the culprit is the agents following human orders in AH and A_TH. It is very important to have the team understand which factors contributed to this phenomena and to have the team be able to prevent it.

2.2 Office Assistants

Another domain that we consider is the Task Management Problem (TMP) in personal software assistants. This is a problem that we are currently addressing as part of CALO (Cognitive Agent that Learns and Organises), a software personal assistant project [5]. In this domain, a set of dependent tasks is to be performed by a group of users before a deadline. An example could be one where a group of users are working on getting a paper done before the deadline. Each user is provided with an agent assistant. Each agent monitors the progress of its user on various tasks, and helps in finishing the tasks before a deadline by doing task reallocations (in case of insufficient progress) at appropriate points in time. Agents also make a decision on whom to reallocate a task, thus having to monitor status of other users who are capable of doing it. More details of TMP are in [13].

This problem is complicated as the agents need to reason about reallocation in the presence of transitional and observational uncertainty. Transitional uncertainty arises because there is non-determinism in the way users make progress. For example, a user might finish two units of a task in one time unit, or might not do anything in one time unit (a task here is considered as a certain number of units of work). Observational uncertainty comes about because it is difficult to observe exact progress of a user or the user's capability level.

Agents can ask their users about the progress made (when there is significant uncertainty about the state) or for decision on re-allocation of the current task. This asking, however, comes at a cost of disturbing the user and occurs only with a certain probability as users may or may not respond to agents request. Thus each agent needs to find an optimal strategy that guides its operation at each time step, till the deadline.

Partially Observable Markov Decision Problems (POMDPs) were used in modeling this TMP problem, owing to the presence of uncertainty. Policy in a POMDP is a mapping from "belief states" (probability distribution over the states in the system) to actions. Each user's agent assistant computes such a policy. (More information on POMDPs can be found in [6]. Though this paper does not require an in-depth understanding of POMDPs, high level familiarity with POMDPs is assumed.)

Unfortunately, within TMP an agent faithfully following orders may result in a low quality solution (low expected utility). There are scenarios where human can provide a decision input at a certain point in time, but later the agent team runs into problems because of faithfully following that decision input. For example, the human can ask the agent not to reallocate the task because of a belief that the task can be finished on time. Yet, since the human is in control of many tasks, she may not be able to finish the task on time. Also, agent faces significant uncertainty about certain factors in the domain, and hence is not in a situation to override human decisions. For example: agent can have significant uncertainty about the progress on a task, due to the transitional and observational uncertainty in the domain. The result is that:

1. *Faithfully following human decision may lead to problems*: This is because humans are in control of many tasks, and depending on the workload over time, some human decisions might need to be corrected over time. Since humans may not provide such corrections, agents may need to overide the human's earlier decision.
2. *Agents need to reduce uncertainty about certain variables*: While agents must occasionally override human decisions, they face uncertainty in estimating human capability (amount of progress accomplishable in one time unit) and

information about progress. If the agent assumes a certain capability level (from previous experiences) and plans accordingly without considering human input, it might run into problems: (a) if it assumes a lower capability level, then the reallocation will happen early and (b) if a higher value is assumed, it wouldn't reallocate until very late, making it difficult for the user taking this task. Similarly, an agent faces uncertainty about actual progress on a task.

3 On Asimov's Laws

In 1942, having never had any personal contact with a robot, Isaac Asimov sat down to write his short story "Runaround" [2] and in doing so enumerated for the first time his three laws of robotics:

– *First Law: A robot may not injure a human being, or, through inaction, allow a human being to come to harm.*
– *Second Law: A robot must obey the orders given it by human beings except where such orders would conflict with the First Law.*
– *Third Law: A robot must protect its own existence as long as such protection does not conflict with the First or Second Law.*

Asimov believed these three laws were both necessary and sufficient, an idea he set out to illustrate in his series of robot stories. While he believed that correctly implemented his three laws would prevent robots from becoming the nightmarish Frankensteins that were the fodder of many science fiction stories, even Asimov admitted that the operationalization of his three laws would not be simple or unambiguous. In this paper, we focus on operationalization of the first two laws, which requires several key issues be addressed in concretely applying them to our domains of interest: (i) Providing definition of "harm" so central to the first law; (ii) Applying these laws in the context of teams of agents rather than individuals; and (iii) Addressing these laws in the presence of uncertainty in both the agent's and the human user's information about each other and about the world state. Previous work has only focused on the first law, and thus on techniques to avoid harm via agents' actions [4, 8, 14]. This previous work dealt with both a single agent and a team of agents, but the emphasis remained on the autonomous actions of these agents. In contrast, the second law emphasizes interactions with humans, and thus its relevance in the context of heterogeneous systems that involve both humans and multiagent teams, that are of interest in this paper.

Indeed, among the issues that must be addressed in concretely applying these laws, the first two — defining harm and applying the laws to teams instead of individuals — are addressed in previous work (albeit differently from our work). However, it is uncertainty of information that the agents and the human user may suffer from, that is the novel issue that must be clearly addressed when we deal with the second law. In the following, we provide a more detailed discussion of these three issues, with an emphasis on the issue of uncertainty. Nonetheless, in contrast with previous work, this paper is the first (to the best of our knowledge) that addresses these three issues together in operationalizing the two laws.

3.1 Definition of Harm

What constitutes harm to a human being? Must a robot obey orders given it by a child, by a madman, by a malevolent human being? Must a robot give up its own expensive and useful existence to prevent a trivial harm to an unimportant human being? What is trivial and what is unimportant? pg 455 [2]

The notion of harm is fundamental to Asimov's laws. Yet, Asimov himself did not imply that harm to be necessarily physical harm to huamns. Indeed, in the story "LIAR" harm is purely mental harm (e.g. someone not getting a promotion they wanted) [2]. So whereas the notion of harm as physical harm to humans is obviously relevant in one of our domains mentioned earlier (disaster rescue), it is also relevant in the office assistant domain, where harm may imply harm to some business (e.g. products not delivered on time) where the office assistant team is deployed. Indeed, in previous work in software personal assistants that is motivated by Asimov's laws [14, 8], the notion of harm includes such effects as deletion of files or meeting cancellation.

In this paper, the notion of harm is operationalized as a "significant" negative loss in utility. So if actions cause a significant reduction in an individual agent's or team's utility, then that is considered as constituting harm. In our disaster rescue simulation domain, such negative utility accrues from loss of (simulated) human life or property. An example of this can be see when subject 3's inputs are followed by 6 fire engine agents, resulting in more buildings being burned than if the inputs were ignored. In our office assistant domain, the lack of the agent team's ability to complete tasks by deadlines provided is what constitutes harm.

3.2 Applying Laws to Teams

I have dealt entirely with the matter of the interaction between [a] single robot and various human beings. ... Suppose two robots are involved, and that one of them, through inadvertance, lack of knowledge, or special circumstances, is engaged in a course of action (quite innocently) that will clearly injure a human being – and suppose the second robot, with greater knowegde or insight, is aware of this. pg. 479-480 [2]

Diana Gordon-Spear's work on *Asimovian agents*[4] addresses teams of agents that guarantee certain safety properties (inspired by the first law above) despite adaptation or learning on part of the agent team. The key complications arise because the actions of multiple agents interact, and thus in preserving such safety property, it is not just the actions of the individual, but their interactions that must be accounted for, in terms of safety. In our work (particularly as seen in the disaster response domain of Sections 2.1 and 4.1), similar complexities arise when applying the laws to teams of agents. No single individual may be able to detect harm by itself; rather the harm may only be detectable when the team of agents is considered as a whole.

3.3 Uncertainty

Even a robot may unwittingly harm a human being, and even a robot may not be fast enough to get to the scene of action in time or skilled enough to take the necessary action. pg 460 [2]

The second law in essence requires that agents obey human orders unless such orders cause harm to (other) humans. Thus, this law opens up the possibility that the agent may disobey an order from a human user, due to the potential for harm. In many previous mixed agent-human systems including our own systems described in Section 2, human inputs are considered final, and the agent cannot override such inputs — potentially with dangerous consequences as shown earlier. Asimov's second law anticipates situations where agents must indeed override such inputs and thus provides a us a key insight to improve the performance of agents and agent teams.

Yet, the key issue is that both the agents and the human users have uncertainty; simply disobeying an order from a human given such uncertainty may be highly problematic. For example, the agents may be uncertain about the information state of the humans, the intellectual or physical capability of the human users, and the state of the world, etc. In such situations, agents may be uncertain about whether the current user order may truly cause (inadvertant) harm to others. It is also feasible that the human user may have given an order under a fog of uncertainty about the true world state that the agent is aware of; and in such situations, the agents' inferences about harmful effects may be accurate.

The key insight in this paper then relates to addressing situations under the second law where an agent may disobey human orders due to its potential for causing harm to other humans: given the uncertainty described above, an agent should not arbitrarily disobey orders from humans, but must first address its own or the human users' uncertainty. Its only upon resolution of such uncertainty that the agent may then disobey an order if it causes harm to others.

The key technical innovation then is recognizing situations where an agent or the human user faces significant uncertainty, and taking actions to resolve such uncertainty. When addressing domains such as the office assistant, agents bring to bear POMDPs. Given this POMDP framework, when an agent is faced with an order from humans with the potential for harm (significant reduction in individual or team utility) it must consider two particular sources of uncertainty before obeying or disobeying such an order: (i) uncertainty about actual user progress on a task and (ii) uncertainty about user capability to perform a task(i.e.- user's rate of performing the task).

If the above two uncertainties are resolved, and the expected utility computations of the POMDP illustrate that the human order leads to reduction in individual or team utility, then this is the case (by Asimov's second law) where an agent may ultimately disobey human orders.

In addressing the simulated disaster rescue domain, the issue centers on potential uncertainty that a human user must face. Here, an agent team acting in the simulated disaster-rescue environment potentially has more certainty about the world state than the human user does. Unfortunately, the disaster is spreading rapidly, and unlike the office environment, the agent team may have little time to deliberate on the uncertain situation, and come up with a detailed policy (as with a POMDP). Instead, the agent team quickly brings up to the notice of the human user key possible sources of potential harm due to the human's order. At this juncture, the human user may be able to reduce his/her uncertainty about the world state, and thus possibly rectify his/her order.

4 Operationalizing Asimov's Laws

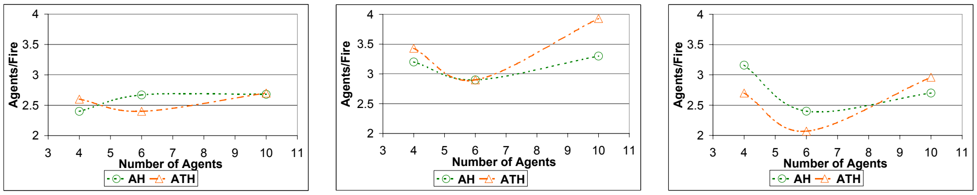
In each of our two domains, appropriately obeying the first and second laws would have improved the situation. Specifically in the second law, the caveat where human directives should be followed, *unless it causes harm to humans*, is not being paid attention to. Instead, as mentioned in Section 2, agents blindly obey human commands, which is problematic. However, as mentioned earlier, in complex domains, there is significant uncertainty. Given such uncertainty, it is quite feasible for the humans to provide imperfect inputs or orders; yet agents may not be certain that these orders are in error, due to the uncertainty that they face. Our position is that in order to start constructing teams of agents and humans that perform well, they must not always take human input as final, yet must only do so after resolving uncertainty.

Given that humans may (unintentionally) provide problematic input, we propose that there are 5 general categories of agents' reactions to problematic human input. In particular, agents may:

- A. Follow the human input exactly
- B. Ignore the human input completely
- C. Make the human aware of the alleged problem in the input
- D. Make the human aware of the alleged problem in the input and offer non-problematic option(s)
- E. Limit human input to only pre-defined non-problematic options to be chosen from by the human

Due to the uncertainty (mentioned in Section 3.3), Option A and B become infeasible. Option A results in suboptimal performance and does not take advantage of the team's resources and potential as seen in Section 2. Option B may end up in better performance, but not only results in angry or confused humans, the agents may also be mistaken due to its own uncertainty and poor performance. Option E is not very practical (too many options to explore) for dynamic domains, or worse, it is impossible to elicit all options. It is then desirable to engage in some of the dialogue described in Options C or D. Our aim is to have joint performance of the agents and humans be better than either of them separately, that is to have the agents correct problems in humans and vice versa.

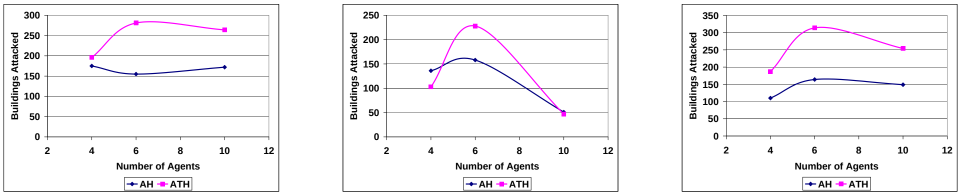
4.1 Disaster Response



(a) Subject 1 (b) Subject 2 (c) Subject 3

Fig. 3. Amount of agents assigned per fire.

Our goal was to have the agent team be able to detect the problematic input seen in the previous experiments and then be able to engage in some type of dialogue with the human user. In order to do this, we continued an in depth analysis of what exactly was causing the degrading performance when 6 agents were at the disposal of the human user. Figure 3 shows the number agents on the x-axis and the average amount of fire engines allocated to each fire on the y-axis. AH and A_TH for 6 agents result in significantly less average fire engines per task (fire) and therefore lower average. For example, as seen in Figure 3, for the A_TH strategy, subject 3 averaged 2.7 agents assigned to each fire when 4 agents were available, whereas roughly 2.0 agents were assigned to each fire when 6 agents were available. It seems counterintuitive that when given more agents, the average amount that were assigned to each fire actually went down. Another interesting thing that we found was that this lower average was not due to the fact that the human user was overwhelmed and making less decisions (allocations). Figures 4(a), 4(b), and 4(c) all show how the number of buildings attacked do not go down in the case of 6 agents, where poor performance is seen.



(a) Subject 1 (b) Subject 2 (c) Subject 3

Fig. 4. Number of buildings attacked.

We can conclude from this analysis that the degradation in performance occurred at 6 agents because fire engine teams were split up, leading to fewer fire-engines being allocated per building on average. Indeed, leaving fewer than 3 fire engines per fire leads to a significant reduction in fire extinguishing capability. Given this, we implemented the ability for the agent team to detect if a reallocation is pulling a teammate from a working group of 3 or more. Once this is detected, there is a high probability that the team performance will be degraded by following the human input. But since there is some uncertainty in the final outcome, the agents do not blindly follow (Option A from above) or ignore (Option B from above). Instead they present the possible problem to the human (Option C from above).

Reject Orders to Split?	Buildings Damaged	Fires Extinguished
No	27	3
No	29	3
No	33	1
Yes	14	5
Yes	18	5
Yes	20	5

Table 1. Benefits to team when rejecting orders allows split of team. In top half, team accepted all human orders, and in bottom half, problematic orders were rejected.

In order to evaluate this implemenation we set up some initial experiments to determine the possible benefits of the team being able to reject the splitting of a coordinated subgroup. We used a team comprised of a human user and 6 agents. We used the same map and the same A_TH strategy as were used in previous experiments. Each of these results were from a short (50 time step) run of the DEFACTO system. The only variable was whether we allowed the agents to raise objections when given allocation orders to split up. In these experiments, the human user listened to all agent objections and did not overide them. The results of this initial experiment can be seen in Table 1. In Table 1, we present results for three problem instances. Performance is measured by calculating the amount of buildings damaged (less is better) and the number of fires extinguished (more is better). As seen from the number of buildings damaged, by allowing the agents to reject some of the human input (see bottom half of Table 1), they were able to more easily contain the fire and not allow it to spread to more buildings.

4.2 Office Assistants

In the TMP domain, the state space of the POMDP consists of a triple of three variables: {Progress on task, Capability level associated with the user, Time till deadline}. As mentioned earlier, of these three variables, an agent may be uncertain about the progress level and the capability level. Actions for the agent can be {$Wait$, *Reallocate*, *AskDecision*, *AskProgress*}. $Wait$ is for the agent to do waiting while user makes progress on the task, while *Reallocate* is for the agent to reallocate the task to a different user. *AskDecision* causes the agent to ask the user whether to reallocate, while *AskProgress* is for removing uncertainty about progress by gathering more information. *AskDecision* also leads to reduction of uncertainty in a user's capability level – an agent invokes *AskDecision* when, according to its estimate of user capability, a task should be reallocated. A user's agreement or disagreement to reallocate provides an opportunity to correct this estimate. However, an agent will disobey this order from a user at a later time if this estimate is corrected.

When an agent executes *AskDecision* to ask whether to reallocate, then, based on the user response, an agent addresses its uncertainty in user capability level as follows. There are three cases of user response to consider:

1. Reallocate: In this instance, the agent had concluded based on its estimate of the user's capability to reallocate. Since the user concurs, the agent's POMDP policy dictates that the task be reallocated to the appropriate user, and the policy terminates.
2. Don't Reallocate: Here the user has disagreed with the agent. Assuming a maximum error of ϵ in its estimation of the user's capability level, this response from the user makes the agent use a capability level increased by ϵ for the future time points. Consequently, the agent resolves its uncertainty in user capability level. If, at a later time, even with this increased user capability, the agent estimates that the user will be unable to fulfill the task, it will then reallocate, thereby overriding prior human input.
3. No Response: Equivalent to a wait action, with cost incurred for asking the user.

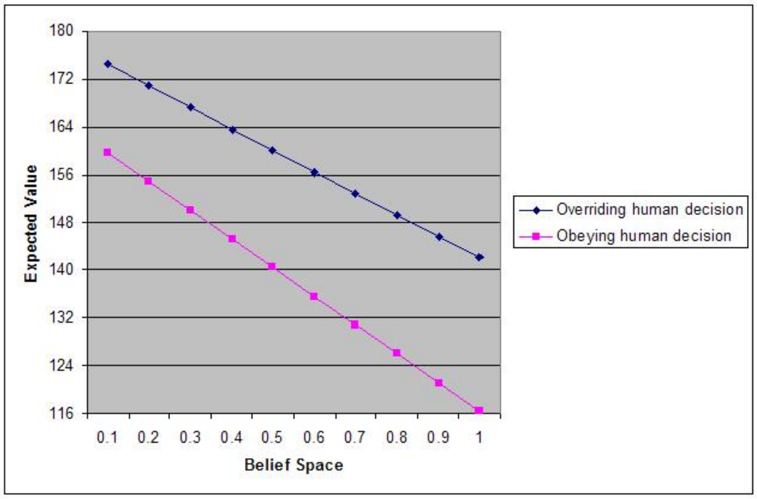


Fig. 5. Comparison of expected values of the two strategies

Figure 5 compares the expected value of two policies for the TMP problem mentioned in Section 2.2. The first policy is one where agents always obeyed human decisions, e.g. if the human gave an order that a task should not be reallocated, the agent absolutely never reallocated the task. In the second policy, the agents sometimes override human orders. In particular, if the human user ordered to *not reallocate*, then the agent initially obeyed the human order. While obeying this order, the agent also increased its estimate of human capability by maximum allowable amount (since the user disagreed with the agent's estimate of reallocation), thus reducing potential uncertainty about human user capability. However, subsequently, the policy reallocated the task if the user was seen to be unable to finish the task on time, even though earlier the user had given an order to not reallocate. This is because the agent had now reduced its uncertainty about human capability, and was now certain that disobeying this user order will avoid harm to the team. In Figure 5, the y-axis plots the value of the two policies mentioned above: obeying human decision vs overriding human decision. The x-axis plots different belief states (probability distribution over world states). We see that the policy to (sometimes) override achieves higher expected value than the policy to always obey human decisions.

5 Related Work and Conclusion

Many projects that deal with agent interactions with humans have started to worry about the safety of those humans. Consequently, they have started to look to Asimov's laws of robotics for some crucial guarantees. This past work ([14, 4, 8]) has focused on asimovian agents but dealt with only the first law (against human harm). We have discussed the relationship of our work to this previous work extensively in Section 3. Another area of related work is mixed initiative planning [7, 1]. For the most part, this work focuses on single-agent to single-human interactions, whereas we focus on multiagent teams. Additionally, our work is to allow for human-agent interaction during execution, as opposed to their work, which is focused on offline plannig. None of this research addresses the issue of uncertainty addressed in our work.

Lastly, there has also been work where humans are beginning to interact with agent/robot teams [10, 3]. These efforts recognize that humans may not provide a timely response. In part to alleviate the lack of such timely response, Scerri et al [9] introduced the notion of *adjustable autonomy strategies*. Our work already incorporates such strategies, but recognizes that human users may still provide such imperfect or incorrect inputs. There has also been some work that involves humans interacting with multiagent teams actually leading to performance degradation due to imperfect input: In past work [12] illustrated that an autonomous team of simulated robots performed better than when aided with human inputs (although in some situations the humans were able to improve the simulated robot performance). However, this work did not address the question of how the simulated robots would recover from such setbacks.

In conclusion, this paper is based on the premise that Asimov's laws provide us desirable guarantees for environments where humans must work with multiagent teams. While previous work has focused operationalizing just the first of

Asimov's laws, this paper focused on the second law. In particular, the paper focused on the key insight provided by that law: agents must not blindly obey human inputs at all points. Instead, agents must pay attention to the caveat that in the law that allows for disobeying human inputs when such input leads to harm. Furthermore, we illustrated that given the uncertainty faced by the agent team and the human users, agents must attempt to reduce such uncertainty before disobeying any human input. We illustrated the results of two different domains that each have different approaches to operationalizing Asimov's laws. In the disaster rescue simulation domain, real-time response precludes detailed planning to address uncertainty; whereas in our office assistant domains, agents performed detailed decision-theoretic planning to address uncertainty. These results show that the new agent-human teams avoid the original set of problems and provide useful behavioral guarantees.

References

1. James F. Allen, Lenhart K. Schubert, George Ferguson, Peter Heeman, Chung H Hwang, Tsuneaki Kato, Marc Light, Nathaniel G. Martin, Bradford W. Miller, Massimo Poesio, and David R. Traum. The trains project: A case study in defining a conversational planning agent. Technical report, Rochester, NY, USA, 1994.
2. Isaac Asimov. *Robot Visions (collection of robot stories).* Byron Preiss Visual Publications Inc, 1990.
3. Jacob W. Crandall, Curtis W. Nielsen, and Michael A. Goodrich. Towards predicting robot team performance. In *SMC*, 2003.
4. Diana F. Gordon. Asimovian adaptive agents. *JAIR*, 13:95–153, 2000.
5. http://www.ai.sri.com/project/CALO, http://calo.sri.com. *CALO: Cognitive Agent that Learns and Organizes*, 2003.
6. M. L. Littman L. P. Kaelbling and A. R. Cassandra. Planning and acting in partially observable stochastic domains. *AI Journal*, 1998.
7. K. Myers. Advisable planning systems. In *Advanced Planning Technology*, 1996.
8. D V. Pynadath and Milind Tambe. Revisiting asimov's first law: A response to the call to arms. In *Intelligent Agents VIII Proceedings of the International workshop on Agents, theories, architectures and languages (ATAL'01)*, 2001.
9. P. Scerri, D. Pynadath, and M. Tambe. Towards adjustable autonomy for the real world. *Journal of Artificial Intelligence Research*, 17:171–228, 2002.
10. P. Scerri, D. V. Pynadath, L. Johnson, P. Rosenbloom, N. Schurr, M. Si, and M. Tambe. A prototype infrastructure for distributed robot-agent-person teams. In *AAMAS*, 2003.
11. Nathan Schurr, Janusz Marecki, Paul Scerri, J. P. Lewis, and Milind Tambe. The defacto system: Training tool for incident commanders. In *The Seventeenth Innovative Applications of Artificial Intelligence Conference (IAAI)*, 2005.
12. Nathan Schurr, Paul Scerri, and Milind Tambe. Impact of human advice on agent teams: A preliminary report. In *Workshop on Humans and Multi-Agent Systems at AAMAS. 2003.*
13. P. Varakantham, R. Maheswaran, and M. Tambe. Exploiting belief bounds: Practical pomdps for personal assistant agents. In *AAMAS*, 2005.
14. D. Weld and O. Etzioni. The first law of robotics: A call to arms. In *AAAI*, Seattle, Washington, 1994. AAAI Press.