Dangers in Multiagent Rescue using DEFACTO *

Janusz Marecki, Nathan Schurr, Milind Tambe
University of Southern California

Paul Scerri
Carnegie Mellon University

Abstract

Enabling interactions of agent-teams and humans for safe and effective Multiagent rescue is a critical area of research, with encouraging progress in the past few years. However, previous work suffers from three key limitations: (i) limited human situational awareness, reducing human effectiveness in directing agent teams, (ii) the agent team's rigid interaction strategies that jeopardize the rescue operation, and (iii) lack of formal tools to analyze the impact of such interaction strategies. This paper presents a software prototype called DEFACTO (Demonstrating Effective Flexible Agent Coordination of Teams through Omnipresence). DEFACTO is based on a software proxy architecture and 3D visualization system, which addresses the three limitations mentioned above. First, the 3D visualization interface enables human virtual omnipresence in the environment, improving human situational awareness and ability to assist agents. Second, generalizing past work on adjustable autonomy, the agent team chooses among a variety of "team-level" interaction strategies, even excluding humans from the loop in extreme circumstances. Third, analysis tools help predict the dangers of using fixed strategies for various agent teams in a future disaster response simulation scenario.

1. Introduction

One of the major issues in multi agent systems is safety understood as the impact of a team of agents on a specific domain task. Analyzing the safety of using multi agent teams interactiong with humans is critical in a large number of current and future applications[2, 5, 14, 3]. For example, current efforts emphasize humans collaboration with robot teams in space explorations, humans teaming with robots and agents for disaster rescue, as well as humans collaborating with multiple software agents for training [4, 6].

This paper focuses on the challenge of improving the effectiveness and analysing the dangers of human collaboration with agent teams. Previous work has reported encouraging progress in this arena, e.g., via proxy-based integration architectures[10], adjustable autonomy[13, 4] and agent-human dialogue [1]. Despite this encouraging progress, previous work suffers from three key limitations. First, when interacting with agent teams acting remotely, human effectiveness is hampered by low-quality interfaces. Techniques that provide telepresence via video are helpful [5], but cannot provide the global situation awareness. Second, agent teams have been equipped with adjustable autonomy (AA)[14] but not the flexibility critical in such AA. Indeed, the appropriate AA method varies from situation to situation. In some cases the human user should make most of the decisions. However, in other cases human involvement may need to be restricted. Such flexible AA techniques have been developed in domains where humans interact with individual agents [13], but whether they apply to situations where humans interact with agent teams is unknown. Third, current systems lack tools to analyze the impact of human involvement in agent teams, yet these are key to flexible AA reasoning.

We report on a software prototype system, DEFACTO (Demonstrating Effective Flexible Agent Coordination of Teams through Omnipresence), that enables agent-human collaboration and addresses the three shortcomings outlined above. First, DEFACTO incorporates a visualizer that allows for the human to have an *omnipresent* interaction with remote agent teams. We refer to this as the Omni-Viewer, and it combines two modes of operation. The Navigation Mode allows for a navigable, high quality 3D visualization of the world, whereas the Allocation Mode provides a traditional 2D view and a list of possible task allocations that the human may perform. Human experts can quickly absorb on-going agent and world activity, taking advantage of both the brain's favored visual object processing skills (relative to textual search, [9]), and the fact that 3D representations can be innately recognizable, without the layer of interpretation required of map-like displays or raw com-

* This research was supported by the United States Department of Homeland Security through the Center for Risk and Economic Analysis of Terrorism Events (CREATE). However, any opinions, findings, and conclusions or recommendations in this document are those of the author and do not necessarily reflect views of the U.S. Department of Homeland Security.

puter logs. The Navigation mode enables the human to understand the local perspectives of each agent in conjunction with the global, system-wide perspective that is obtained in the Allocation mode.

Second, to provide flexible AA, we generalize the notion of *strategies* from single-agent single-human context [13]. In our work, agents may flexibly choose among team strategies for adjustable autonomy instead of only individual strategies; thus, depending on the situation, the agent team has the flexibility to limit human interaction, and may in extreme cases exclude humans from the loop. Third, we provide a formal mathematical basis of such team strategies. These analysis tools help agents in flexibly selecting the appropriate strategy for a given situation.

We present results from detailed experiments with DEFACTO, which reveal two major surprises. First, contrary to previous results[14], human involvement is not always beneficial to an agent team— despite their best efforts, humans may sometimes end up hurting an agent team's performance. Second, increasing the number of agents in an agent-human team may also degrade the team performance, even though increasing the number of agents in a pure agent team under identical circumstances improves team performance. Fortunately, in both the surprising instances above, DEFACTO's flexible AA strategies alleviate such problematic situations.

2. DEFACTO System Details

DEFACTO consists of two major components: the Omni-Viewer and a team of proxies (see Figure 1). The Omni-Viewer allows for global and local views. The proxies allow for team coordination and communication, but more importantly also implement flexible human-agent interaction via Adjustable Autonomy. Currently, we have applied DEFACTO to a disaster rescue domain. The incident commander of the disaster acts as the *user* of DEFACTO. This disaster can either be "man made" (terrorism) or "natural" (earthquake). We focus on two urban areas: a square block that is densely covered with buildings (we use one from Kobe, Japan) and the University of Southern California campus, which is more sparsely covered with buildings. In our scenario, several buildings are initially on fire, and these fires spread to adjacent buildings if they are not quickly contained. The goal is to have a human interact with the team of fire engines in order to save the most buildings. Our overall system architecture applied to disaster response can be seen in Figure 1. While designed for real world situations, DEFACTO can also be used as a training tool for incident commanders when hooked up to a simulated disaster scenario.

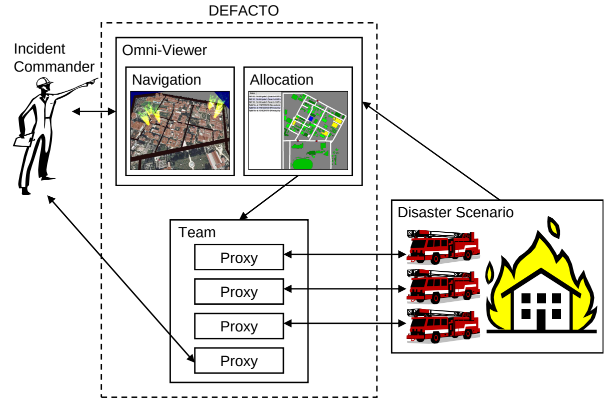
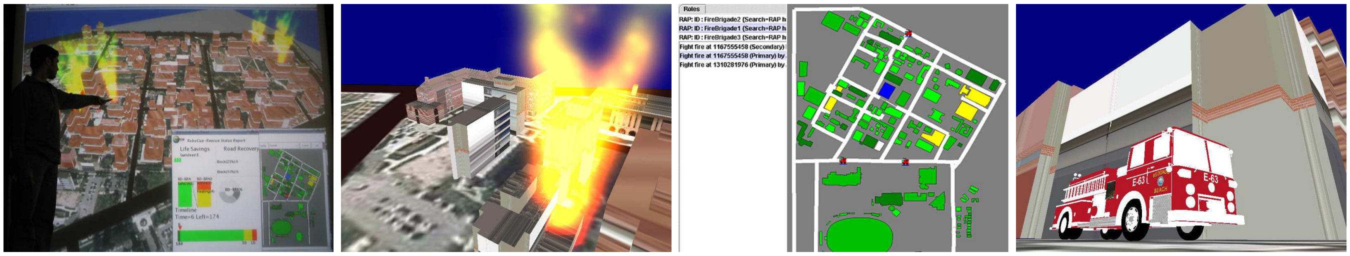


Figure 1. DEFACTO system applied to a disaster rescue.

2.1. Omni-Viewer

Our goal of allowing fluid human interaction with agents requires a visualization system that provides the human with a global view of agent activity as well as showing the local view of a particular agent when needed. Hence, we have developed an omnipresent viewer, or Omni-Viewer, which will allow the human user diverse interaction with remote agent teams. While a global view is obtainable from a two-dimensional map, a local perspective is best obtained from a 3D viewer, since the 3D view incorporates the perspective and occlusion effects generated by a particular viewpoint. The literature on 2D- versus 3D-viewers is ambiguous. For example, spatial learning of environments from virtual navigation has been found to be impaired relative to studying simple maps of the same environments [11]. On the other hand, the problem may be that many virtual environments are relatively bland and featureless. Ruddle points out that navigating virtual environments can be successful if rich, distinguishable landmarks are present [12].

To address our discrepant goals, the Omni-Viewer incorporates both a conventional map-like 2D view, Allocation Mode (Figure 2-c) and a detailed 3D viewer, Navigation Mode (Figure 2-a). The Allocation mode shows the global overview as events are progressing and provides a list of tasks that the agents have transfered to the human. The Navigation mode shows the same dynamic world view, but allows for more freedom to move to desired locations and views. In particular, the user can drop to the virtual ground level, thereby obtaining the world view (local perspective) of a particular agent. At this level, the user can "walk" freely around the scene, observing the local logistics involved as various entities are performing their duties. This can be helpful in evaluating the physical ground circumstances and altering the team's behavior accordingly. It



(a) **(b)** **(c)** **(d)**

Figure 2. Omni-Viewer during a scenario: (a) An Incident Commander using the Navigation mode spots multiple fires (b) The Commander navigates to quickly grasp the situation (c) The Commander is transferred control of the task to fight the fire and uses the Allocation mode to send a fire engine there (d) The fire has been extinguished.

also allows the user to feel immersed in the scene where various factors (psychological, etc.) may come into effect.

In order to prevent communication bandwidth issues, we assume that a high resolution 3D model has already been created and the only data that is transfered during the disaster are important changes to the world. Generating this suitable 3D model environment for the Navigation mode can require months or even years of manual modeling effort, as is commonly seen in the development of commercial videogames. However, to avoid this level of effort we make use of the work of You et. al. [15] in rapid, minimally assisted construction of polygonal models from LiDAR (Light Detection and Ranging) data. Given the raw LiDAR point data, we can automatically segment buildings from ground and create the high resolution model that the Navigation mode utilizes. The construction of the USC campus and surrounding area required only two days using this approach. LiDAR is an effective way for any new geographic area to be easily inserted into the Omni-Viewer.

2.2. Proxy: Teamwork and Adjustable Autonomy

We have built teams based on previous proxy software [13], that is in the public domain. The proxies were extended to our domain in order to take advantage of existing methods of communication, coordination, and task allocation for the team. However, these aspects are not the focus of this paper.

Instead, we focus on another key aspect of the proxies: Adjustable Autonomy. Adjustable autonomy refers to an agent's ability to dynamically change its own autonomy, possibly to transfer control over a decision to a human. Previous work on adjustable autonomy could be categorized as either involving a single person interacting with a single agent (the agent itself may interact with others) or a single person directly interacting with a team. In the single-agent single-human category, the concept of flexible transfer-of-control strategy has shown promise [13]. A transfer-of-control strategy is a preplanned sequence of actions to transfer control over a decision among multiple entities, for example, an AH_1H_2 strategy implies that an agent (A_T) attempts a decision and if the agent fails in the decision then the control over the decision is passed to a human H_1, and then if H_1 cannot reach a decision, then the control is passed to H_2. Since previous work focused on single-agent single-human interaction, strategies were individual agent strategies where only a single agent acted at a time.

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. Flexibility in such strategies implies that an agent dynamically chooses the one that is optimal, based on the situation, among multiple such strategies (H_1A, AH_1, AH_1A, etc.) rather than always rigidly choosing one strategy. The notion of flexible strategies, however, has not been applied in the context of humans interacting with agent-teams. Thus, a key question is whether such flexible transfer of control strategies are relevant in agent-teams, particularly in a large-scale application such as ours.

DEFACTO aims to answer this question by implementing transfer-of-control strategies in the context of agent teams. One key advance in DEFACTO, however, is that the strategies are not limited to individual agent strategies, but also enables team-level strategies. For example, rather than transferring control from a human to a single agent, a team-level strategy could transfer control from a human to an agent-team. Concretely, each proxy is provided with all strategy options; the key is to select the right strategy given the situation. An example of a team level strategy would combine A_T Strategy and H Strategy in order to make A_TH Strategy. The default team strategy, A_T, keeps control over a decision with the agent team for the entire duration of the decision. The H strategy always immediately transfers control to the human. A_TH strategy is the conjunction of team level A_T strategy with H strategy. This strategy aims to significantly reduced the burden on the user by allowing the decision to first pass through all agents before finally going to the user, if the agent team fails to reach a decision.

3. Mathematical Model of Strategy Selection

We develop a novel mathematical model for these team level adjustable autonomy strategies in order to enable team-level strategy selection. We first quickly review background on individual strategies from Scerri [13] before presenting our team strategies. Whereas strategies in Scerri's work are based on a single decision that is sequentially passed from agent to agent, we assume that there are multiple homogeneous agents concurrently working on multiple tasks interacting with a single human user. We exploit these assumptions (which fit our domain) to obtain a reduced version of our model and simplify the computation in selecting strategies.

3.1. Background on individual strategies

A decision, d, needs to be made. There are n entities, $e_1 \ldots e_n$, who can potentially make the decision. These entities can be human users or agents. The expected quality of decisions made by each of the entities, $\mathbf{EQ} = \{EQ_{e_i,d}(t) : \mathcal{R} \rightarrow \mathcal{R}\}_{i=1}^{n}$, is known, though perhaps not exactly. $\mathbf{P} = \{P_\top(t) : \mathcal{R} \rightarrow \mathcal{R}\}$ represents continuous probability distributions over the time that the entity in control will respond (with a decision of quality $EQ_{e,d}(t)$). The cost of delaying a decision until time t, denoted as $\{\mathcal{W} : t \rightarrow R\}$. The set of possible wait-cost functions is $\mathbf{W}$. $\mathcal{W}(t)$ is non-decreasing and at some point in time, Γ, when the costs of waiting stop accumulating (i.e., $\forall t \geq \Gamma, \forall \mathcal{W} \in \mathbf{W}, \mathcal{W}(t) = \mathcal{W}(\Gamma)$).

To calculate the EU of an arbitrary strategy, the model multiplies 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 if e_c represents the entity currently in decision-making control:

$$EU = \int_0^\infty P_\top(t) EU_{e_c,d}(t) \ .dt \qquad (1)$$

Since we are primarily interested in the effects of delay caused by transfer of control, we can decompose the expected utility of a decision at a certain instant, $EU_{e_c,d}(t)$, into two terms. The first term captures the quality of the decision, independent of delay costs, and the second captures the costs of delay: $EU_{e_c,d}t = EQ_{e,d}(t) - \mathcal{W}(t)$. To calculate the EU of a strategy, the probability of response function and the wait-cost calculation must reflect the control situation at that point in the strategy. If a human, H_1 has control at time t, $P_\top(t)$ reflects H_1's probability of responding at t.

3.2. Introduction of team level strategies

A_T **Strategy:** Starting from the individual model, we introduce team level A_T strategy, denoted as A_T in the following way: We start with Equation 2 for single agent A_T and single task d. We obtain Equation 3 by discretizing time, $t = 1, ..., T$ and introducing set Δ of tasks. Probability of agent A_T performing a task d at time t is denoted as $P_{a,d}(t)$. Equation 4 is a result of the introduction of the set of agents $AG = a_1, a_2, ..., a_k$. We assume the same quality of decision for each task performed by an agent and that each agent A_T has the same quality so that we can reduce $EQ_{a,d}(t)$ to $EQ(t)$. Given the assumption that each agent A_T at time step t performs one task, we have $\sum_{d\in\Delta} P_{a,d}(t) = 1$ which is depicted in Equation 5. Then we express $\sum_{a=a_1}^{a_k} \sum_{d\in\Delta} P_{a,d}(t) \times W_{a,d}(t)$ as the total team penalty for time slice t, i.e, at time slice t we subtract one penalty unit for each not completed task as seen in Equation 6. Assuming penalty unit $PU = 1$ we finally obtain Equation 7.

$$EU_{a,d} = \int_0^\infty P_{\top a}(t) \times (EQ_{a,d}(t) - \mathcal{W}(t)).dt \qquad (2)$$

$$EU_{a,\Delta} = \sum_{t=1}^{T} \sum_{d\in\Delta} P_{a,d}(t) \times (EQ_{a,d}(t) - \mathcal{W}(t)) \qquad (3)$$

$$EU_{A_T,\Delta} = \sum_{t=1}^{T} \sum_{a=a_1}^{a_k} \sum_{d\in\Delta} P_{a,d}(t) \times (EQ_{a,d}(t) - W_{a,d}(t)) \qquad (4)$$

$$EU_{A_T,\Delta,AG} = \sum_{t=1}^{T} (\sum_{a=a_1}^{a_k} EQ(t) - \sum_{a=a_1}^{a_k} \sum_{d\in\Delta} P_{a,d}(t) \times W_{a,d}(t)) \qquad (5)$$

$$EU_{A_T,\Delta,AG} = \sum_{t=1}^{T} (|AG| \times EQ(t) - (|\Delta| - |AG| \times t) \times PU) \qquad (6)$$

$$EU_{A_T,\Delta,AG} = |AG| \times \sum_{t=1}^{T} (EQ(t) - (\frac{|\Delta|}{AG} - t)) \qquad (7)$$

H **Strategy:** The difference between $EU_{H,\Delta,AG}$ and $EU_{A_T,\Delta,AG}$ results from three key observations: First, the human is able to choose strategic decisions with higher probability, therefore his $EQ_H(t)$ is greater than $EQ(t)$ for both individual and team level A_T strategies. Second, we hypothesize that a human cannot control all the agents AG at disposal, but due to cognitive limits will focus on a smaller subset, AG_H of agents (evidence of limits on AG_H appears later in Figure 5-a). $|AG_H|$ should slowly converge to B, which denotes its upper limit, but never exceed AG. Each function $f(AG)$ that models AG_H should be consistent with three properties: i) if $B \rightarrow \infty$ then $f(AG) \rightarrow AG$;

ii) $f(AG) < B$; iii) $f(AG) < AG$. Third, there is a delay in human decision making compared to agent decisions. We model this phenomena by shifting H to start at time slice t_H. For $t_H - 1$ time slices the team incurs a cost $|\Delta| \times (t_H - 1)$ for all incomplete tasks. By inserting $EQ_H(t)$ and AG_H into the time shifted utility equation for A_T strategy we obtain the H strategy (Equation 8).

$A_T H$ **Strategy:** The $A_T H$ strategy is a composition of H and A_T strategies (see Equation 9).

$$EU_{H,\Delta,AG} = |AG_H| \times \sum_{t=t_H}^{T} (EQ_H(t) - (\frac{|\Delta|}{AG_H} - (t - t_H))) - |\Delta| \times (t_H - 1) \quad (8)$$

$$EU_{A_T H,\Delta,AG} = |AG| \times \sum_{t=1}^{t_H-1} (EQ(t) - (\frac{|\Delta|}{|AG|} - t)) + |AG_H| \times \sum_{t=t_H}^{T} (EQ_H(t) - (\frac{|\Delta| - |AG|}{|AG_H|} - (t - t_H)))) \quad (9)$$

Strategy utility prediction: Given our strategy equations and the assumption that $EQ_{H,\Delta,AG}$ is constant and independent of the number of agents we plot the graphs representing strategy utilities (Figure 3). Figure 3 shows the number of agents on the x-axis and the expected utility of a strategy on the y-axis. We focus on humans with different skills: (a) low EQ_H, low B (b) high EQ_H, low B (c) low EQ_H, high B (d) high EQ_H, high B. The last graph representing a human with high EQ_H and high B follows results presented in [13] (and hence the expected scenario), we see the curve of AH and $A_T H$ flattening out to eventually cross the line of A_T. Moreover, we observe that the increase in EQ_H increases the slope for AH and $A_T H$ for small number of agents, whereas the increase of B causes the curve to maintain a slope for larger number of agents, before eventually flattening out and crossing the A_T line.

4. Experiments and Evaluation

Our DEFACTO system was evaluated in three key ways, with the first two focusing on key individual components of the DEFACTO system and the last attempting to evaluate the entire system. First, we performed detailed experiments comparing the effectiveness of Adjustable Autonomy (AA) strategies over multiple users. In order to provide DEFACTO with a dynamic rescue domain we chose to connect it to a simulator. We chose the previously developed RoboCup Rescue simulation environment [8]. In this simulator, fire engine agents can search the city and attempt to extinguish any fires that have started in the city. To interface with DEFACTO, each fire engine is controlled by a proxy in order to handle the coordination and execution of AA strategies. Consequently, the proxies can try to allocate fire engines to fires in a distributed manner, but can also transfer control to the more expert user. The user can then use the Omni-Viewer in Allocation mode to allocate engines to the fires that he has control over. In order to focus on the AA strategies (transferring the control of task allocation) and not have the users ability to navigate interfere with results, the Navigation mode was not used during this first set of experiments.

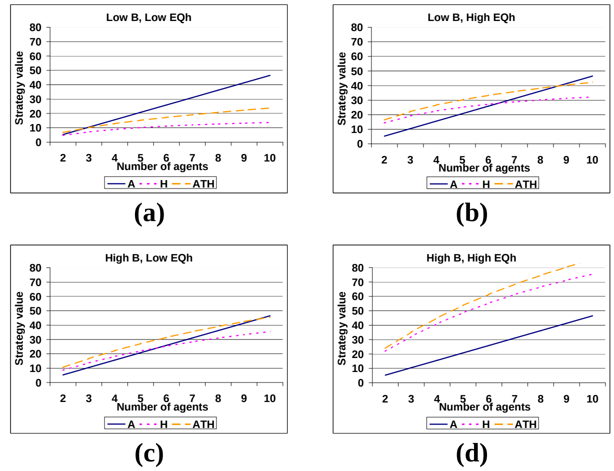


Figure 3. Model predictions for various users.

The results of our experiments are shown in Figure 4, 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 $A_T H$; their performance was compared with the completely autonomous A_T strategy. AH is an individual agent strategy, tested for comparison with $A_T H$, where agents act individually, and pass those tasks to a human user that they cannot immediately perform. Each experiment was conducted with the same initial locations of fires and building damage. For each strategy we tested, varied the number of fire engines between 4, 6 and 10. Each chart in Figure 4 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, strategy A_T saves 50 building with 4 agents. Each data point on the graph is an average of three runs. Each run itself took 15 minutes, and each user was required to participate in 27 experiments, which together with 2 hours of getting oriented with the system, equates to about 9 hours of experiments per volunteer.

Figure 4 enables us to conclude the following:

- *Human involvement with agent teams does not necessarily lead to improvement in team performance.*

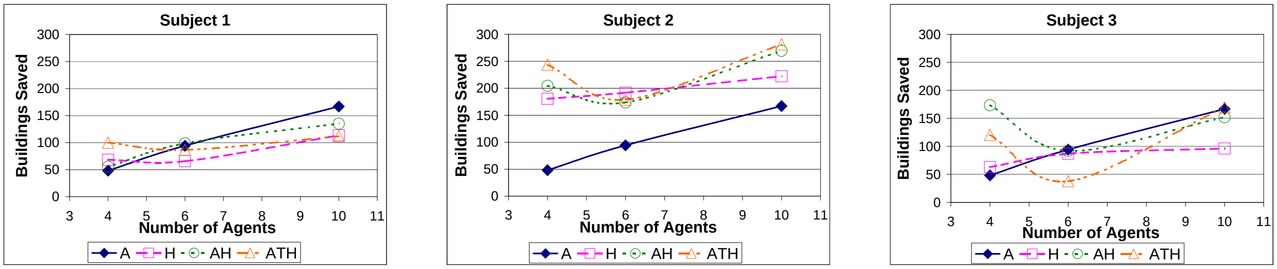


Figure 4. Performance of subjects 1, 2, and 3.

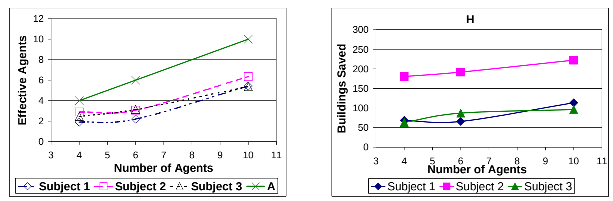
Contrary to expectations and prior results, human involvement does not uniformly improve team performance, as seen by human-involving strategies performing worse than the A_T strategy in some instances. For instance, for subject 3, human involving strategies such as AH provide a somewhat higher quality than A_T for 4 agents, yet at higher numbers of agents, the strategy performance is lower than A_T. While our strategy model predicted such an outcome in cases of *High B, Low* EQ_H, the expected scenario was *High B, High* EQ_H.

- *Providing more agents at a human's command does not necessarily improve the agent team performance* As seen for subject 2 and subject 3, increasing agents from 4 to 6 given AH and A_TH strategies is seen to degrade performance. In contrast, for the A_T strategy, the performance of the fully autonomous agent team continues to improve with additions of agents, thus indicating that the reduction in AH and A_TH performance is due to human involvement. As the number of agents increase to 10, the agent team does recover.
- *No strategy dominates through all the experiments given varying numbers of agents.* For instance, at 4 agents, human-involving strategies dominate the A_T strategy. However, at 10 agents, the A_T strategy outperforms all possible strategies for subjects 1 and 3.
- *Complex team-level strategies are helpful in practice*: A_TH leads to improvement over H with 4 agents for all subjects, although surprising domination of AH over A_TH in some cases indicates that AH may also a useful strategy to have available in a team setting.

Note that the phenomena described range over multiple users, multiple runs, and multiple strategies. The most important conclusion from these figures is that flexibility is necessary to allow for the optimal AA strategy to be applied. The key question is then whether we can leverage our mathematical model to select among strategies. However, we must first check if we can model the phenomenon in our domain accurately. To that end, we compare the predictions at the end of Section 3 with the results reported in Figure 4. If we temporarily ignore the "dip" observed at 6 agents in AH and A_TH strategies, then subject 2 may be modeled as a *High B, High* EQ_H subject, while subjects 1 and 3 modeled via *High B, Low* EQ_H. (Figure 5-(b) indicates an identical improvement in H for 3 subjects with increasing agents, which suggests that B is constant across subjects.) Thus, by estimating the EQ_H of a subject by checking the "H" strategy for small number of agents (say 4), and comparing to A strategy, we may begin to select the appropriate strategy.

Strategy	H			AH			A_TH		
# of agents	4	6	10	4	6	10	4	6	10
Subject 1	91	92	154	118	128	132	104	83	64
Subject 2	138	129	180	146	144	72	109	120	38
Subject 3	117	132	152	133	136	97	116	58	57

Table 1. Total amount of allocations given.



(a) **(b)**

Figure 5. (a) AG_H **and (b)** H **performance**

Unfortunately, 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 4), whereas our mathematical model would have predicted an increase in performance as the number of agents increased (as seen in Figure 3). It would be useful to understand which of our key assumptions in the model has led to such a mismatch in prediction.

The crucial assumptions in our model were that while numbers of agents increase, AG_H steadily increases and EQ_H remains constant. Thus, the dip at 6 agents is essentially affected by either AG_H or EQ_H. We first tested AG_H in our domain. The amount of effective agents, AG_H, is cal-

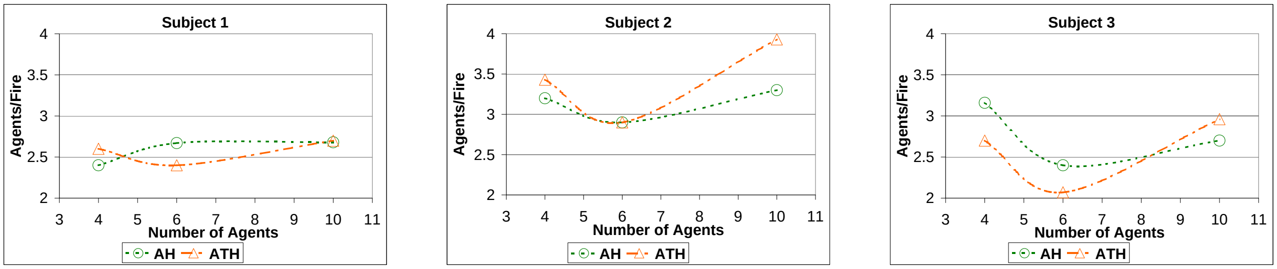


Figure 7. Amount of agents per fire assigned by subjects 1, 2, and 3

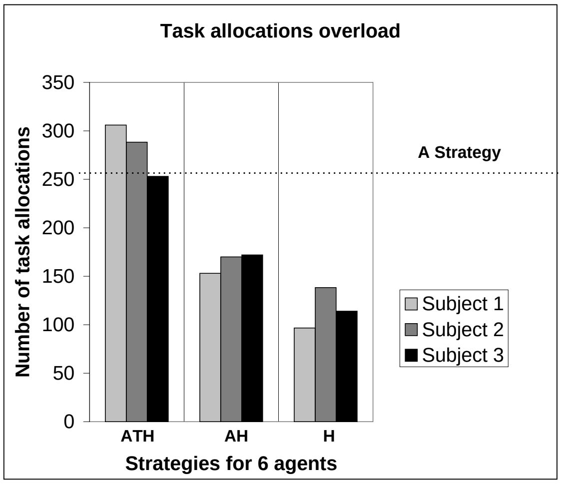


Figure 6. Task allocation overload for the team of 6 agents

culated by dividing how many total allocations each subject made by how many the A_T strategy made per agent, assuming A_T strategy effectively uses all agents. Figure 5-(a) shows the number of agents on the x-axis and the number of agents effective used, AG_H, on the y-axis; the A_T strategy, which is using all available agents, is also shown as a reference. However, the amount of effective agents is actually about the same in 4 and 6 agents. This would not account for the sharp drop we see in the performance. We then shifted our attention to the EQ_H of each subject. One reduction in EQ_H could be because subjects simply did not send as many allocations totally over the course of the experiments. This, however is not the case as can be seen in Table 1 where for 6 agents, the total amount of allocations given is comparable to that of 4 agents. To investigate further, we checked if the quality of human allocation had degraded. For our domain, the more fire engines that fight the same fire, the more likely it is to be extinguished and in less time. For this reason, the amount of agents that were tasked to each fire is a good indicator of the quality of allocations that the subject makes. Our model expected the amount of agents that each subject tasked out to each fire would remain independent of the number of agents. Figure 7 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 $A_T H$ for 6 agents result in significantly less average fire engines per task (fire) and therefore less average EQ_H.

Strategy	4 agents	6 agents	10 agents
AH	34	**75**	14
$A_T H$	54	**231**	47

Table 2. Task conflicts for subject 2.

The next question is then to understand why for 6 agents AH and $A_T H$ result in lower average fire engines per fire. One hypothesis is the possible interference among the agents' self allocations vs human task allocations at 6 agents. Table 2 shows the number of task changes for 4, 6 and 10 agents for AH and $A_T H$ strategies, showing that maximum occurs at 6 agents. A task change occurs because an agent pursuing its own task is provided another task by a human or a human-given task is preempted by the agent. Thus, when running mixed agent-human strategies, the possible clash of tasks causes a significant increase task changes, resulting in the total amount of task allocations overreaching the number of task allocations for the A strategy (Figure 6) . While the reason for such interference peaking at 6 may be domain specific, the key lesson is that interference has the potential to occur in complex team-level strategies. Our model would need to take into account such interference effects by not assuming a constant EQ_H.

The second aspect of our evaluation was to explore the benefits of the Navigation mode (3D) in the Omni-Viewer over solely an Allocation mode (2D). We performed 2 tests on 20 subjects. All subjects were familiar with the university campus. Test 1 showed Navigation and Allocation mode screenshots of the university campus to subjects. Subjects were asked to identify a unique building on campus, while timing each response. The average time for a subject to find the building in 2D was 29.3 seconds, whereas the 3D allowed them to find the same building in an average of 17.1

seconds. Test 2 again displayed Navigation and Allocation mode screenshots of two buildings on campus that had just caught fire. In Test 2, subjects were asked first asked to allocate fire engines to the buildings using only the Allocation mode. Then subjects were shown the Navigation mode of the same scene. 90 percent of the subjects actually chose to change their initial allocation, given the extra information that the Navigation mode provided.

5. Related Work and Summary

We have discussed related work throughout this paper, however, we now provide comparisons with key previous agent software prototypes and research. Given our application domains, Scerri et al's work on robot-agent-person (RAP) teams for disaster rescue is likely the most closely related [13]. Our work takes a significant step forward in comparison. First, the omni-viewer enables navigational capabilities improving human situational awareness not present in previous work. Second, we provide a mathematical model based on strategies, which we experimentally verify, absent in that work. Third, we provide extensive experimentation, and illustrate that some of the conclusions reached in [13] were indeed preliminary, e.g., they conclude that human involvement is always beneficial to agent team performance, while our more extensive results indicate that sometimes agent teams are better off excluding humans from the loop. Human interactions in agent teams is also investigated in [15,2], and there is significant research on human interactions with robot-teams [5, 3]. However they do not use flexible AA strategies and/or team-level AA strategies. Furthermore, our experimental results may assist these researchers in recognizing the potential for harm that humans may cause to agent or robot team performance. Significant attention has been paid in the context of adjustable autonomy and mixed-initiative in single-agent single-human interactions [7, 1]. However, this paper focuses on new phenomena that arise in human interactions with agent teams.

This paper addresses the issue of safety in multi-agent systems understood as the performance the multi agent system shows when applied to a real world domain. To this end, we present a large-scale prototype, DEFACTO, that is based on a software proxy architecture and 3D visualization system and provides three key advances over previous work. First, DEFACTO's Omni-Viewer enables the human to both improve situational awareness and assist agents, by providing a navigable 3D view along with a 2D global allocation view. Second, DEFACTO incorporates flexible AA strategies, even excluding humans from the loop in extreme circumstances. Third, analysis tools help predict the bahavior of the agent team and choose the safest strategy for the given domain.

We performed detailed experiments using DEFACTO, leading to some surprising results. These results illustrate that an agent team must be equipped with flexible strategies for adjustable autonomy, so that they may select the safest strategy autonomously.

References

[1] J. F. Allen. The TRAINS project: A case study in building a conversational planning agent. *Journal of Experimental and Theoretical AI (JETAI)*, 7:7–48, 1995.

[2] M. H. Burstein, A. M. Mulvehill, and S. Deutsch. An approach to mixed-initiative management of heterogeneous software agent teams. In *HICSS*, page 8055. IEEE Computer Society, 1999.

[3] J. W. Crandall, C. W. Nielsen, and M. A. Goodrich. Towards predicting robot team performance. In *SMC*, 2003.

[4] G. Dorais, R. Bonasso, D. Kortenkamp, P. Pell, and D. Schreckenghost. Adjustable autonomy for human-centered autonomous systems on mars. In *Mars*, 1998.

[5] T. Fong, C. Thorpe, and C. Baur. Multi-robot remote driving with collaborative control. *IEEE Transactions on Industrial Electronics*, 2002.

[6] R. Hill, J. Gratch, S. Marsella, J. Rickel, W. Swartout, and D. Traum. Virtual humans in the mission rehearsal exercise system. In *KI Embodied Conversational Agents*, 2003.

[7] E. Horvitz. Principles of mixed-initiative user interfaces. In *Proceedings of ACM SIGCHI Conference on Human Factors in Computing Systems (CHI'99)*, pages 159–166, Pittsburgh, PA, May 1999.

[8] H. Kitano, S. Tadokoro, I. Noda, H. Matsubara, T. Takahashi, A. Shinjoh, and S. Shimada. Robocup rescue: Search and rescue in large-scale disasters as a domain for autonomous agents research. In *IEEE SMC*, volume VI, pages 739–743, Tokyo, October 1999.

[9] A. Paivio. Pictures and words in visual search. *Memory & Cognition*, 2(3):515–521, 1974.

[10] D. V. Pynadath and M. Tambe. Automated teamwork among heterogeneous software agents and humans. *Journal of Autonomous Agents and Multi-Agent Systems (JAAMAS)*, 7:71–100, 2003.

[11] A. Richardson, D. Montello, and M. Hegarty. Spatial knowledge acquisition from maps and from navigation in real and virtual environments. *Memory and Cognition*, 27(4):741–750, 1999.

[12] R. Ruddle, S. Payne, and D. Jones. Navigating buildings in desktop virtual environments: Experimental investigations using extended navigational experience. *J. Experimental Psychology - Applied*, 3(2):143–159, 1997.

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

[14] 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.

[15] U. N. Suya You, Jinhui Hu and P. Fox. Urban site modeling from lidar. In *Proc. 2nd Int'l Workshop Computer Graphics and Geometric Modeling (CGGM)*, pages 579–588, 2003.