

# Towards Flexible Coordination of Human-Agent Teams

**Nathan Schurr, Janusz Marecki, Milind Tambe**

University of Southern California  
Powell Hall of Engineering  
3737 Watt Way  
Los Angeles, CA 90089-0781  
Phone: 213-740-9560  
Fax: 213-740-7877  
{marecki, schurr, tambe}@usc.edu

**Paul Scerri**

Carnegie Mellon University  
Robotics Institute  
5000 Forbes Avenue  
Pittsburgh, PA 15213  
Phone: 412-268-2145  
Fax: 412-268-6436  
pscerri@cs.cmu.edu

Corresponding Author: Nathan Schurr

Keywords: Multiagent Systems, Adjustable Autonomy, Teamwork

## Abstract

Enabling interactions of agent-teams and humans 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 limit team performance, and (iii) lack of formal tools to analyze the impact of such interaction strategies. This article 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 performance of (and choose among) different interaction strategies. DEFACTO is illustrated in a future disaster response simulation scenario, and extensive experimental results are presented.

## 1 Introduction

Human interaction with agent teams is critical in a large number of current and future applications[2, 5, 18, 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, 7].

This article focuses on the challenge of improving the effectiveness of human collaboration with agent teams. Previous work has reported encouraging progress in this arena, e.g., via proxy-based integration architectures [13], adjustable autonomy [17, 4] and agent-human dialog [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 tele-presence via video are helpful [5], but cannot provide the global situation awareness. Second, agent teams have been equipped with adjustable autonomy (AA)[18] 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 [17], 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.

This article reports 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. This is referred 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, [12]), and the fact that 3D representations can be innately recognizable, without the layer of interpretation required of map-like displays or raw computer 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, this article generalizes the notion of *strategies* from single-agent single-human context [17]. In this 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, this article provides a formal mathematical basis of such team strategies.

These analysis tools help agents in flexibly selecting the appropriate strategy for a given situation.

Results are presented from detailed experiments with DEFACTO, which reveal two major surprises. First, contrary to previous results[18], 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.

DEFACTO serves as a prototype of a future disaster response system. In particular, the key hypothesis in DEFACTO is that, in the future, teams comprised of agent-assisted response vehicles, robots and people can work together during a disaster rescue. A more current application is training incident commanders by simulating agent behaviors to act as the disaster rescue personnel (resources) and allowing the incident commander to allocate them. DEFACTO has been repeatedly demonstrated to key police and fire department personnel in Los Angeles area, with very positive feedback.

## **2 Mathematical Model of Strategy Selection**

Whereas strategies in Scerri’s work [17] are based on a single decision that is sequentially passed from agent to agent, here it is assumed that there are multiple homogeneous agents concurrently working on multiple tasks interacting with a sin-

gle human user. These assumptions (which fit the domain) are exploited to obtain a reduced version of the model and simplify the computation in selecting strategies.

## 2.1 Background on individual strategies

A decision,  $d$ , needs to be made. There are  $n$  entities,  $e_1 \dots e_n$ , 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,  $\Gamma$ , 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:

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

Since this article's primary interest is in the effects of delay caused by transfer of control, the expected utility of a decision at a certain instant,  $EU_{e_c,d}(t)$ ,

is decomposed 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  $t$ .

## 2.2 Introduction of team level strategies

**$A_T$  Strategy:** Starting from the individual model, team level  $A_T$  strategy, denoted as  $A_T$ , are introduced in the following way: Start with Equation 2 for single agent  $A_T$  and single task  $d$ . Obtain Equation 3 by discretizing time,  $t = 1, \dots, T$  and introducing set  $\Delta$  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  $AG = a_1, a_2, \dots, a_k$ . 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  $EQ_{a,d}(t)$  reduces to  $EQ(t)$ . Given the assumption that each agent  $A_T$  at time step  $t$  performs one task,  $\sum_{d \in \Delta} P_{a,d}(t) = 1$  which is depicted in Equation 5. Then 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  $t$ , subtract one penalty unit for each not completed task as seen in Equation 6. Assuming penalty unit  $PU = 1$  finally resulting Equation 7.

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

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

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

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

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

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

**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, this article hypothesizes 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 6-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. This phenomena is modeled 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 to 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 the strategy equations and the assumption that  $EQ_{H,\Delta,AG}$  is constant and independent of the number of agents the graphs representing strategy utilities are plotted (Figure 1). Figure 1 shows the number of agents on the x-axis and the expected utility of a strategy on the y-axis. The focus is 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), the curve of  $AH$  and  $A_TH$  appears to be flattening out to eventually cross the line of  $A_T$ . Moreover, observe that the increase in  $EQ_H$  increases the slope for  $AH$  and  $A_TH$  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.

### 3 DEFACTO System Details

DEFACTO consists of two major components: the Omni-Viewer and a team of proxies (see Figure 2). 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, DEFACTO is applied 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). The experiments in a later section focus on two urban areas: a square block that is densely covered with buildings (from Kobe, Japan) and the University of Southern California (USC) campus, which is more sparsely covered with buildings. In this 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. The overall system architecture applied to disaster response can be seen in Figure 2. 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.

### **3.1 Omni-Viewer**

An omnipresent viewer, or Omni-Viewer (see Figure 3), has been developed in order to allow the human user fluid 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 [14]. 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 [15].

To address these discrepant goals, the Omni-Viewer incorporates both a conventional map-like 2D view, Allocation Mode (Figure 3-d) and a detailed 3D viewer, Navigation Mode (Figure 3-c). The Allocation mode shows the global overview as events are progressing and provides a list of tasks that the agents have transferred 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 par-

ticular, 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 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, it is assumed that a high resolution 3D model has already been created and the only data that is transferred 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 video-games. However, to avoid this level of effort, the work of You et. al. [20] is leveraged toward rapid, minimally assisted construction of polygonal models from LiDAR (Light Detection and Ranging) data. Given the raw LiDAR point data, buildings can automatically be segmented from the 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.

### **3.2 Proxy: Team Coordination**

The critical role of intelligent agents will be to manage coordination between all members of the response team. Specifically, DEFACTO uses coordination algorithms inspired by theories of teamwork to manage the distributed response [17].

The general coordination algorithms are encapsulated in *proxies*, with each team member having its own proxy and representing it in the team. The current version of the proxies is called *Machinetta* [16] and extends the successful Teamcore proxies [13]. *Machinetta* is implemented in Java and is freely available on the web. Notice that the concept of a reusable proxy differs from many other “multi-agent toolkits” in that it provides the coordination *algorithms*, e.g., algorithms for allocating tasks, as opposed to the *infrastructure*, e.g., APIs for reliable communication.

The *Machinetta* software consists of five main modules, three of which are domain independent and two of which are tailored for specific domains. The three domain independent modules are for coordination reasoning, maintaining local beliefs (state) and adjustable autonomy. The domain specific modules are for communication between proxies and communication between a proxy and a team member. The modules interact with each other only via the local state with a blackboard design and are designed to be “plug and play.” Thus new adjustable autonomy algorithms can be used with existing coordination algorithms. The coordination reasoning is responsible for reasoning about interactions with other proxies, thereby implementing the coordination algorithms. The adjustable autonomy algorithms reason about the interaction with the team member, providing the possibility for the team member to make any coordination decision instead of the proxy.

---

**Communication:** communication with other proxies

**Coordination:** reasoning about team plans and communication

**State:** the working memory of the proxy

**Adjustable Autonomy:** reasoning about whether to act autonomously or pass control to the team member

**RAP Interface:** communication with the team member

---

Teams of proxies implement *team oriented plans* (TOPs) which describe joint activities to be performed in terms of the individual *roles* to be performed and any constraints between those roles. Generally, TOPs are instantiated dynamically from TOP templates at runtime when preconditions associated with the templates are filled. Typically, a large team will be simultaneously executing many TOPs. For example, a disaster response team might be executing multiple fight fire TOPs. Such fight fire TOPs might specify a breakdown of fighting a fire into activities such as checking for civilians, ensuring power and gas is turned off, and spraying water. Constraints between these roles will specify interactions such as required execution ordering and whether one role can be performed if another is not currently being performed. Notice that TOPs do not specify the coordination or communication required to execute a plan; the proxy determines the coordination that should be performed.

Current versions of Machinetta include state-of-the-art algorithms for plan instantiation [11], role allocation [25], information sharing [24], task deconfliction [11], and adjustable autonomy [17]. Key algorithms, including role allocation, resource allocation, information sharing and plan instantiation are based on the use of *tokens* which are “pushed” onto the network and routed to where they are required by the proxies. For example, the role allocation algorithm, explained here [25], represents each role to be allocated with a token and pushes the tokens around the

network until a sufficiently capable and available team member is found to execute the role. The implementation of the coordination algorithms uses the abstraction of a simple mobile agent to implement the tokens, leading to robust and efficient software.

### 3.3 Proxy: 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 [17]. 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 ( $A$ ) 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 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 reduce 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.

## 4 Experiments and Evaluation

The 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, detailed experiments were performed comparing the effectiveness of Adjustable Autonomy (AA) strategies over multiple users. In order to provide DEFACTO with a dynamic rescue domain it is connected to a sim-

ulator. The previously developed RoboCup Rescue simulation environment [9] is used. 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.

The results of these experiments are shown in Figure 5, 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, three strategies are tested, specifically,  $H$ ,  $AH$  and  $A_TH$ ; their performance was compared with the completely autonomous  $A_T$  strategy.  $AH$  is an individual agent strategy, tested for comparison with  $A_TH$ , 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 tested, varied the number of fire engines between 4, 6 and 10. Each chart in Figure 5 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 5 enables us to conclude the following:

- *Human involvement with agent teams does not necessarily lead to improvement in 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  $A_T$  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 the 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 im-*

provement over  $H$  with 4 agents for all subjects, although surprising domination of  $AH$  over  $A_T H$  in some cases indicates that  $AH$  may also be 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 the mathematical model can be leveraged to select among strategies. However, first the model must be checked if it can explain the phenomenon in the domain accurately. To that end, the predictions are compared at the end of Section 2 with the results reported in Figure 5. If the “dip” observed at 6 agents in  $AH$  and  $A_T H$  strategies is temporarily ignored, 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 6-(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, the appropriate strategy may now be selected.

Unfortunately, the strategies including the humans and agents ( $AH$  and  $A_T H$ ) for 6 agents show a noticeable decrease in performance for subjects 2 and 3 (see Figure 5), whereas the mathematical model would have predicted an increase in performance as the number of agents increased (as seen in Figure 1). It would be useful to understand which of the key assumptions in the model has led to such a mismatch in prediction.

The crucial assumptions in the model were that while numbers of agents in-

crease,  $AG_H$  steadily increases and  $EQ_H$  remains constant. Thus, the dip at 6 agents is essentially affected by either  $AG_H$  or  $EQ_H$ . First,  $AG_H$  is tested in the domain. The amount of effective agents,  $AG_H$ , is calculated 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 6-(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 seen in the performance. Then attention is turned 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, the quality of human allocation is studied to see if it has degraded. For the 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. The 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  $A_TH$  for 6 agents result in significantly less average fire engines per task (fire) and therefore less average  $EQ_H$ .

The next question is then to understand why for 6 agents  $AH$  and  $A_TH$  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  $A_TH$  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. 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. The model would need to take into account such interference effects by not assuming a constant  $EQ_H$ .

The second aspect of the evaluation was to explore the benefits of the Navigation mode (3D) in the Omni-Viewer over solely an Allocation mode (2D). 2 tests were performed on 20 subjects. All subjects were familiar with the USC 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.

Third, the complete DEFACTO system has been periodically demonstrated to key government agencies, public safety officials and disaster researchers for assessing its utility by the ultimate consumers of the technology, with exciting feedback.

Indeed they were eager to deploy DEFACTO and begin using it as a research tool to explore the unfolding of different disasters. For example, during one of the demonstrations on Nov 18, 2004 Gary Ackerman, a Senior Research Associate at the Center for Nonproliferation Studies at the Monterey Institute of International Studies pointed out in reference to DEFACTO, *“This is exactly the type of system we are looking for”* to study the potential effect of terrorist attacks. Also, the authors have met with several public safety officials about using DEFACTO as a training tool for their staff. According to Los Angeles County Fire Department Fire Captain Michael Lewis: *“Effective simulation programs for firefighters must be realistic, relevant in scope, and imitate the communication challenges on the fire ground. DEFACTO focuses on these very issues.”*

## **5 Related Work**

First, current methods of training within the Los Angeles Fire Department (LAFD) are discussed. In order to train incident commanders, the LAFD uses a projection screen to simulate the disaster (Figure 8-(a)). In addition, the participating incident commander is seated at a desk, directing an assistant to take notes (Figure 8-(b)). Other firefighters remain in the back of the room and communicate to the incident commander via radios. Firefighters are taken temporarily out of duty in order to help act out these pre-determined scenarios in order to test the incident commander’s abilities.

Second, current simulation tools are mentioned. Among the current tools aimed at simulating rescue environments, it is important to mention products like JCATS [22] and EPICS [10]. JCATS represents a self-contained, high-resolution

joint simulation in use for entity-level training in open, urban and subterranean environments. Developed by Lawrence Livermore National Laboratory, JCATS gives users the capability to detail the replication of small group and individual activities during a simulated operation. At this point however, JCATS cannot simulate agents. Finally, EPICS is a computer-based, scenario-driven, high-resolution simulation. It is used by emergency response agencies to train for emergency situations that require multi-echelon and/or inter-agency communication and coordination. Developed by the U.S. Army Training and Doctrine Command Analysis Center, EPICS is also used for exercising communications and command and control procedures at multiple levels. Similar to JCATS however, EPICS does not currently allow agents to participate in the simulation.

Third, related work in the area of agents is discussed. Hill et al's work is a similar immersive training tool [21]. Their work focused more on multi-modal dialog and emphasize single agent interaction along predefined story lines, whereas this work focuses on adjustable autonomy and coordinating large numbers of agents in a dynamic, complex fire-fighting domain. In the past, agent-based simulations have been designed with the aim of training military helicopter pilots as well [6]. Our simulations allow for more complex scenarios and also allow for adjustable autonomy between the trainee and the team of agents. Also, there is some related work in agents being done at Honeywell Laboratories that assists first responders [23]. In their work, however they focus on helping human first responders communicate and coordinate, as opposed to this work, where a single first responder manage and assist a team of agents.

This work is in the same thread as Scerri et al's previous work on robot-agent-person (RAP) teams for disaster rescue [18] is closely related to DEFACTO. This

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, provide team-level strategies are provided, which are experimentally verified, absent in that work. Third, extensive experimentation is provided, and illustrates that some of the conclusions reached in [18] were indeed preliminary. For example, they conclude that human involvement is always beneficial to agent team performance, while these more extensive results indicate that sometimes agent teams are better off excluding humans from the loop. Human interactions in agent teams has also been investigated in [2, 20], 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, the experimental results here 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 [8, 1]. However, this article focuses on new phenomena that arise in human interactions with agent teams.

## **6 Conclusion**

This article presents a large-scale operational prototype, DEFACTO. DEFACTO incorporates state of the art proxy framework, 3D visualization and Adjustable Autonomy (AA) human-interaction reasoning. This 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 performance of (and choose among) different interaction strategies. 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 appropriate strategy autonomously. Exciting feedback from DEFACTO's ultimate consumers illustrates its promise and potential for real-world

## 7 Acknowledgements

Thanks to J. P. Lewis and Nikhil Kasinadhuni for their work on the DEFACTO Navigation Viewer. Also, thanks to Ronald Roemer, David Perez, and Roland Sprewell for their time and invaluable input to this project.

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.

An earlier version of this paper appears here [19].

## 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 *Hawaii International Conference on System Sciences*, page 8055. IEEE Computer Society, 1999.



- [3] J. W. Crandall, C. W. Nielsen, and M. A. Goodrich. Towards predicting robot team performance. In *IEEE International Conf on Systems, Man and Cybernetics*, 2003.
- [4] G. Dorais, R. Bonasso, D. Kortenkamp, P. Pell, and D. Schreckenghost. Adjustable autonomy for human-centered autonomous systems on mars. In *Presented at the Mars Society Conference.*, 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. Chen, J. Gratch, P. Rosenbloom, and M. Tambe. Intelligent agents for the synthetic battlefield: A company of rotary wing aircraft. In *Proceedings of Innovative Applications of Artificial Intelligence Conferene (IAAI)*, 1997.
- [7] R. Hill, J. Gratch, S. Marsella, J. Rickel, W. Swartout, and D. Traum. Virtual humans in the mission rehearsal exercise system. In *Kuenstliche-Intelligenz Embodied Conversational Agents*, 2003.
- [8] 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.
- [9] 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 International Conf on Systems, Man and Cybernetics*, volume VI, pages 739–743, Tokyo, October 1999.
- [10] L. L. N. Laboratory. Jcats - joint conflict and tactical simulation. In [http://www.jfcom.mil/about/fact\\_jcats.htm](http://www.jfcom.mil/about/fact_jcats.htm), 2005.
- [11] E. Liao, P. Scerri, and K. Sycara. A framework for very large teams. In *AAMAS'04 Workshop on Coalitions and Teams*, 2004.
- [12] A. Paivio. Pictures and words in visual search. *Memory & Cognition*, 2(3):515–521, 1974.
- [13] 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.

- [14] 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.
- [15] R. Ruddle, S. Payne, and D. Jones. Navigating buildings in desktop virtual environments: Experimental investigations using extended navigational experience. *Journal of Experimental Psychology - Applied*, 3(2):143–159, 1997.
- [16] P. Scerri, E. Liao, Y. Xu, M. Lewis, G. Lai, and K. Sycara. *Theory and Algorithms for Cooperative Systems*, chapter Coordinating very large groups of wide area search munitions. World Scientific Publishing, 2004.
- [17] P. Scerri, D. Pynadath, and M. Tambe. Towards adjustable autonomy for the real world. *Journal of Artificial Intelligence Research*, 17:171–228, 2002.
- [18] 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 *Proceedings of the International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, 2003.
- [19] N. Schurr, J. Marecki, P. Scerri, J. Lewis, and M. Tambe. The defacto system: Training tool for incident commanders. In *Proceedings of the Seventeenth Innovative Applications of Artificial Intelligence Conference (IAAI)*, 2005.
- [20] 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.
- [21] W. Swartout, R. Hill, J. Gratch, W. L. Johnson, C. Kyriakakis, C. LaBore, R. Lindheim, S. Marsella, D. Miraglia, B. Moore, J. Morie, J. Rickel, M. Thiebaut, L. Tuch, R. Whitney, and J. Douglas. Toward the holodeck: integrating graphics, sound, character and story. In *Proceedings of the Fifth International Conference on Autonomous Agents*, pages 409–416. ACM Press, 2001.
- [22] A. S. Technology. Epics - emergency preparedness incident commander simulation. In <http://epics.astcorp.com>, 2005.
- [23] V. G. R. V. R. Thomas Wagner, John Phelps. Coordinators: Coordination managers for first responders. In *Proceedings of the International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, 2004.

- [24] Y. Xu, M. Lewis, K. Sycara, and P. Scerri. Information sharing in very large teams. In *In AAMAS'04 Workshop on Challenges in Coordination of Large Scale MultiAgent Systems*, 2004.
- [25] Y. Xu, P. Scerri, B. Yu, S. Okamoto, M. Lewis, and K. Sycara. An integrated token-based algorithm for scalable coordination. In *Proceedings of the International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, 2005.

## 8 Authors' Short Bios

Nathan Schurr is a PhD candidate of Computer Science in the Viterbi School of Engineering at University of Southern California. His research interests are in human interaction with multiagent systems. He was awarded the Viterbi School of Engineering Homeland Security Center Doctoral Fellowship. He served on the Program Committee for AI Technologies for Homeland Security session at the American Association of Artificial Intelligence (AAAI) Spring Symposium 2005.

Janusz Marecki is a PHD student of Computer Science at University of Southern California (USC). His research areas are primarily autonomous agents and multi-agent systems. He earned a Bachelor of Science degree at Academy of Computer Science and Management (Poland) and a Master of Science degree at the Jagiellonian University (Poland). He was awarded the Erasmus scholarship to study at Universite Montpellier II in France. He also worked for the European Laboratory for Nuclear Research (CERN).

Milind Tambe is an Associate Professor of Computer Science at University of Southern California(USC). He received his Ph.D. from the School of Computer Science at Carnegie Mellon University. He leads the TEAMCORE research group at USC (<http://teamcore.usc.edu>), with research interests in multi-agent systems, specifically multi-agent teamwork, adjustable autonomy and distributed negotia-

tions. His research on these topics has led to some of the most highly cited papers in the field, as well as the ACM SIGART Agents Research award (at AAMAS 2005), the Okawa foundation research grant award (2003), AAMAS best paper award (2002), selection in the “best of” papers of Agents’99 and ICMAS’98 conferences, and the RoboCup scientific challenge award (1999). He was general co-chair for the International Joint Conference on Agents and Multiagent Systems (AAMAS) 2004, and program co-chair of the International conf on multi-agent systems (ICMAS) 2000. He is also currently associate editor of the Journal of Artificial Intelligence Research (JAIR), and on the editorial board of the Journal of Autonomous Agents and Multi-agent Systems (JAAMAS). A current member of the board of directors of the International foundation for multiagent systems, he has also served on the board of trustees of RoboCup, the RobotWorld Cup Federation.

Paul Scerri is a Systems Scientist at Carnegie Mellon University’s Robotics Institute. His primary research interests is on coordination of large teams and human interaction with such teams, specifically via the use of adjustable autonomy. He received his PhD from Linkopings University in Sweden in 2001. Before moving to Carnegie Mellon, he worked at the University of Southern California’s Information Sciences Institute.

## 9 Tables

Strategy	$H$			$AH$			$A_T H$		
# 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.

Strategy	4 agents	6 agents	10 agents
<i>AH</i>	34	<b>75</b>	14
<i>A<sub>T</sub>H</i>	54	<b>231</b>	47

Table 2: Task conflicts for subject 2.

## 10 Figure Captions

Figure 1: Model predictions for various users.

Figure 2: DEFACTO system applied to a disaster rescue.

Figure 3: Omni-Viewer during a scenario: (a) Multiple fires start across the USC campus (b) The Incident Commander uses the Navigation mode to quickly grasp the situation (c) Navigation mode shows a closer look at one of the fires (d) Allocation mode is used to assign a fire engine to the fire (e) The fire engine has arrived at the fire (f) The fire has been extinguished.

Figure 4: Proxy Architecture

Figure 5: Performance of subjects 1, 2, and 3.

Figure 6: (a)  $AG_H$  and (b)  $H$  performance.

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

Figure 8: Current Training Methods: (a) projected photo of fire and (b) incident commanders at a table

# 11 Figures

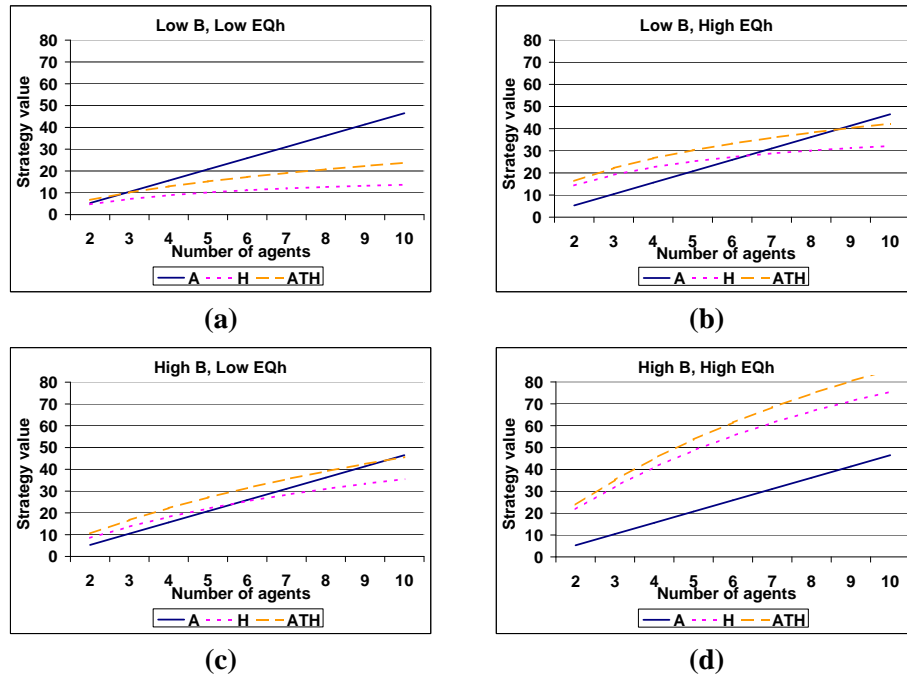


Figure 1:



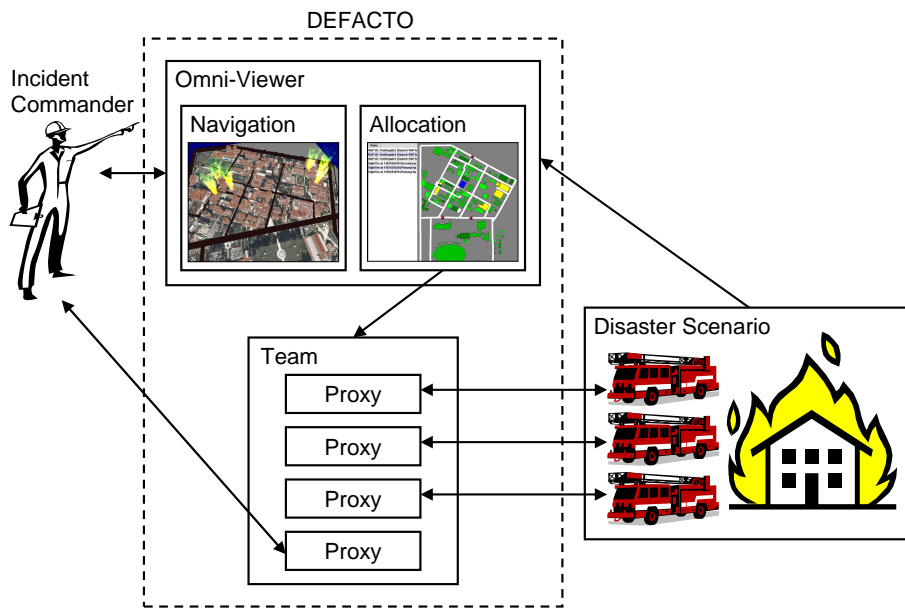


Figure 2:



(a)



(b)



(c)



(d)



(e)



(f)

Figure 3:

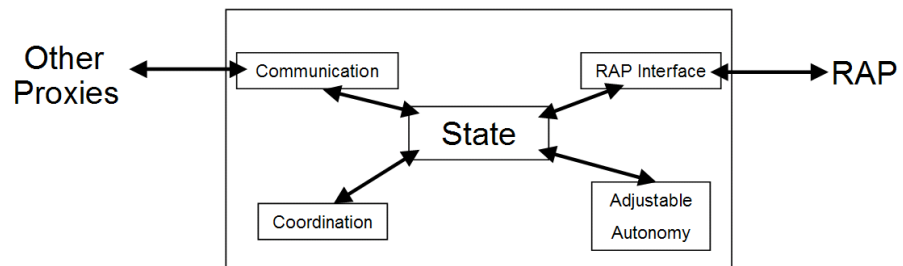


Figure 4:

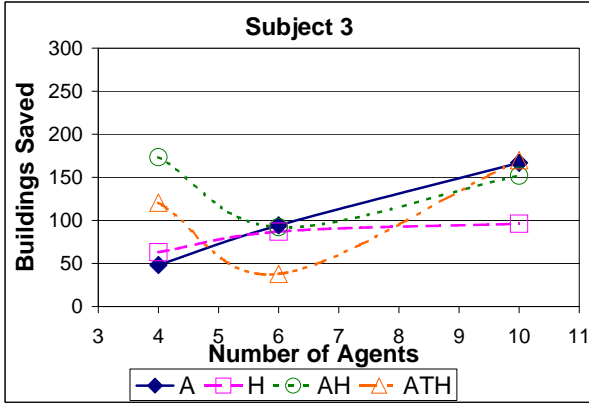
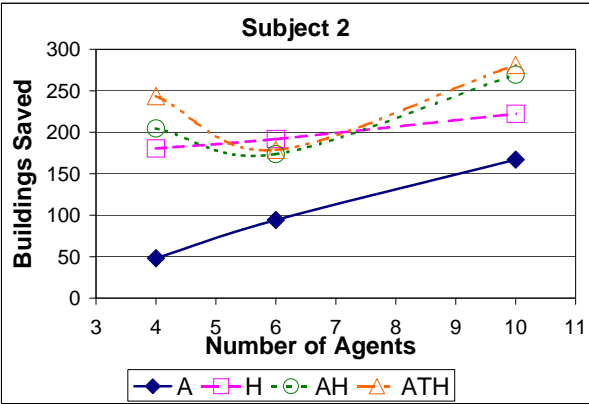
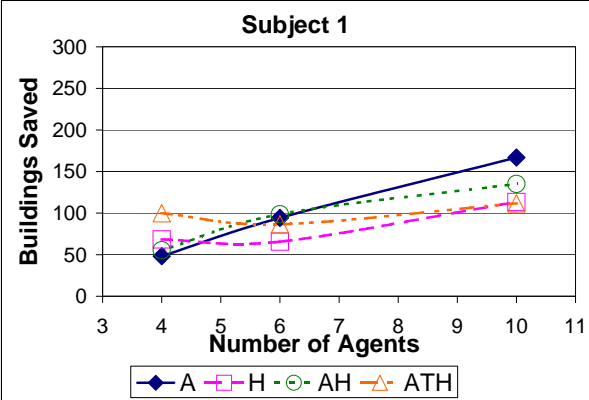
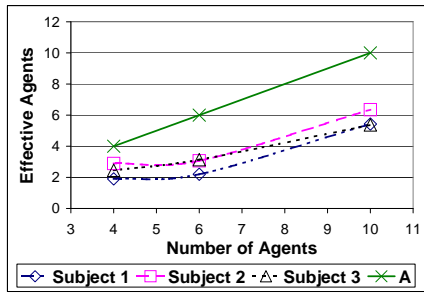
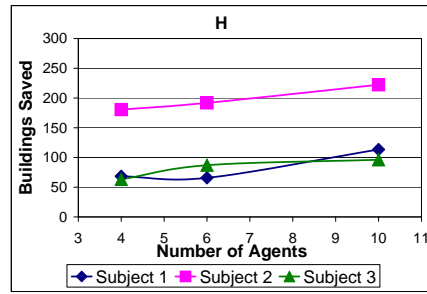


Figure 5:



(a)



(b)

Figure 6:

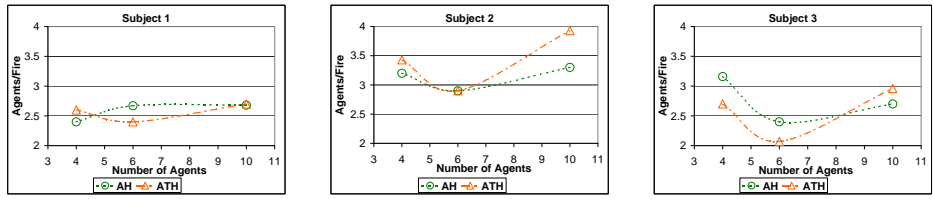


Figure 7:



**(a)**

**(b)**

Figure 8: