Chapter 1

THE DEFACTO SYSTEM: COORDINATING HUMAN-AGENT TEAMS FOR THE FUTURE OF DISASTER RESPONSE*

N. Schurr[1], J.Marecki[1], J.P. Lewis[1], M. Tambe [1] and P. Scerri[2]

[1] *University of Southern California*
Powell Hall of Engineering,
3737 Watt Way, Los Angeles, CA 90089-0781
{schurr, marecki, tambe} @usc.edu, zilla@computer.org

[2] *Carnegie Mellon University,*
5000 Forbes Avenue
Pittsburgh, PA 15213
pscerri@cs.cmu.edu

Abstract Enabling effective interactions between agent teams and humans for disaster response is a critical area of research, with encouraging progress in the past few years. However, previous work suffers from two key limitations: (i) limited human situational awareness, reducing human effectiveness in directing agent teams and (ii) the agent team's rigid interaction strategies that limit team performance. 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 two limitations described 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.

---

*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.

Keywords: Multiagent Systems, Adjustable Autonomy, Teamwork, Disaster Response.

1.1 Introduction

We envision future disaster response to be performed with a mixture of humans performing high level decision-making, intelligent agents coordinating the response and humans and robots performing key physical tasks. These heterogeneous teams of robots, agents, and people [18] will provide the safest and most effective means for quickly responding to a disaster, such as a terrorist attack. A key aspect of such a response will be agent-assisted vehicles working together. Specifically, agents will assist the vehicles in planning routes, determining resources to use and even determining which fire to fight. However, despite advances in agent technologies, human involvement will be crucial. Allowing humans to make critical decisions within a team of intelligent agents or robots is prerequisite for allowing such teams to be used in domains where they can cause physical, financial or psychological harm. These critical decisions include not only the decisions that, for moral or political reasons, humans must be allowed to make, but also coordination decisions that humans are better at making due to access to important global knowledge, general information or support tools.

Already, 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, 6].

This paper focuses on the challenge of improving the effectiveness of applications 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 dialogue [1]. Despite this encouraging progress, previous work suffers from two key limitations. First, when interacting with agent teams acting remotely, human effectiveness is hampered by interfaces that limit their ability to apply decision-making skills in a fast and accurate manner. 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)[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.

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 two shortcomings outlined above. The system incorporates state of the art artificial intelligence, 3D visualization and human-interaction reasoning into a unique high fidelity system for research into human agent coordination in complex environments. DEFACTO incorporates a visualizer that allows for the human to have an *omnipresent* interaction with remote agent teams, overcoming the first limitation described above. 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, [11]), 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, we generalize the notion of *strategies* from single-agent single-human context [17]. 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.

We present results 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 is currently instantiated as a prototype of a future disaster response system. DEFACTO has been repeatedly demonstrated to key police and fire department personnel in Los Angeles area, with very positive feedback.

1.2 DEFACTO System Details

In this chapter we will describe two major components of DEFACTO: the Omni-Viewer and the proxy-based teamwork (see Figure 1.1). The Omni-Viewer is an advanced human interface for interacting with an agent-assisted response effort. The Omni-Viewer provides for both global and local views of an unfolding situation, allowing a human decision-maker to precisely the information required for a particular decision. A team of completely distributed proxies, where each proxy encapsulates advanced coordination reasoning based on the theory of teamwork, controls and coordinates agents in a simulated environment. The use of the proxy-based team brings realistic coordination complexity to the prototype and allows more realistic assessment of the interactions between humans and agent-assisted response. Currently, we have applied DEFACTO to a disaster rescue domain. The incident commander of the disaster acts as the *human 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 our university campus (withheld for blind review), 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 1.1. 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.

1.2.1 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 [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].

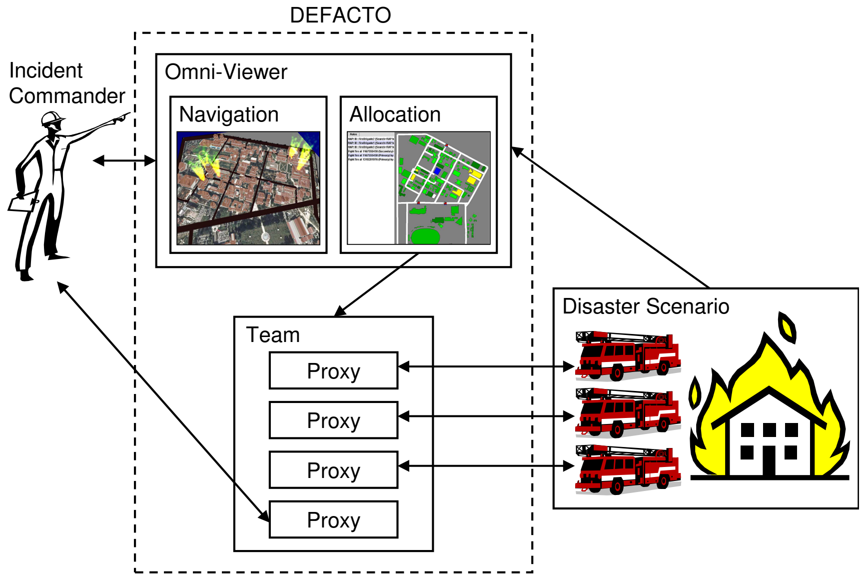


Figure 1.1. DEFACTO system applied to a disaster rescue.

To address our discrepant goals, the Omni-Viewer incorporates both a conventional map-like 2D view, Allocation Mode (Figure 1.2-c) and a detailed 3D viewer, Navigation Mode (Figure 1.2-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 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 video-games. However, to avoid this level of effort we make use of the work of You et. al. [19] 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

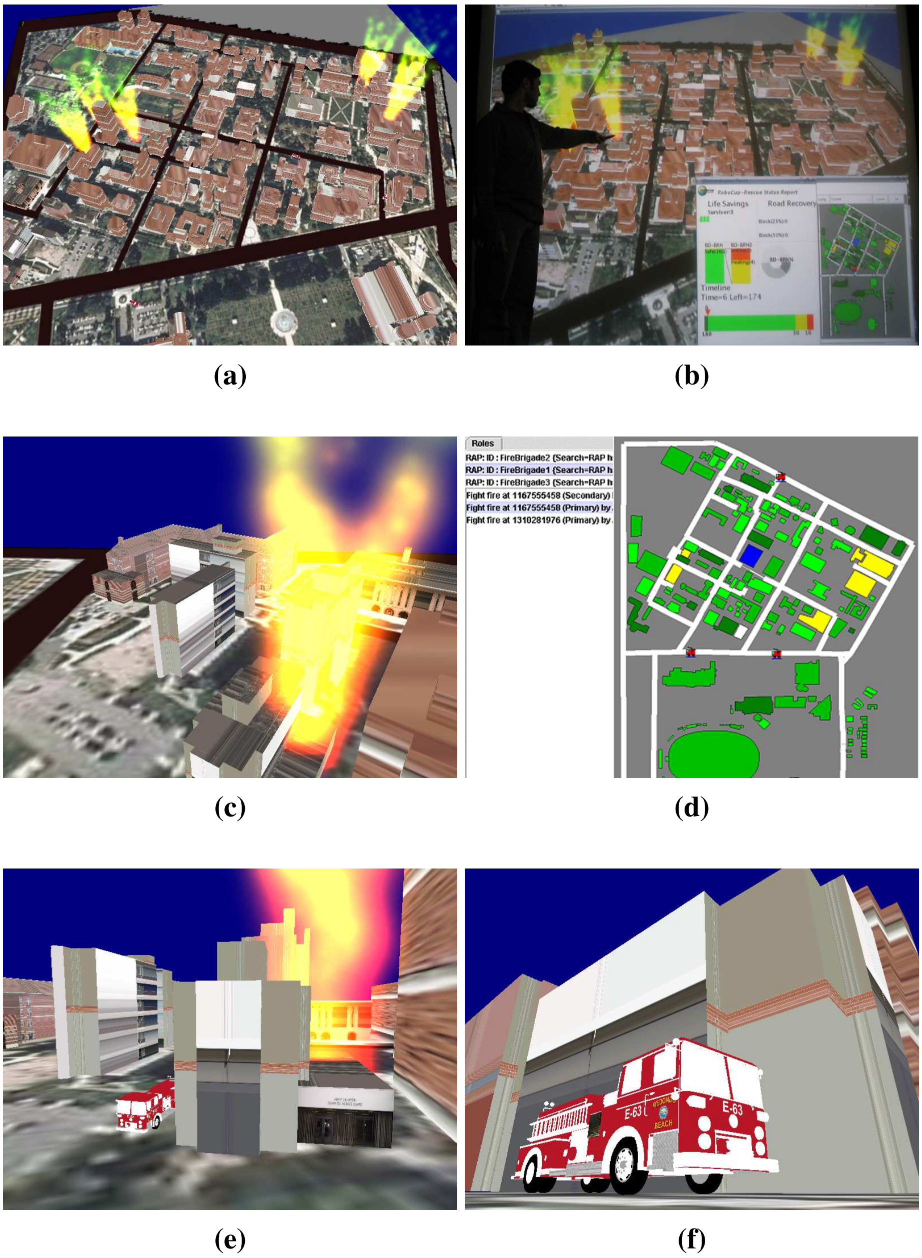


Figure 1.2. Omni-Viewer during a scenario: (a) Multiple fires start across the campus (b) The Incident Commander uses the Navigation mode to quickly grasp the situation (c) Navigation mode shows a closer look at one of the fires (d) Allocation mode is used to assign a fire engine to the fire (e) The fire engine has arrived at the fire (f) The fire has been extinguished.

and create the high resolution model that the Navigation mode utilizes. The construction of the campus (withheld for blind review) 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.

We use the JME game engine to perform the actual rendering due to its cross-platform capabilities. JME is an extensible library built on LWJGL (Light Weight Java Game Library), which interfaces with OpenGL and OpenAL. This environment easily provided real-time rendering of the textured campus environment on mid-range commodity PCs. JME utilizes a scene graph to order the rendering of geometric entities. It provides some important features such as OBJ format model loading (which allows us to author the model and textures in a tool like Maya and load it in JME) and also various assorted effects such as particle systems for fires.

1.2.2 Proxy: Teamwork

A key hypothesis in this work is that intelligent distributed agents will be a key element of a future disaster response. Taking advantage of emerging robust, high bandwidth communication infrastructure we believe that a critical role of these intelligent agents will be to manage coordination between all members of the response team. Specifically, we are using coordination algorithms inspired by theories of teamwork to manage the distributed response [20]. 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 "multiagent 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.

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

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

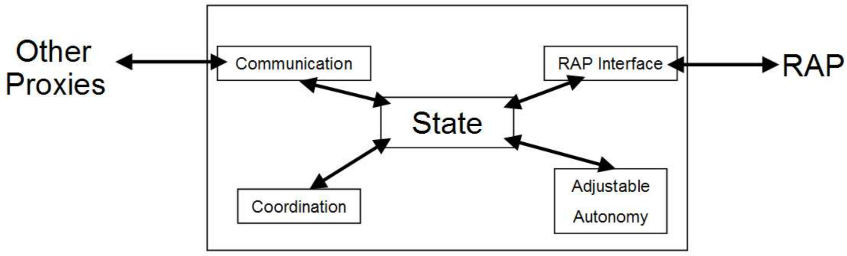


Figure 1.3. Proxy Architecture

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, e.g., new adjustable autonomy algorithms can be used with existing coordination algorithms. The coordination reasoning is responsible for reasoning about interactions with other proxies, thus 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. For example, the adjustable autonomy module can reason that a decision to accept a role to rescue a civilian from a burning building should be made by the human who will go into the building rather than the proxy. In practice, the overwhelming majority of coordination decisions are made by the proxy, with only key decisions referred to team members.

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[]. Typically, 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[9], role allocation[10], information sharing[22], task deconfliction[9] and adjustable autonomy[17]. Many of these algorithms utilize a logical *associates network*[21] statically connecting all the team members. The associates network is a *scale free network* which allows the team to balance the complexity of needing to know about all the team and maintaining cohesion [12]. Using the associates network 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, LA-DCOP [10], 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.

1.2.3 Proxy: Adjustable Autonomy

In this paper, we focus on a key aspect of the proxy-based coordination: 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_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.

1.3 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.

The results of our experiments are shown in Figure 1.4, 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_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 we tested, varied the number of fire engines

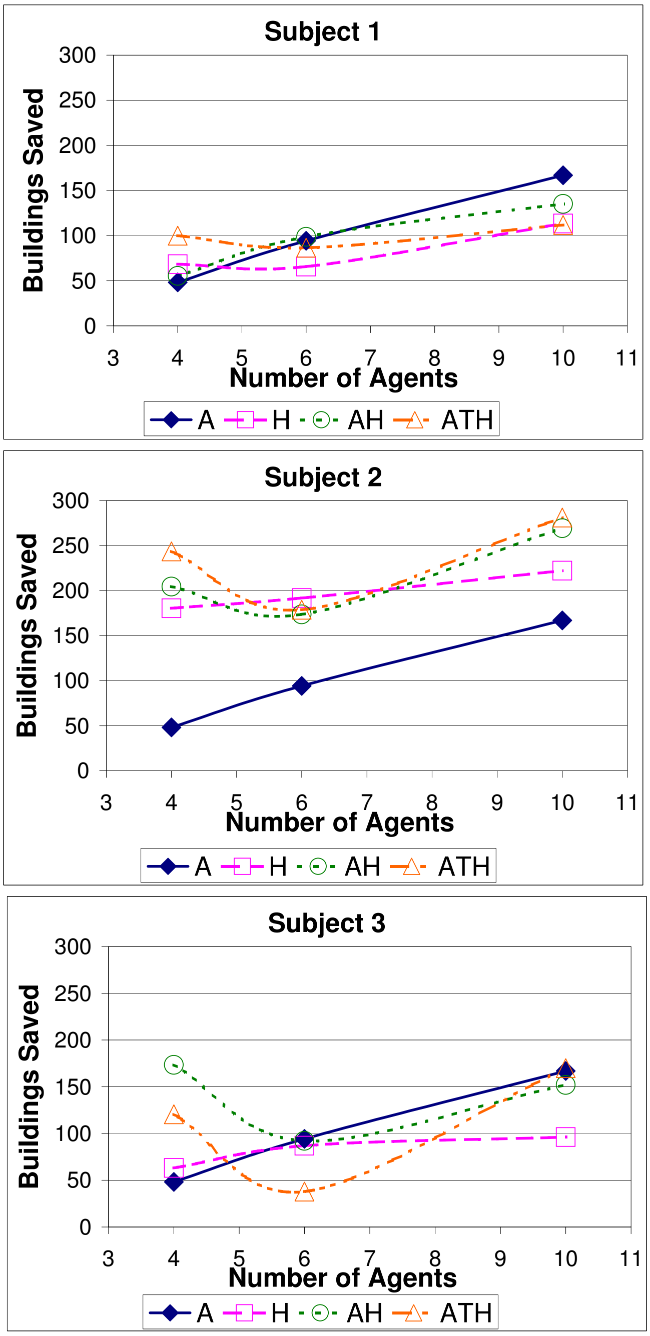


Figure 1.4. Performance of subjects 1, 2, and 3.

between 4, 6 and 10. Each chart in Figure 1.4 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 1.4 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.

- *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.

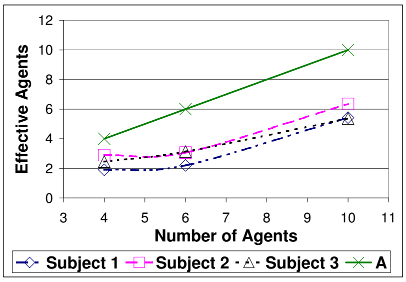
- *No strategy dominates through all the experiments given varying numbers of agents.* For instance, at 4 agents, human-involving strategies dominate the A_T strategy. However, at 10 agents, the A_T strategy outperforms all possible strategies for subjects 1 and 3.

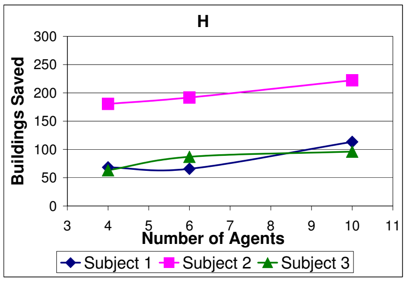
- *Complex team-level strategies are helpful in practice*: A_TH leads to improvement over H with 4 agents for all subjects, although surprising domination of AH over A_TH in some cases indicates that AH may also a useful strategy to have available in a team setting.

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 how to select the appropriate strategy for a team involving a human whose expected decision quality is EQ_H. In fact, 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

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 1.1. Total amount of allocations given.





(a) **(b)**

Figure 1.5. (a) AG_H and (b) H performance

appropriate strategy for teams involving more agents. In general, higher EQ_H lets us still choose strategies involving humans for a more numerous team. For large teams however, the number of agents AG_H effectively controlled by the human does not grow linearly thus A_T strategy becomes dominant.

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 1.4). It would be useful to understand which factors contributed to this phenomena.

Our crucial predictions 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 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 1.5-(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,

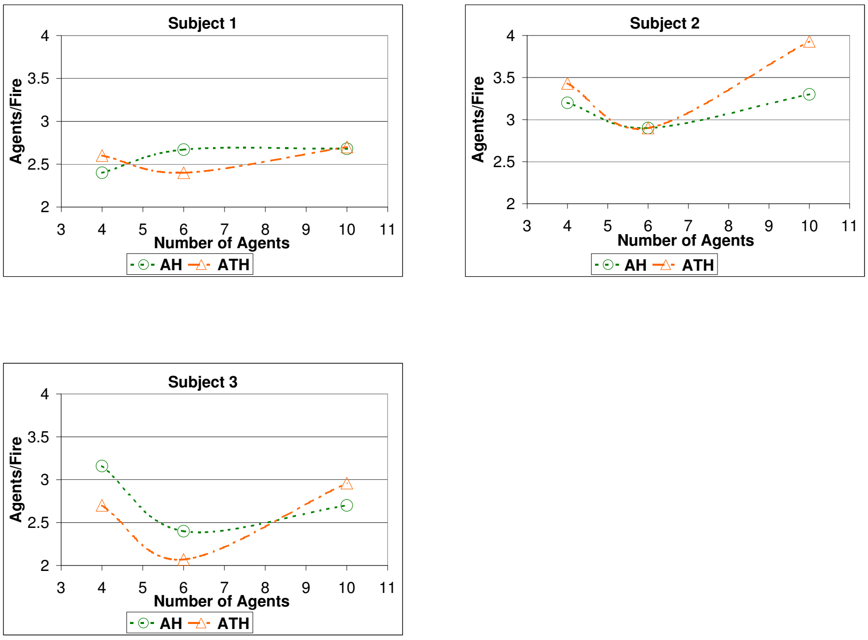


Figure 1.6. Amount of agents per fire assigned by subjects 1, 2, and 3

however is not the case as can be seen in Table 1.1 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 1.5-(b). Figure 1.6 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

Strategy	4 agents	6 agents	10 agents
AH	34	**75**	14
A_TH	54	**231**	47

Table 1.2. Task conflicts for subject 2.

potential to occur in complex team-level strategies. Consequently, modeling strategy predictions would require taking 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 USC 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.

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, we 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."*

1.4 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.

Among the current tools aimed at simulating rescue environments it is important to mention products like TerraSim, JCATS and EPICS. TerraTools is a complete simulation database construction system for automated and rapid generation of high-fidelity 3D simulation databases from cartographic source materials. Developed by TerraSim, Inc. TerraTools provides the set of integrated tools aimed at generating various terrains, however, it cannot simulate rescue operations not it has any notion of intelligence. 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. Although it provides a great human training environment, JCATS does not allow to simulate intelligent 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, intelligent agents and agent-human interaction cannot be simulated.

Given our application domains, Scerri et al's work on robot-agent-person (RAP) teams for disaster rescue is likely the most closely related to DEFACTO [18]. Our work takes a significant step forward in comparison. First, the omniviewer enables navigational capabilities improving human situational awareness not present in previous work. Second, we provide team-level strategies, which we experimentally verify, absent in that work. Third, we provide extensive experimentation, and illustrate that some of the conclusions reached in [18] 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 [2, 19], 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 presents a large-scale prototype, DEFACTO, that is based on a software proxy architecture and 3D visualization system and provides two 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. 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 the appropriate strategy can be selected. Exciting feedback from DEFACTO's ultimate consumers illustrates its promise and potential for real-world application.

References

[1] James 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] Mark H. Burstein, Alice M. Mulvehill, and Stephen Deutsch. An approach to mixed-initiative management of heterogeneous software agent teams. In *HICSS*, page 8055. IEEE Computer Society, 1999.

[3] Jacob W. Crandall, Curtis W. Nielsen, and Michael 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] Terrence Fong, Charles Thorpe, and Charles 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] Eric 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] Hiroaki Kitano, Satoshi Tadokoro, Itsuki Noda, Hitoshi Matsubara, Tomoichi Takahashi, Atsushi Shinjoh, and Susumu 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] E. Liao, P. Scerri, and K. Sycara. A framework for very large teams. In *AAMAS'04 Workshop on Coalitions and Teams*, 2004.

[10] Steven Okamoto. Dcop in la: Relaxed. Master's thesis, University of Southern California, 2003.

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

[12] Regis Vincent Paul Scerri and Roger Mailler. Comparing three approaches to large scale coordination. In *Proceedings of AAMAS'04 Workshop on Challenges in the Coordination of Large Scale MultiAgent Systems*, 2004.

[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. *J. Experimental Psychology - Applied*, 3(2):143–159, 1997.

[16] P. Scerri, E. Liao, Yang. 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 *AAMAS*, 2003.

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

[20] Milind Tambe. Agent architectures for flexible, practical teamwork. *National Conference on AI (AAAI97)*, pages 22–28, 1997.

[21] Duncan Watts and Steven Strogatz. Collective dynamics of small world networks. *Nature*, 393:440–442, 1998.

[22] 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.