The DEFACTO System: Training Tool for Incident Commanders*

Nathan Schurr, Janusz Marecki, J.P. Lewis and Milind Tambe
University of Southern California, PHE 514
3737 Watt Way, Los Angeles, CA 90089-0781
{schurr, marecki, tambe}@usc.edu, zilla@computer.org
phone: 213.740.9560 fax: 213.740.7877

Paul Scerri
Carnegie Mellon University,
5000 Forbes Avenue, Pittsburgh, PA 15213
pscerri@cs.cmu.edu
phone: 412.268.2145 fax: 412.268.6436

Abstract

Techniques for augmenting the automation of routine coordination are rapidly reaching a level of effectiveness where they can simulate realistic coordination on the ground for large numbers of emergency response entities (e.g. fire engines, police cars) for the sake of training. Furthermore, it seems inevitable that future disaster response systems will utilize such technology. We have constructed a new system, DEFACTO (Demonstrating Effective Flexible Agent Coordination of Teams through Omnipresence), that integrates state-of-the-art agent reasoning capabilities and 3D visualization into a unique high fidelity system for training incident commanders. The DEFACTO system achieves this goal via three main components: (i) Omnipresent Viewer - intuitive interface, (ii) Proxy Framework - for team coordination, and (iii) Flexible Interaction - between the incident commander and the team. We have performed detailed preliminary experiments with DEFACTO in the fire-fighting domain. In addition, DEFACTO has been repeatedly demonstrated to key police and fire department personnel in Los Angeles area, with very positive feedback.

Track: Emerging Application

Application Domain: Incident Commander Training Tool for Disaster Response

AI Techniques Employed: Multiagent Coordination, Proxy Framework, Adjustable Autonomy

Application Status: Operational Prototype

Introduction

In the wake of large-scale national and international terrorist incidents, it is critical to provide first responders and rescue personnel with tools and techniques that will enable them to evaluate response readiness and tactics, measure inter-agency coordination and improve training and decision making capability. We focus in particular on building tools for training and tactics evaluation for incident commanders, who are in charge of managing teams of fire fighters at critical incidents. Such tools would provide intelligent software agents that simulate first responder tactics, decisions, and behaviors in simulated urban areas and allow the incident commander (human) to interact. These agents form teams, where each agent simulates a fire engine, which plans and acts autonomously in a simulated environment. Through interactions with these software agents, an incident commander can evaluate tactics and realize the consequences of key decisions, while responding to such disasters.

To this end, we have constructed a new system, DEFACTO (Demonstrating Effective Flexible Agent Coordination of Teams through Omnipresence). DEFACTO incorporates state of the art artificial intelligence, 3D visualization and human-interaction reasoning into a unique high fidelity system for training incident commanders. By providing the incident commander interaction with the coordinating agent team in a complex environment, the commander can gain experience and draw valuable lessons that will be applicable in the real world. The DEFACTO system achieves this via three main components: (i) Omnipresent Viewer - intuitive interface, (ii) Proxy Framework - for team coordination, and (iii) Flexible Interaction - between the incident commander and the team.

Omnipresent Viewer: As sensor capabilities and networks improve, command and control centers will have access to very large amounts of up-to-date information about the environment. A key open question is how that information can be used to allow most effective incident command decision-making. We are exploring one promising option that leverages advances in 3D visualization and virtual reality: a viewer for virtual omnipresence.

Proxy Framework: Techniques for automation of routine coordination are rapidly reaching a level of effectiveness and maturity where it seems inevitable that future disaster response systems will utilize such technology to manage a coordination response. Such automation will have a dramatic, though at present unclear, impact on the role and nature of humans in the response. To examine the impact of such technology, we have included state-of-the-art autonomous coordination into the heart of the DEFACTO system. This allows other aspects of the system to be developed with a realistic representation of how the coordination will proceed. Conversely, it can assist developers of coordination technology to identify the key problems for real-world deployment of their technology.

*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.

Flexible Interaction: Previous work has shown that in a multiagent context, flexible interactions between autonomous agents and humans are critical to the success of the system. While intuitive human interfaces and intelligent team coordination are important, critical failures will occur unless human-agent interaction is appropriately managed. Hence, we are developing new, flexible techniques for managing this interaction. These techniques build on theories of teamwork to manage the distributed response (Scerri, Pynadath, & Tambe 2002).

Disaster response provides a dynamic world in which decisions must be made correctly and quickly because human safety is at risk. However, it also holds potential for agents to take on responsibility in order to free the humans involved from being unnecessarily overburdened in a time of crisis. When using DEFACTO, incident commanders have the opportunity to see the disaster and the coordination/resource constraints unfold so that they can be better prepared when commanding over an actual disaster. Applying DEFACTO to disaster response holds a great potential benefit to the training of incident commanders in the fire department. DEFACTO has been repeatedly demonstrated to key police and fire department personnel in Los Angeles area with very positive feedback.

In addition, we have performed detailed preliminary experiments with DEFACTO being used as a training tool in the firefighting domain. We present results from these experiments that reveal results pointing to the value of using such tools. First, contrary to previous results (Scerri *et al.* 2003), 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 degrade the team performance, even though increasing the number of agents in a pure agent team under identical circumstances improves team performance. These results illustrate that DEFACTO can create a challenging environment for incident commanders, and that there is a need for the flexible interaction strategies between teams and incident commanders incorporated into DEFACTO (which domain experts verify is a requirement in this domain (LAFD 2001)).

With DEFACTO, our objective is to both enable the human to have a clear idea of the team's state and improve agent-human team performance. We want DEFACTO agent-human teams to better prepare firefighters for current human-only teams and better prepare them for the future, in which agent-human teams will be inevitable and invaluable. We believe that leveraging the above aspects DEFACTO will result in better disaster response methods and better incident commanders.

System Overview

First, DEFACTO incorporates a visualizer that allows for the human to have an *omnipresent* interaction with remote agent teams. We refer to this as the Omni-Viewer (see Figure 1), and it combines two modes of operation. The Navigation Mode allows for a navigable, high quality 3D visualization

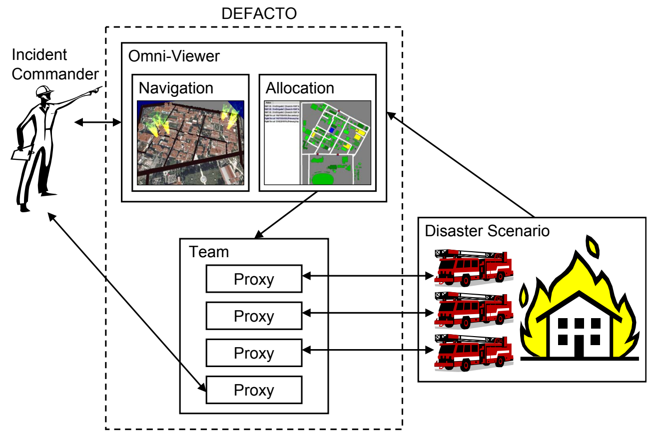


Figure 1: DEFACTO system architecture.

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, (Paivio 1974)), 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, DEFACTO utilizes a proxy framework that handles both the coordination and communication for the human-agent team (see Figure 1). We use coordination algorithms inspired by theories of teamwork to manage the distributed response (Scerri, Pynadath, & Tambe 2002). The general coordination algorithms are encapsulated in *proxies*, with each team member having its own proxy and representing it in the team.

Third, we also use the above mentioned proxy framework to implement adjustable autonomy (AA) between each member of the human-agent team. The novel aspect of DEFACTO's flexible AA is that we generalize the notion of *strategies* from single-agent single-human context to agent-team single-human. 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.

Domain

The DEFACTO system is currently focused on illustrating the potential of future disaster-response to disasters that may arise as a result of large-scale terrorist attacks. Constructed as part of the effort at the first center for research excellence on homeland security (the CREATE center), DEFACTO is motivated by a scenario of great concern to first responders within Los Angeles and other metropolitan areas. In our consultations with the Los Angeles fire department and per-

(a) AG_H (b) H

Figure 2: Current Training Methods: (a) projected photo and (b) incident commanders at a table

sonnel from the CREATE center, this scenario is of great concern. In particular, a shoulder-fired missile could potentially be used to attack a low-flying civilian jet-liner that is preparing to land at Los Angeles International Airport causing the jet-liner to crash into an urban area and resulting in a large-scale disaster on the ground. This scenario could lead to multiple fires in multiple locations with potentially many critically injured civilians. While there are many longer-term implications of such an attack, we focus on assisting first responders, namely fire fighters.

We had the opportunity to study the current methods that the Los Angeles Fire Department (LAFD) implements to train incident commanders. The LAFD uses a projection screen to simulate the disaster (Figure 2-(a)). In addition, the participating incident commander is seated at a desk, directing an assistant to take notes (Figure 2-(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. DEFACTO aims to automate this process and enable higher fidelity and larger scale training scenarions which require less personnel.

Underlying Technologies

In this section, we will describe the technologies used in three major components of DEFACTO: the Omni-Viewer, proxy-based team coordination, and proxy-based adjustable autonomy. 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 obtain 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. These same proxies also enable us to implement the adjustable autonomy necessary to balance the decisions of the agents and human.

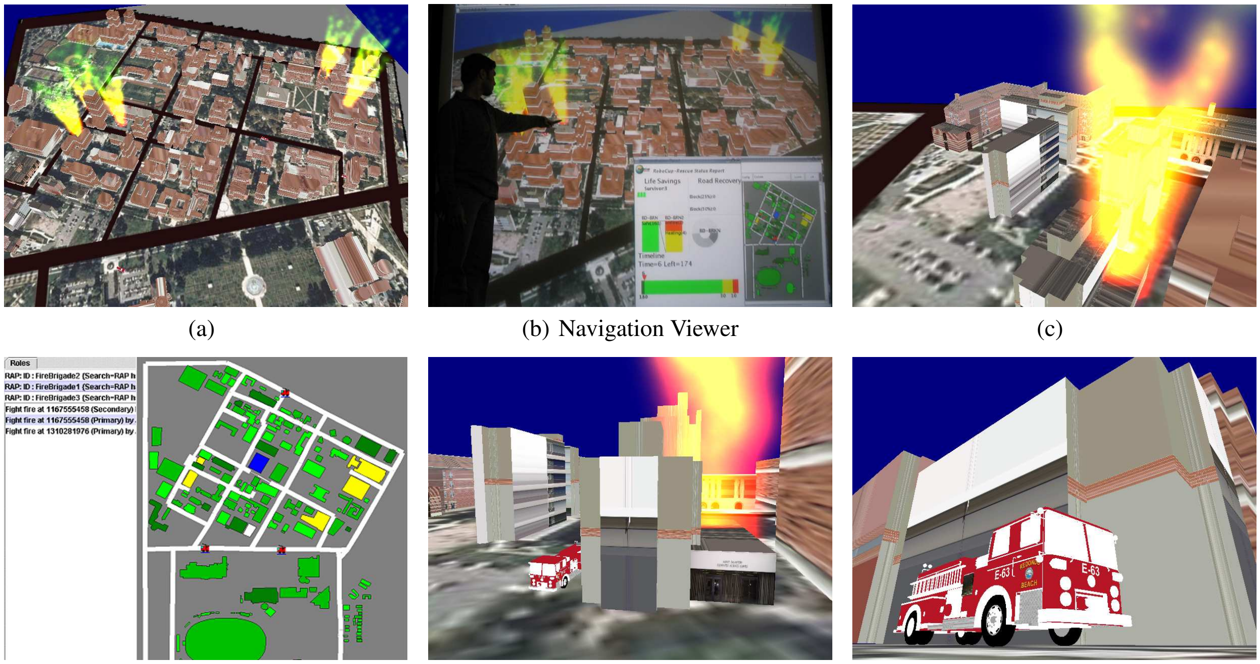
We have applied DEFACTO to a disaster rescue domain, where the incident commander of the disaster acts as the *human user* of DEFACTO. We focus on two urban areas: a square block that is densely covered with buildings (we use one from Kobe, Japan) and the University of Southern California (USC) campus, which is more sparsely covered with buildings. In our scenario, several buildings are initially on fire, and these fires spread to adjacent buildings if they are not quickly contained. The goal is to have a human interact with the team of fire engines in order to save the most buildings. Our overall system architecture applied to disaster response can be seen in Figure 1.

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 shows 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 (Richardson, Montello, & Hegarty 1999). 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 (Ruddle, Payne, & Jones 1997).

To address our 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-b). 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. (Suya You & Fox 2003) in rapid, minimally assisted construction of polygonal models



(a) (b) Navigation Viewer (c)

(d) Allocation Viewer (e) (f)

Figure 3: 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.

from LiDAR (Light Detection and Ranging) data. Given the raw LiDAR point data, we can automatically segment buildings from ground and create the high resolution model that the Navigation mode utilizes. The construction of the USC campus and surrounding area required only two days using this approach. LiDAR is an effective way for any new geographic area to be easily inserted into the Omni-Viewer.

Proxy: Team Coordination

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 (Scerri, Pynadath, & Tambe 2002). 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* (Scerri *et al.* 2003) and extends the successful Teamcore proxies (Pynadath & Tambe 2003). 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.

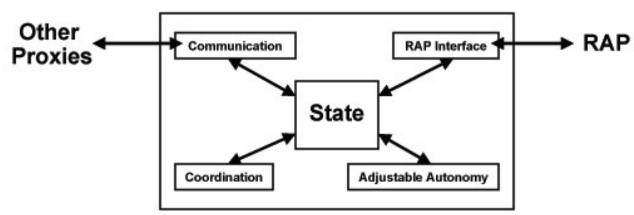


Figure 4: Proxy Architecture

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 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.

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 a token-based role allocation algorithm. The decision for the agent becomes whether to assign values represented by tokens it currently has to its variable or to pass the tokens on. First, the team member can choose the minimum capability the agent should have in order to assign the value. This minimum capability is referred to as the *threshold*. The threshold is calculated once (Algorithm 1, line 6), and attached to the token as it moves around the team.

Second, the agent must check whether the value can be assigned while respecting its local resource constraints (Algorithm 1, line 9). If the value cannot be assigned within the resource constraints of the team member, it must choose a value(s) to reject and pass on to other teammates in the form of a token(s) (Algorithm 1, line 12). The agent keeps values that maximize the use of its capabilities (performed in the MAXCAP function, Algorithm 1, line 10).

Algorithm 1 TOKENMONITOR$(Cap, Resources)$

1: $V \leftarrow \emptyset$
2: **while** true **do**
3:   $msg \leftarrow getMsg()$
4:   $token \leftarrow msg$
5:   **if** $token.threshold = NULL$ **then**
6:     $token.threshold \leftarrow$ COMPUTETHRESHOLD$(token)$
7:   **if** $token.threshold \leq Cap(token.value)$ **then**
8:     $V \leftarrow V \cup token.value$
9:     **if** $\sum_{v \in V} Resources(v) \geq agent.resources$ **then**
10:       $out \leftarrow V-$ MAXCAP$(Values)$
11:       **for all** $v \in out$ **do**
12:         PASSON$(newtoken(v))$
13:         $Values \leftarrow Values - out$
14:   **else**
15:     PASSON$(token)$ /* $threshold > Cap(token.value)$ */

Proxy: Adjustable Autonomy

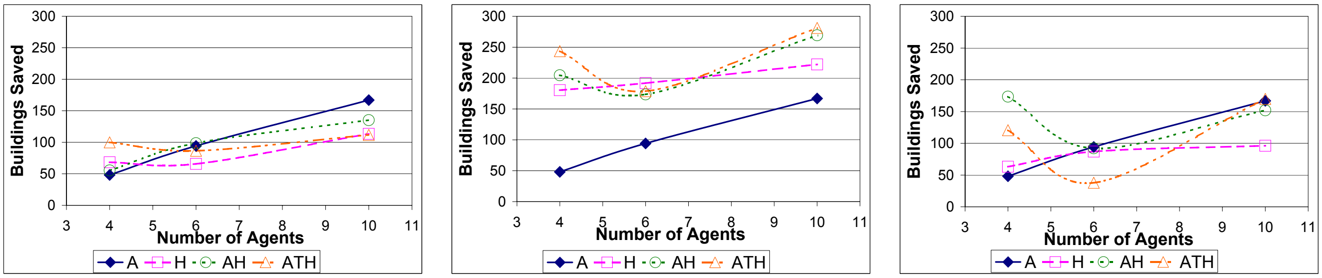
In this paper, we focus on a key aspect of the proxy-based coordination: Adjustable Autonomy. Adjustable autonomy refers to an agent's ability to dynamically change its own autonomy, possibly to transfer control over a decision to a human. Previous work on adjustable autonomy could be categorized as 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 (Scerri, Pynadath, & Tambe 2002). 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.

Experiments and Evaluation

Our DEFACTO system was evaluated through experiments comparing the effectiveness of Adjustable Autonomy (AA) strategies over multiple users. In order to provide DEFACTO with a dynamic rescue domain, we connected it to



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

Figure 5: Performance.

the previously developed RoboCup Rescue simulation environment (Kitano *et al.* 1999). 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 (incident commander). 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 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, we tested three strategies, specifically, H, AH and $A_T H$; their performance was compared with the completely autonomous A_T strategy. AH is an individual agent strategy, tested for comparison with $A_T H$, where agents act individually, and pass those tasks to a human user that they cannot immediately perform. Each experiment was conducted with the same initial locations of fires and building damage. For each strategy we tested, varied the number of fire engines between 4, 6 and 10. Each chart in Figure 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:

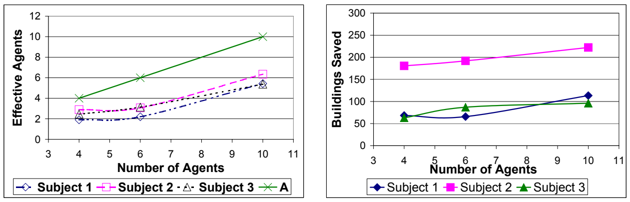
- *Human involvement with agent teams does not necessarily lead to improvement in team performance.* Contrary to expectations and prior results, human involvement does not uniformly improve team performance, as seen by human-involving strategies performing worse than the A_T strategy in some cases. For instance, for subject 3 AH strategy provides higher team performance than A_T for 4 agents, yet at 10 agents human influence is clearly not beneficial.
- *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 $A_T H$ 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 $A_T H$ 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*: $A_T H$ leads to improvement 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 need 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, then, 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_T strategy, we may begin to select the 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, and thus A_T strategy becomes dominant.

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). It

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.



(a) (b)

Figure 6: Agents effectively used (AG_H) by the subjects (a) and Performance of strategy H. (b)

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 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 we see in the performance. We then shifted our attention to the EQ_H of each subject. One reduction in EQ_H could be because subjects simply did not send as many allocations totally over the course of the experiments. This, however is not the case as can be seen in Table 1 where for 6 agents, as 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 6-(b). Figure 7 shows the number agents on the x-axis and the average amount of fire engines allocated to each fire on the y-axis. AH and $A_T H$ for 6 agents result in significantly less average fire engines per task (fire) and therefore lower average EQ_H.

Related Work

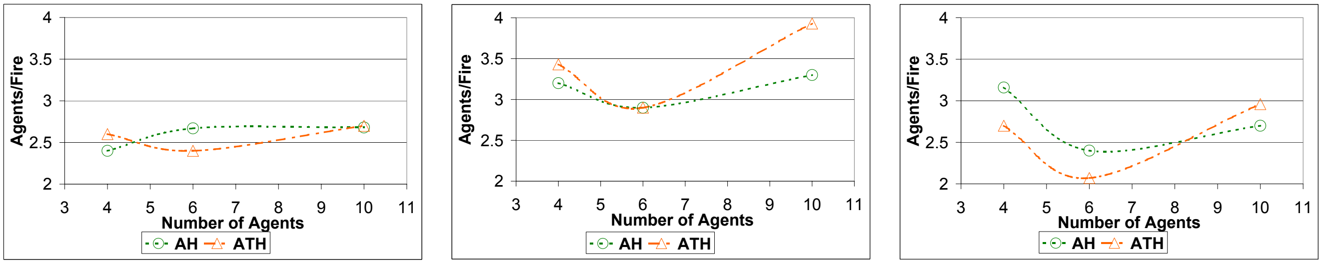
We have discussed related work throughout this paper. However, we now provide comparisons with key previous agent software prototypes and research. Hill et al's work is a similar immersive training tool (Swartout *et al.* 2001). However, their work focused more on multi-modal dialogue and emphasize single agent interaction along predefined story lines, whereas our work focuses on adjustable autonomy and coordinating large numbers of agents in a dynamic, complex fire-fighting domain.

Among the current tools aimed at simulating rescue environments, it is important to mention products like JCATS (Technology 2005) and EPICS (Laboratory 2005). 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.

Given our application domains, Scerri et al's work on robot-agent-person (RAP) teams for disaster rescue (Scerri *et al.* 2003) is closely related to DEFACTO. 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 (Scerri *et al.* 2003) 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 has also been investigated in (Burstein, Mulvehill, & Deutsch 1999; Suya You & Fox 2003), and there is significant research on human interactions with robot-teams (Fong, Thorpe, & Baur 2002; Crandall, Nielsen, & Goodrich 2003). 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 (Horvitz 1999; Allen 1995). However, this paper focuses on new phenomena that arise in human interactions with agent teams.

Summary

This paper presents an operational prototype training tool for incident commanders: DEFACTO. DEFACTO incorporates state of the art artificial intelligence, 3D visualization and human-interaction reasoning into a unique high fidelity system for training incident commanders. The



(a) Subject 1

(b) Subject 2

(c) Subject 3

Figure 7: Amount of agents assigned per fire.

DEFACTO system achieves this via three main components: (i)Omnipresent Viewer - intuitive interface, (ii)Proxy Framework - for team coordination, and (iii) Flexible Interaction - between the incident commander and the team. We present results preliminary experiments with DEFACTO in the firefighting domain. These results illustrate that an agent team must be equipped with flexible strategies for adjustable autonomy so that the appropriate strategy can be selected. Positive feedback from DEFACTO's ultimate consumers, LAFD, illustrates its promise and potential for real-world application.

Acknowledgments

We would like to thank the Los Angeles Fire Department for explaining to us the incident commander training procedures and their useful comments on the DEFACTO system.

References

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

Burstein, M. H.; Mulvehill, A. M.; and Deutsch, S. 1999. An approach to mixed-initiative management of heterogeneous software agent teams. In *Hawaii International Conference on System Sciences*, 8055. IEEE Computer Society.

Crandall, J. W.; Nielsen, C. W.; and Goodrich, M. A. 2003. Towards predicting robot team performance. In *IEEE International Conference on Systems, Man and Cybernetics (SMC)*.

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

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

Kitano, H.; Tadokoro, S.; Noda, I.; Matsubara, H.; Takahashi, T.; Shinjoh, A.; and Shimada, S. 1999. Robocup rescue: Search and rescue in large-scale disasters as a domain for autonomous agents research. In *IEEE International Conference on Systems, Man and Cybernetics (SMC)*, volume VI, 739–743.

Laboratory, L. L. N. 2005. Jcats - joint conflict and tactical simulation. In *http://www.jfcom.mil/about/fact_jcats.htm*.

LAFD. 2001. High-rise incident command system. In *Los Angeles Fire Department*.

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

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

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

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

Scerri, P.; Pynadath, D. V.; Johnson, L.; Rosenbloom, P.; Schurr, N.; Si, M.; and Tambe, M. 2003. A prototype infrastructure for distributed robot-agent-person teams. In *International Joint Conference on Autonomous Agents and Multi-Agent Systems (AAMAS)*.

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

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

Swartout, W., R. H.; Gratch, J.; Johnson, W. L.; Kyriakakis, C.; LaBore, C.; Lindheim, R.; Marsella, S.; Miraglia, D.; Moore, B.; Morie, J.; Rickel, J.; Thiebaux, M.; Tuch, L.; Whitney, R.; and Douglas, J. 2001. Toward the holodeck: integrating graphics, sound, character and story. In *Proceedings of the Fifth International Conference on Autonomous Agents*, 409–416. ACM Press.

Technology, A. S. 2005. Epics - emergency preparedness incident commander simulation. In *http://epics.astcorp.com*.