Using Multiagent Teams to Improve the Training of Incident Commanders*

Nathan Schurr, Pratik Patil, Fred Pighin, Milind Tambe,
University of Southern California, Los Angeles, CA 90089, {schurr, pratiksp, pighin, tambe}@usc.edu

ABSTRACT

The DEFACTO system is a multiagent based tool for training incident commanders for large scale disasters. In this paper, we highlight some of the lessons that we have learned from our interaction with the Los Angeles Fire Department (LAFD) and how they have affected the way that we continued the design of our training system. These lessons were gleaned from LAFD feedback and initial training exercises and they include: system design, visualization, improving trainee situational awareness, adjusting training level of difficulty and situation scale. We have taken these lessons and used them to improve the DEFACTO system's training capabilities. We have conducted initial training exercises to illustrate the utility of the system in terms of providing useful feedback to the trainee.

1. INTRODUCTION

The recent hurricanes that have hit the gulf coast of the US have served to reaffirm the need for emergency response agencies to be better prepared for large scale disasters. Both natural and man-made (terrorism) disasters are growing in scale, however the response to these incidents continues to be managed by a single person, namely the incident commander. The incident commander must monitor and direct the entire event while maintaining complete responsibility. Because of this, incident commanders must start to be trained to handle these large scale events and assist in the coordination of the responding team.

In order to fulfill this need and leverage the advantages of multiagents, we have continued to develop the DEFACTO system (Demonstrating Effective Flexible Agent Coordination of Teams via Omnipresence). DEFACTO is a multiagent based tool for training incident commanders for large scale disasters (man-made or natural).

Our system combines a high fidelity simulator, a redesigned human interface, and a multiagent team driving all of the behaviors. Training incident commanders provides a dynamic scenario in which decisions must be made correctly and quickly because human safety is at risk. When using DEFACTO, incident commanders have the opportunity to see the disaster in simulation and the coordination and resource constraints unfold so that they can be better prepared when commanding over an actual disaster. Applying DEFACTO to disaster response aims to benefit the training of incident commanders in the fire department.

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. We believe that by leveraging multiagents, DEFACTO will result in better disaster response methods and better incident commanders.

Previously, we have discussed building our initial prototype system, DEFACTO [8]. Recently, the Los Angeles Fire Department (LAFD) have begun to evaluate the DEFACTO system. In this paper, we highlight some of the lessons that we have learned from our interaction with the LAFD and how they have affected the way that we continued to design of our training system. These lessons were gleaned from LAFD feedback and initial training exercises.

The lessons learned from the feedback from the LAFD include: system design, visualization, improving trainee situational awareness, adjusting training level of difficulty and situation scale. We have taken these lessons and used them to improve the DEFACTO system's training capabilities.

We have also performed initial training exercise experiments to illustrate the utility of the system in terms of providing useful feedback to the trainee. We ended up finding that allowing more fire engines to be at the disposal of the incident commander sometimes not only didn't improve, but rather worsened team performance. There were even some instances in which the agent team would have performed better had the team never listened to human advice at all. We also provide analysis of such behaviors, thereby illustrating the utility of DEFACTO resulting from the feedback given to trainees.

2. MOTIVATION

In this section, we will first start with an explanation of the current methods for training that the LAFD currently use. Then we explain some of the advantages that our multiagent approach has over these methods.

The incident commander's main duties during a fire shoulder all responsibility for the safety of the firefighters. In order to do this, the incident commander must have constant contact with the firefighters and have a complete picture of the entire situation. The incident commander must make certain that dangerous choices are

*This research was supported by the United States Department of Homeland Security through the Center for Risk and Economic Analysis of Terrorism Events (CREATE) under grant number N00014-05-0630. However, any opinions, findings, and conclusions or recommendations in this document are those of the authors and do not necessarily reflect views of the United States Department of Homeland Security.

(a) Current Incident Commander Training Exercise

(b) Fire Captain Roemer using the DEFACTO training system

Figure 1: Old vs. New training methods

avoided and the firefighters are informed and directed as needed.

We were allowed to observe a Command Post Exercise that simulated the place where the incident commander is stationed during a fire (see Figure 1(a)). The Incident commander has an assistant by his side who keeps track on a large sheet of paper where all of the resources (personnel and equipment) are located. A sketch of the fire is also made on this sheet, and the fire and fire engines' location is also managed.

The Command Post is currently simulated by projecting a single static image of a fire in an apartment. In the back of the room, several firefighters are taken off duty in order to play the role of firefighters on the scene. They each communicate on separate channels over walkie talkies in order to coordinate by sharing information and accepting orders. The fire spreading is simulated solely by having one of the off-duty firefighters in the back speaking over the walkie talkie and describing the fire spreading.

The LAFD's current approach, however, has several limitations. First, it requires a number of officers to be taken off duty, which decreases the number of resources available to the city for a disaster during training. Second, the disaster conditions created are not accurate in the way that they appear or progress. Since the image that the incident commander is seeing is static, there is no information about state or conditions of the fire that can be ascertained from watching it, which is contrary to the actual scene of a disaster response. Furthermore, the fire's behavior is determined by the reports of the acting fire fighters over the walkie talkie, which at times might not be a plausible progression of fire in reality. Third, this method of training restricts it to a smaller scale of fire because of the limited personnel and rigid fire representation.

Our system aims to enhance the training of the incident commanders (see Figure 1(b)). Our approach allows for training to not be so personnel heavy, because fire fighter actors will be replaced by agents. By doing this we can start to train incident commanders with a larger team. Through our simulation, we can also start to simulate larger events in order to push the greater number of available resources to their limit. Also, by simulating the fire progression, we can place the Incident commander in a more realistic situation and force them to react to realistic challenges that arise.

3. SYSTEM ARCHITECTURE

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 has been introduced before [8], however has since been redesigned after incorporating lessons learned from the LAFD. The Omni-Viewer now 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 training system and allows a 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. This architecture has been described in a more extended fashion in [8]; we present a brief report here.

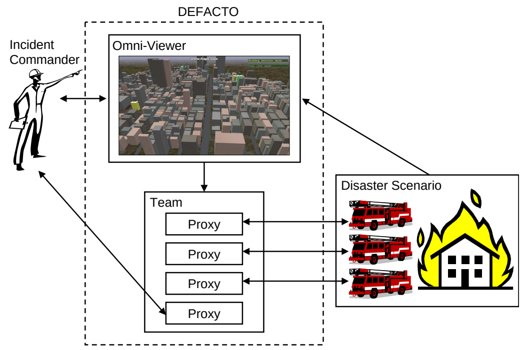


Figure 2: System Architecture

DEFACTO operates in a disaster response simulation environment. The simulation environment itself is provided by the RoboCup Rescue Simulator [3]. To interface with DEFACTO, each fire engine is controlled by a proxy in order to handle the coordination and execution of adjustable autonomy 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 to allocate engines to the fires that he has control over. 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 greatest number of buildings. Our overall system architecture applied to disaster response can be seen in Figure 2.

3.1 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.

To address our discrepant goals, the Omni-Viewer allows for both a conventional map-like top down 2D view and a detailed 3D viewer. The viewer shows the global overview as events are progressing and provides a list of tasks that the agents have transferred to the human, but also provides the freedom to move to desired locations and views. In particular, the user can drop to the virtual ground level, thereby obtaining the perspective (local view) of a particular agent. At this level, the user can fly 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.

3.2 Proxy: Team Coordination

A key hypothesis in this work is that intelligent distributed agents will be a key element of a 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 [6]. The general coordination algorithms are encapsulated in *proxies*, with each team member having its own proxy which represents it in the team. The current version of the proxies is called *Machinetta* [7] and extends the earlier Teamcore proxies [5]. 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. These proxies and their architecture have been discussed in detail in [8].

3.3 Proxy: Adjustable Autonomy

One key aspect of the proxy-based coordination is "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 [6]. 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 has introduced the notion of team-level adjustable autonomy 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. LESSONS LEARNED FROM INITIAL DEPLOYMENT FEEDBACK

Through our communication with strategic training division of the LAFD (see Figure 1(b)), we have learned a lot of lessons that have influenced the continuing development of our system.

4.1 Adjustable Autonomy in Practice

Our most important lesson learned from talking with the LAFD and seeing their exercises is that adjustable autonomy correctly maps over to what happens in the actual disaster response. The adjusting of autonomy is easily seen as the event scales up and down in size and intensity. For a smaller scale response to, for example, a residential single story house fire, the incident commander will usually make all allocation decisions and thus practice the A strategy. For a larger scale event, a lot of the burden of most allocations are left to the team and other entities in the hierarchy, while the Incident Commander is left to concentrate on the bigger picture. In this case, the Incident commander is notified if a specifically problematic situation, for example not enough resources to attack a particular fire. This strategy is essentially what we refer to as A_TH in our experiments, in which, the team first tries to assign someone to the fire with the resources they have, and if not able to then pass it off to the Incident Commander for help. As the situation were to die down and the size of the team were to decrease, more autonomy would be shifted to the Incident Commander due to an increased ability to make allocations for the team.

It is very helpful to know that these strategies not only are capable of making our agent teams perform well and interface with the Incident Commander, but that they also reflect similar strategies that current firefighting teams are using.

Figure 3: Selecting for closer look at a Fire Engine.

4.2 Questioning the Incident Commander

Another lesson that relates to our agent design is that we learned how a team on the ground may possibly not agree with the command (allocation to a fire) given by the Incident Commander. This will usually be due to the fact that the Incident Commander has a broad global view of the disaster, whereas the agents each have a more detailed local view. This mismatch in information can, at times, lead to detrimental team allocations. In an actual disaster response, this is handled by the allocated team both questioning the order and providing the Incident Commander with the missing information.

This has led us to consider a team of agents that can disagree with human inputs. This issue has not been addressed in our implementation as of yet, but it is relevant given the results that will be presented later in the training exercise experiments. There are experimental settings in which the team performance would have been improved, had they rejected the Incident Commander's input.

4.3 Perspective

Just as in multiagent systems, the Incident commander must overcome the challenge of managing a team that each possess only a partial local view. This is highlighted in fighting a fire by incident commanders keeping in mind that there are five views to every fire (4 sides and the top). Only by taking into account what is happening on all five sides of the fire, can the fire company make an effective decision on how many people to send where. Because of this, a local view (see Figure 4(a)) can augment the global view (see Figure 4(b)) becomes helpful in determining the local perspectives of team members. For example, by taking the perspective of a fire company in the back of the building, the incident commander can be aware that they might not see the smoke from the second floor, which is only visible from the front of the building. The incident commander can then make a decision to communicate that to the fire company or make an allocation accordingly.

The 3D perspective of the Omni-Viewer was initially thought to be an example of a futuristic vision of the actual view given to the incident commander. But after allowing the fire fighters to look at the display, they remarked, that they have such views available to them already, especially in large scale fires (the very fires we are trying to simulate). At the scene of these fires are often a news helicopter is at the scene and the incident commander can patch into the feed and display it at his command post. Consequently our training simulation can already start to prepare the Incident Commander to incorporate a diverse array of information sources.

4.4 Fire Behavior

We also learned how important smoke and fire behavior is to the firefighters in order to affect their decisions. Upon our first showing of initial prototypes to the Incident Commanders, they looked at our simulation, with flames swirling up out of the roof (see Figure 5(a)). We artificially increased fire intensity in order to show off the fire behavior and this hampered their ability to evaluate the situation and allocations. They all agreed that every firefighter should be pulled out because that building is lost and might fall at any minute! In our efforts to put a challenging fire in front of them to fight, we had caused them to walk away from the training. Once we start to add training abilities, such as to watch the fire spread in 3D, we have to also start to be more aware of how to accurately show a fire that the Incident Commander would face. We have consequently altered the smoke and fire behavior (see Figure 5(b)). The smoke appears less "dramatic" to the a lay person than a towering inferno, but it provides a more effective training environment.

4.5 Gradual Training

Initially, we were primarily concerned with changes to the system that allowed for a more accurate simulation of what the Incident Commander would actually see. Alternatively, we have also added features, not because of their accuracy, but also to aid in training by isolating certain tasks. Very often in reality and in our simulations, dense urban areas obscure the ability to see where all of the resources (i.e. fire engines) are and prevent a quick view of the situation (see Figure 6(a)). To this aim, we have added a new mode using the 3D, but having the buildings each have no height, which we refer to as Flat World (see Figure 6(b)). By using this flat view, the trainee is allowed to concentrate on the allocation of resources, without the extra task of developing an accurate world view with obscuring high rise buildings.

4.6 User Intent

A very important lesson that we learned from the LAFD, was that the Incident Commander cannot be given all information for the team and thus the human does not know all about the status of the team members and vice versa. Consequently, this lack of complete awareness of the agent team's intentions can lead to some harmful allocations by the human (Incident Commander). In order for information to be selectively available to the Incident Commander, we have allowed the Incident Commander to query for the status of a particular agent. Figure 3 shows an arrow above the Fire Engine at the center of the screen that has been selected. On the left, the statistics are displayed. The incident commander is able to select a particular fire engine and find out the equipment status, personnel status, and the current tasks that are being performed by the fire fighters aboard that engine. This detailed information can be accessed if desired by the Incident Commander, but is not thrown to the screen by all agents, in order to not overwhelm the Incident Commander.

4.7 Scale

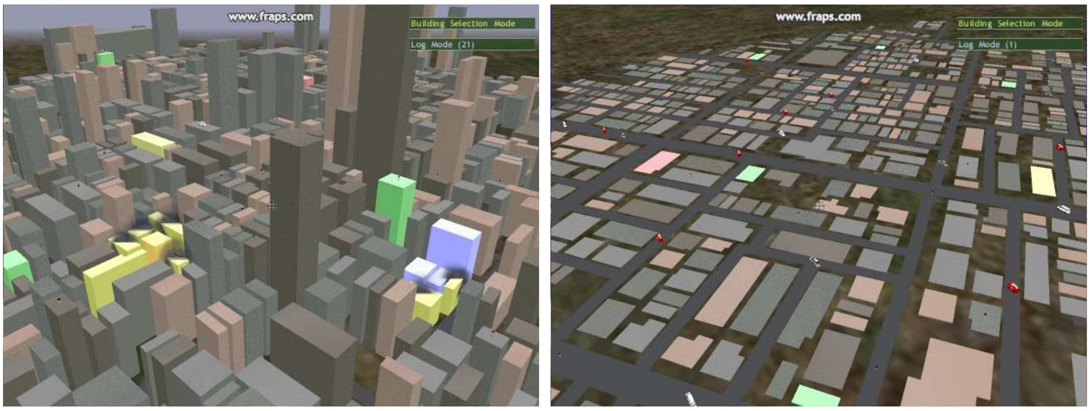
In addition, we have also learned of new challenges that we are currently attempting to tackle by enhancing the system. One of the biggest challenges in order to start simulating a large urban fire is the sheer scale of the resources that must be managed. According to the fire captains, in order to respond to a single high rise building with a few floors on fire, roughly 200 resources (fire engines, paramedics etc.) would need to be managed at the scene. Coordinating such a large number of agents on a team is a challenge. Also, as the incident scales to hundreds of resources, the Incident Commander ends up giving more autonomy to the team or else face

(a) Local Perspective (b) Global Perspective

Figure 4: Local vs. Global Perspectives in the Omni-Viewer

(a) Old Fire (b) New Smoke

Figure 5: Improvement in fire visualization



(a) Normal (b) Flat World

Figure 6: Improvement in locating resources (fire engines and ambulances)

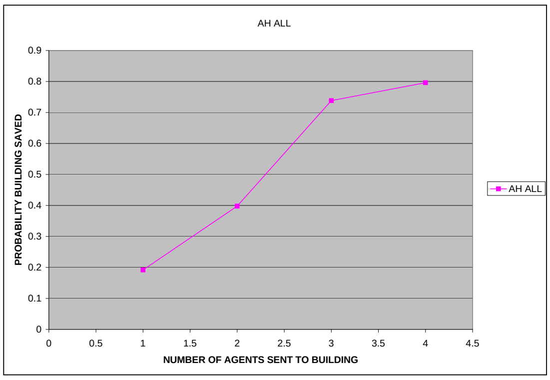


Figure 7: AH for all subjects.

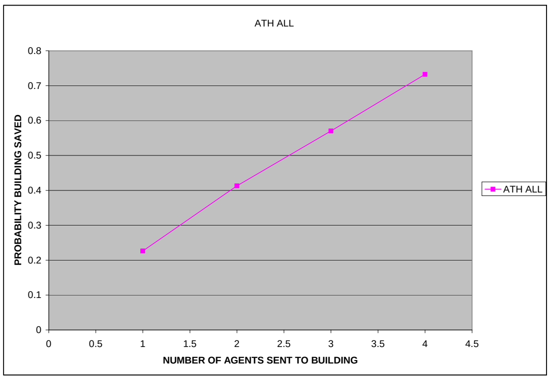


Figure 8: ATH for all subjects.

being overwhelmed. We believe that adjustable autonomy will start to play a bigger and more essential roll in allowing for the Incident Commander to monitor the larger situations.

5. LESSONS LEARNED FROM TRAINING EXERCISES

In this section, we will present results and analysis from a set of training exercises. Our initial experimental results have been published earlier [8], however the analysis presented here is new.

5.1 Training Exercises

In order to study the potential of DEFACTO, we performed some training exercises with volunteers. These initial experiments showed us that humans can both help and hurt the team performance. The key point is that DEFACTO allows such experiments with training exercises and more importantly allows for analysis and feedback regarding the exercises. Thus trainees can gain useful insight as to why their decisions led to problematic/beneficial situations.

The results of our training exercise experiments are shown in Figure 9, 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 (individual agent, then human) and $A_T H$ (agent team, then human); 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 9 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 9 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.

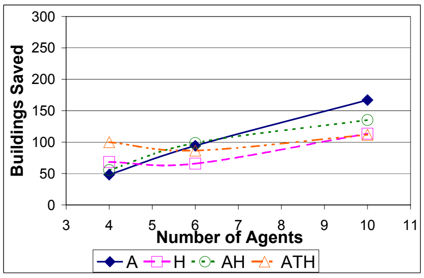
- *Complex team-level strategies are helpful in practice*: $A_T H$ leads to improvement over H with 4 agents for all subjects, although surprising domination of AH over $A_T H$ in some cases indicates that AH may also need a useful strategy to have available in a team setting.

Note that the phenomena described range over multiple users, multiple runs, and multiple strategies. 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 9). It would be useful to understand which factors contributed to this phenomena from a trainee's perspective.

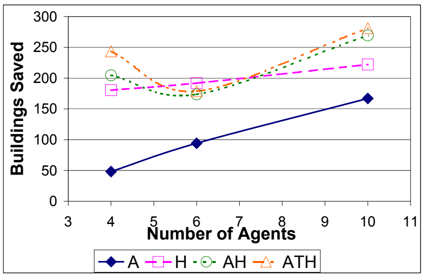
5.2 Analysis

We decided to perform a more in depth analysis of what exactly was causing the degrading performance when 6 agents were at the disposal of the Incident Commander. Figure 10 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. Another interesting thing that we found was that this lower average was not due to the fact that the Incident Commander was overwhelmed and making less decisions (allocations). Figures 11(a), 11(b), and 11(c) all show how the number of buildings attacked do not go down in the case of 6 agents, where poor performance is seen.

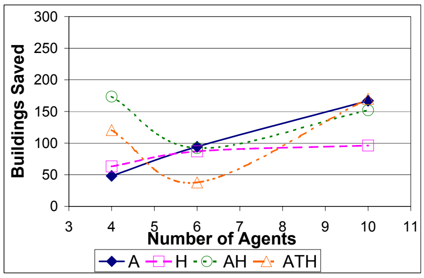
Figures 7 and 8 show the number of agents assigned to a building on the x-axis and the probability that the given building would be saved on the y-axis. The correlation between these values demonstrate the correlation between number of agents assigned to the quality of the decision.



(a) Subject 1

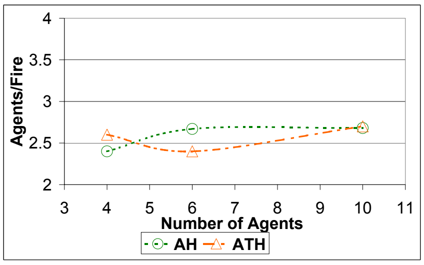


(b) Subject 2

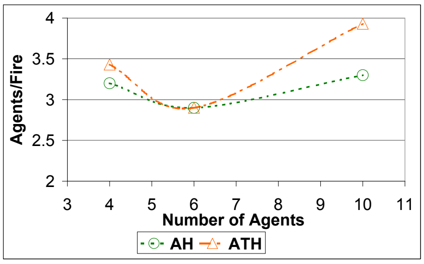


(c) Subject 3

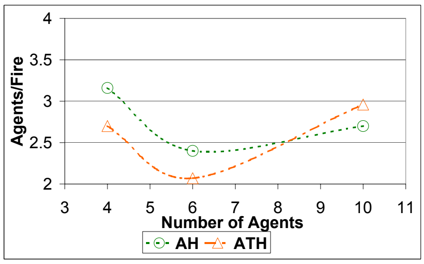
Figure 9: Performance.



(a) Subject 1

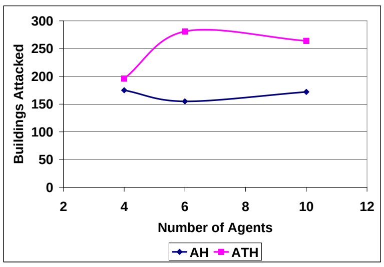


(b) Subject 2

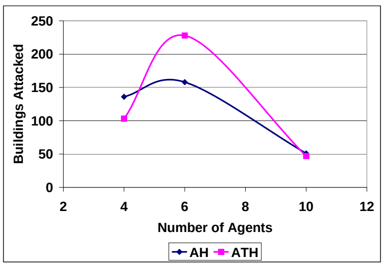


(c) Subject 3

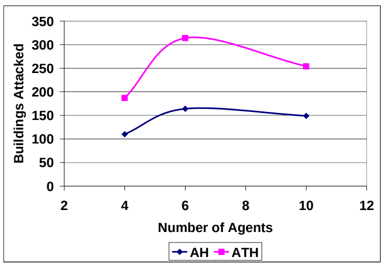
Figure 10: Amount of agents assigned per fire.



(a) Subject 1



(b) Subject 2



(c) Subject 3

Figure 11: Number of buildings attacked.

We can conclude from this analysis that the degradation in performance occurred at 6 agents because fire engine teams were split up, leading to fewer fire-engines being allocated per building on average. Indeed, leaving fewer than 3 fire engines per fire leads to a significant reduction in fire extinguishing capability. We can provide such feedback of overall performance, showing the performance reduction at six fire engines, and our analysis to a trainee. The key point here is that DEFACTO is capable of allowing for such exercises, and their analyses, and providing feedback to potential trainees, so they improve their decision making, Thus, in this current set of exercises, trainees can understand that with six fire engines, they had managed to split up existing resources inappropriately.

6. RELATED WORK AND SUMMARY

In terms of related work, it is important to mention products like JCATS [9] and EPICS [4]. 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. More recently multiagents have been successfully applied to training navy tactics [10] and teams of Uninhabited Air Vehicles [1, 2]. Our work is similar to these in spirit, however our focus and lessons learned are based on the train of Incident Commanders in disaster rescue environments.

In summary, in order to train Incident Commanders for large scale disasters, we have been working on the DEFACTO training system. This multiagent system tool has begun to be used by fire captains from the Los Angeles Fire Department. We have learned some valuable lessons from their feedback and the analysis of some initial training exercise experiments. These lessons were gleaned from LAFD feedback and initial training exercises. The lessons learned from the feedback from the LAFD include: system design, visualization, improving trainee situational awareness, adjusting training level of difficulty and situation scale. We have taken these lessons and used them to improve the DEFACTO system's training abilities. We have conducted initial training exercises to illustrate the utility of the system in terms of providing useful feedback to the trainee. Through DEFACTO, we hope to improve training tools for and consequently improve the preparedness of Incident Commanders.

7. ACKNOWLEDGMENTS

Thanks to CREATE center for their support. Also, thanks to Fire Captains of the LAFD: Ronald Roemer, David Perez, and Roland Sprewell for their time and invaluable input to this project.

8. REFERENCES

[1] J. W. Baxter and G. S. Horn. Controlling teams of uninhabited air vehicles. In *Proceedings of the fourth international joint conference on Autonomous agents and multiagent systems (AAMAS)*, 2005.

[2] S. Karim and C. Heinze. Experiences with the design and implementation of an agent-based autonomous uav controller. In *Proceedings of the fourth international joint conference on Autonomous agents and multiagent systems (AAMAS)*, 2005.

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

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

[5] 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.

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

[7] 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.

[8] N. Schurr, J. Marecki, P. Scerri, J. P. Lewis, and M. Tambe. The defacto system: Training tool for incident commanders. In *The Seventeenth Innovative Applications of Artificial Intelligence Conference (IAAI)*, 2005.

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

[10] W. A. van Doesburg, A. Heuvelink, and E. L. van den Broek. Tacop: A cognitive agent for a naval training simulation environment. In *Proceedings of the fourth international joint conference on Autonomous agents and multiagent systems (AAMAS)*, 2005.