# AVSAR: a collaboration system for disaster search and rescue operations using autonomous vehicles

**Feniosky Peña-Mora[1] and Mathias Craig[2]**

ABSTRACT | The disaster relief community is increasingly focused on issues of critical physical infrastructure in search and rescue operations. As the disaster relief and civil engineering community attempts to expand its abilities in this arena, it is being confronted with constraints related to manpower, risks to human personnel, and system stability. The community can address these barriers by integrating autonomous vehicles and intelligent software agents into its traditionally human elements. The military has been actively pursuing this goal in order to minimize human casualties and expand its functionality, and a technology transfer to the disaster arena would be greatly beneficial. The transition from the military to the disaster relief community is a logical step because of the great number of similarities between the two areas. Both are concerned with operations carried out in hostile, chaotic environments, where many participants from different areas of expertise collaborate to reach an objective, and both are constrained by the quality of intercommunication and the effectiveness of their equipment. Experience gained by the military in the field of autonomous vehicles has shown that while the ratio of autonomous vehicles to humans remains low, there is little trouble in directly controlling these vehicles as personnel can be dedicated to this task alone. However, as the number of autonomous vehicles increases to include personal human assistants and entire teams of vehicles, the task of control and collaboration becomes increasingly difficult. To date, most autonomous vehicle control work has been done with a one-to-one structure where one human controls one vehicle. While this works well when the vehicles are relatively simple and the number of vehicles is small, it does not translate well into the ideal situation of large populations of complex autonomous vehicles. Under these circumstances, intelligent software agents, residing both on the autonomous vehicles and on the communication devices, are needed to handle the task of distributed decision-making. This autonomous decision making ability is particularly critical for the cases where the autonomous vehicles fall out of contact with their human commanders or remote experts such as geotechnical,

1. Department of Civil and Environmental Engineering, University of Illinois at Urbana-Champaign, IL, USA
2. Department of Civil and Environmental Engineering, Massachusetts Institute of Technology, Cambridge, MA, USA

International Journal of IT in Architecture, Engineering and Construction
Volume 2 / Issue 2 / May 2004. © Millpress

structural, and earthquake engineers. This paper examines past work done for and by the military in the area of autonomous vehicle systems and examines its application to the field of disaster relief involving critical physical infrastructures. It then presents a system that meets the needs of a combined human – intelligent software agent – autonomous vehicle SR (Search and Rescue) team, operating on critical physical infrastructure in an unstable and hostile environment. The collaboration infrastructure includes an information policy layer and a client application layer that address the need for inter-user communication and flexible command structures, which can be dynamically arranged to meet the situational need.

KEYWORDS | collaborative environments, disaster relief, search and rescue, autonomous vehicles, intelligent software agents, self-organization, control structures, information policy

## 1 Scenario

Emerald City is a large city with a densely populated urban center. On what begins as a typical productive workday, the city is attacked without warning. There are severe disruptions to many components of the civil infrastructure including major freeways, bridges, rail lines, and the airport. In addition, most of the communication infrastructure is damaged or destroyed and it is suspected that chemical and biological weapon agents have been used along with a variety of explosive devices. With no communication infrastructure in place, potentially hazardous materials released into the environment, and a severely damaged transportation infrastructure, the search and rescue (SR) operation gets underway. The SR team is composed of specialists from many fields ranging from civil engineers to medical personnel to hazardous material experts. There is no time to train people in the required fields of expertise, so team members are flown in from many different parts of the country. Some are equipped with powerful communication devices while others are limited to cell phones and personal digital assistants. In order to assess the hazardous material threat and gain access to blocked areas, while simultaneously minimizing the health threat to humans, a number of autonomous vehicles, both aerial and land based, are integrated into the SR group. The vehicles are equipped with various sensor payloads to detect and assess the field threats that are identified by the human members of the SR team. They are capable of being controlled remotely but also host a number of intelligent software agents that possess decision-making capabilities. The agents, which also reside on the collaboration devices, perform support functions such as navigation course planning, dynamic map generation, and resource allocation. They possess high-level communication abilities, allowing them to interact with each other as well as with humans.

Before any SR operations can be carried out, the communication infrastructure needs to be restored. To accomplish this, the SR team deploys all terrain vehicle-mounted mobile servers with wireless networking antennas into the disaster area and establishes an area of coverage that allows the SR participants to communicate. However, given the instability of the network due to the mobility of the servers, the hostility of the environment, and the mobility of the SR clients, a collaboration infrastructure layer is needed to reside on top of the hardware to ensure maximum connectivity and a failure tolerant data storage mechanism.

In order to facilitate the SR operations and the interactions between team members, a planning layer is placed on top of the collaboration infrastructure layer. The planning layer consists of client applications that allow the users to manipulate data and communicate amongst each other. A graphical, standardized, user-friendly environment is provided for the human users. With this system in place, the SR team is ready to perform its primary function – save human lives. Using the collaboration system to coordinate their activities, the SR team deploys throughout Emerald City to assess the damage and find survivors. They initially focus on the downtown center where the largest number of people were caught in the daytime attack.

One rescue team finds that the attack has extensively damaged an old landmark building, which is at the core of the urban civil infrastructure. The team notices some visible damage to the building's foundation and needs to assess the stability of the building before proceeding to evacuate its trapped and injured occupants. Using the SR collaboration network, the search and rescue team is able to contact an off-site structural engineering firm with access to computer models of the building. The SR team collects as much data on the status of the building as it can, using its portable sensors and by coordinating information gathering operations with the sensor-laden autonomous vehicles, and passes it on to the structural engineering firm. The firm then enters the data into the computer model in order to create a model of the building in its current state based on the best information available from the field. The structural engineers, who have knowledge of the appropriate building codes and specifications, then use computer aided design and structural analysis packages to consult with the on-site search and rescue team as to the best approach to operate in and around the building. Armed with this critical information, the SR team is able to create an evacuation plan that avoids the highest risk areas and all the occupants are safely escorted from the building.

## 2 Introduction

Disaster relief is a field of global interest. Current international events have lead to an intense examination of, and desire to improve, search and rescue procedures in response to both natural and human-made disasters. One area of considerable interest is the integration of civil engineers as part of the first response teams. Moreover, there is also a great interest in integrating autonomous vehicles and intelligent software agents into traditionally human search and rescue endeavors. The advantages of integrating autonomous vehicles and intelligent software agents are numerous and include vastly expanding the types and quantity of terrain that can be searched and reducing the risk to human rescuers. They also open the possibility of examining and repairing difficult to reach critical civil infrastructure damaged in disasters.

The research presented in this paper leverages years of work done by the Draper Laboratory and MIT in the area of collaborative infrastructure in hostile and unstable environments under the MICE (Mobile Integrated Collaboration Environment) umbrella ((Yusuke, 2001), (Kuang, 2001), (Sen, 2001), and (Aldunate, 2002)). Some of the most salient aspects of the MICE initiative have been the development of an infrastructure that handles frequent and unpredictable losses of contact, interoperates with multiple computational devices such as cell phones, PDAs, and laptops, and operates on multiple platforms for communication among human rescue workers, remote experts such as structural engineers, and autonomous vehicles involved in search and rescue efforts. Building on this, the current work presents an architecture where autonomous vehicles are used to augment a human search and rescue operation composed of rescue workers on the ground and civil engineering remote experts. This new architecture, known as the Autonomous Vehicle Search and Rescue (AVSAR) planning system, allows for direct mission planning, whereby human controllers, in collaboration with remote experts, direct the movements of autonomous vehicles, as well as autonomous mission planning done by the autonomous vehicles themselves. The system also provides a flexible control structure allowing one-to-one, one-to-many, many-to-one, and many-to-many relationships.

This research into the integration of distributed team collaboration and multi-agent, multi-vehicle autonomous systems is expected to have significant impact both in research and in industrial domains. The impact on research is expected to manifest itself in the creation of more intricate protocols for human-to-autonomous vehicle interactions as well as the development of mechanisms for ever more autonomous operation. The impact on industry is expected to follow from the demonstration of the benefits of a tightly integrated human and autonomous vehicle collaboration space where human risk factors are minimized and capabilities are greatly expanded.

## 3 The MICE infrastructure

The MICE infrastructure is a client-server architecture designed to support collaborations in unstable, chaotic, and unpredictable environments. Because search and rescue operations often take place under hostile conditions, considerable effort was placed in designing the infrastructure to allow for ubiquitous operation given large uncertainties in network connectivity and client availability. In order to achieve this, special attention was paid to three design objectives: Distributed databases, database synchronization, and client hand-off. In addition to addressing the issue of network instability, the system also tackled the issue of participant inclusion, or allowing as many people to participate as possible from the disaster zone as well as remote locations. This meant designing the system to be platform and device independent. While some restrictions are inherent in different devices due to differences in screen real estate and computing power, the system design allows users to participate in a manner limited only by their device. The MICE system, when deployed in conjunction with the mobile, vehicle-based, wireless communication sensors into the disaster area, provides a robust infrastructure for any SR effort. Operating behind the scenes and without requiring any effort from the user, the system provides a stable virtual working environment in an unstable physical environment.

### 3.1 High-Level System Diagram and Description

Figure 1 shows the general, high-level collaboration architecture of the MICE system, omitting many repetitive interconnections in order to simplify the view. The diagram shows three packages: one for the clients, one for the servers, and one for the simulation system. The client package contains a CollabClient class that stores the state information of the client in order to present it to, and register with, the CollabEngines running on the servers. The CollabEngines keep a register of all the collaboration sessions held at any given time. The registration of a CollabClient with a CollabEngine forms a dynamic link between the client and the server and causes all collaboration requests to be routed to this server until the link is severed. The package also contains all the client applications such as the geospatial, temporal, and controller applications used for creating disaster response plans. All of these applications communicate with the connected server through a host of helper classes, which manage the communication among applications. The server package has classes to coordinate communication with the clients, communication with other servers, and to handle data storage and retrieval. The CollabEngine is also the primary class that deals with the clients' requests as well as with communication with the other servers, and the DBServer is the class that handles all database transactions. If, for example, a client makes a data change and passes this to the server through the CollabEngine, then the CollabEngine relays this to all the DBServers that it is in contact with. The DBServers then interact with their respective databases to record the change. The final package is the simulation package. This package provides a back-end to the system which allows the system to be run with simulated vehicles instead of physical vehicles. The controller client applications make direct socket connections to the Gateway object in order to send requests to instantiate vehicle simulators. Once

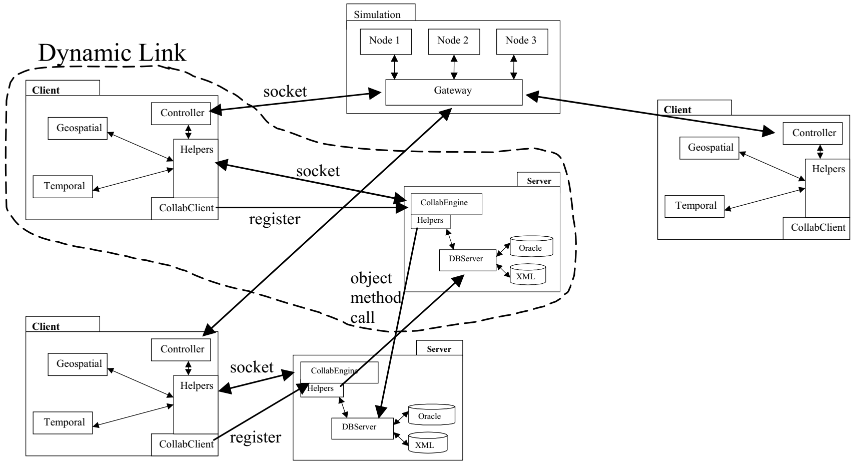


**Figure 1.** High-Level MICE system diagram. Redundant associations are omitted for clarity

the Gateway receives such a request, it passes it on to a node, where the simulator is instantiated and confirmation is sent to the controllers. The controllers then make a direct socket connection to the simulator running on the node (not shown in diagram) which allows them to receive the vehicle state information as well as send commands.

### 3.2 Platform Independence

Platform independence was achieved by using Java as the primary coding language. Java is capable of meeting this role because it operates on a layer of abstraction on top of native operating systems. Java also provides a subset language, called J2ME, which is designed for portable devices such as Java-enabled cell phones, PDAs, and pagers. This allowed for the creation of user interfaces for various devices.

Java was also a natural selection for the primary programming language because it has software packages that make it very easy to create user-friendly graphic-based applications. As the SR personnel generally operate in a stressful, chaotic environment, it is crucial that the system be easy to use and visually intuitive. Free hands may be an important requirement and having intuitive graphical input options can be much more efficient than a text based system.

### 3.3 Device Independence

It is critical to maximize the number of people in the field who can participate in the collaboration system. The strength of the system lies in its ability to facilitate the sharing of knowledge and information and therefore its effectiveness is related to the number of people participating. It is unrealistic to think that people active in the field during search and rescue operations would all have access to high-powered laptops, and it is therefore critical that the collaboration system provide interfaces to many portable devices, some of which may not have been deployed to date (Liu, 2002). While many of these devices are used primarily for communication, recent studies have shown their potential use in critical infrastructure monitoring ((Garrett, 1999), (Law, 2001)). The collaboration system presented in this paper provides interfaces to both a simulated Java-enabled phone and pager (Fig. 2), as well as an iPaq, laptop, and desktop computer.

Due to the limitations of the portable devices, such as computing power and limited programming syntax,

**Figure 2.** Java enabled cell phone and pager

both the cell phone and the pager function primarily as viewing platforms with limited input capabilities. As the J2ME coding language becomes more extensive with time, the functionality of these devices will expand to include some planning capabilities.

### 3.4 Client Hand-off

The collaboration system is composed of clients and servers, both of which are mobile. The clients continually search for service as they move about and are either picked up by an eligible server or are forced to work offline. When clients leave the range of service of their current server they again attempt to locate an eligible server. In this fashion, clients are continually being picked up and dropped from service. Whenever a client moves between two servers with a slightly overlapping service area, a smooth hand-off occurs between the two servers, much as a cell-phone user is handed from one tower to the next as he/she drives down a freeway. This service provided by the MICE system is critical as the SR user will be too preoccupied with rescue activities to manually search for service. The importance of this feature is evident in the Emerald City scenario, when the city is struck and the mobile communication infrastructure is deployed on-scene, creating a total coverage area made up of a patchwork of the individual coverage areas of the deployed servers. As the SR teams move around the affected area to assess damage to the critical infrastructure and locate human survivors, they will enter and exit these individual areas of coverage. This component of the MICE system ensures that the SR personnel are provided with the smoothest connectivity during the operation given the distributed nature of the servers.

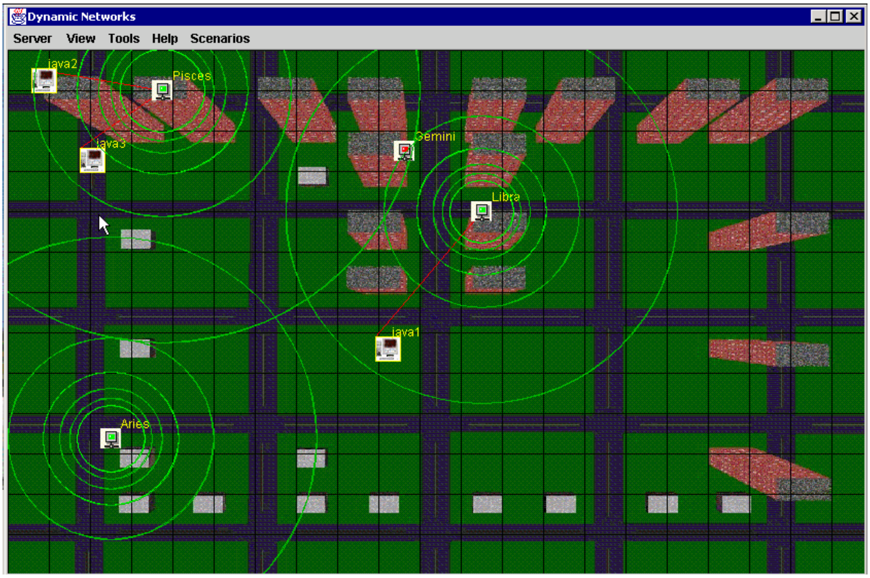


**Figure 3.** Client-Server infrastructure view. Data links are shown as straight lines

The infrastructure, consisting of the clients, servers, communication links, and server ranges, is shown in a network application in Figure 3.

### 3.5 Distributed Databases

Network connectivity is highly unreliable in the hostile environment of disaster relief operations. In order to deal with this, participants in the collaboration system keep a local repository of relevant data using flat files or a light database management system, depending on their device's capabilities. Keeping a most recent copy of relevant data allows the users to continue to function while they are out of communication with the group at large. The method used for storing data is highly dependent on the device type being used. In the current system, the clients operating on laptops and desktops use Oracle8i for data storage while the thin-client cell phone and pager keep local copies of the data in flat files. Future work in this area will focus on designing an XML based text storage system for the thin clients that will allow them to store the most critical data in a standardized manner.

The distributed database design also serves the purpose of creating backups for critical data. In a system where participants carrying critical data disappear from contact temporarily and sometimes permanently, it is important to have multiple copies of that data (Aldunate, 2002).

### 3.6 Database Synchronization

One of the major challenges introduced by having distributed databases is the issue of synchronization (Zondervan, 1998). Once data is stored in more than one locale it becomes important to consider how data is versioned, replicated, and propagated (Katz, 1990). This is a critical issue in many distributed workspace conflict resolution problems and a considerable amount of research has been done on distributed design processes in construction management ((Howard, 1999), (Krishnamurthy, 1994)). In addition, flexible relational models have been developed that extend the classical relational data model and provide semantics for database operations in the presence of inconsistent data (Agarwal, 1995).

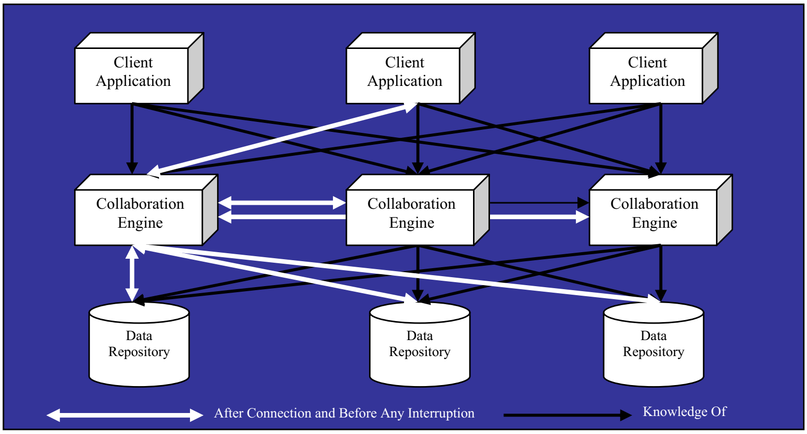


**Figure 4.** Update synchronization network

The MICE system possesses a mechanism to notify all connected databases when a data change is made locally (Fig. 4). When a change is made in a client application, a notification is sent to the collaboration engine to which it is linked. The collaboration engine then notifies all the other collaboration engines of the change. Lastly, the collaboration engine updates all of the connected databases.

Currently, the MICE system employs a mechanism whereby a user, out of contact with the group at large, works and stores data locally until contact is made. Once contact is made, the user that was isolated is updated with the group's changes. The current version of the system includes a limited versioning resolution mechanism in the case of data revision collisions; however, mechanisms that are more sophisticated will be investigated for future work in this area. These mechanisms will include authority based data ownership, where data changes must be approved by an authorized party, time-stamping (Noel, 1996), where prioritization is related to the temporal order of the data changes, and version state mechanisms (Krishnamurthy, 1994), where data versions are classified in an operational state. Future work on strengthening the MICE infrastructure will incorporate these concepts.

## 4 Background review

As different components of the system came under consideration, various topics were investigated in order to gain a better understanding of the problem and solution domains. Briefly reviewed are three of the most relevant topics: search and rescue operations, intelligent software agents, and autonomous decision-making.

### 4.1 Search and Rescue

This section is taken in large part from Yusukemi (Yusukemi, 2001) and also draws from documents describing search and rescue procedures from private, state, and federal, institutions ((NSARC, 2000), (MEOPS, 2000), and (LaValla, 1987)).

When disaster strikes, an entire chain of events is set in motion in order to evaluate the damage caused by the disaster, contain any propagation, and rescue human victims. The following provides an overview of a typical search and rescue (SR) operation in the United States and the enhancements that the MICE/AVSAR system can provide:

- Response always begins at the local level. Local fire departments, emergency management, and local and state law enforcement are the first to arrive at the disaster area and begin SR and response.

- Following the disaster, the local emergency manager may request assistance from the state; if the need is great, the state may in turn requests federal assistance; in the event of a large-scale disaster, FEMA deploys three of the closest task forces. While FEMA is deploying search and rescue personnel to the disaster site, they can also use the MICE/AVSAR system to remotely coordinate a network of structural engineers, ground personnel, and sensor-laden autonomous vehicles to begin assessing the damage to the critical physical infrastructure.
- Structural specialists, who are licensed professional engineers, then arrive on site with the task of making the rescue area safe for the rescuers, essentially becoming the fourth leg of the first responders (Prieto, 2002). They provide direct input to the SR team members about structural integrity of the building and the risk of secondary collapses. They also collaborate with remote agencies such as FEMA and a network of remote structural engineers who have access to computer models of the structures and can run high-powered simulations.
- The search team ventures into the affected area. Once victims are located, the search group begins the daunting task of breaking and cutting through thousands of pounds of concrete, metal, and wood to reach them. Again, the MICE/AVSAR system allows collaboration between the crews on the ground working to shore up the damaged infrastructure and off-site structural engineering specialists who have access to high-powered computer models of the infrastructure. This collaboration allows the ground crews to perform their duties much more effectively by benefiting from the large pool of off-site resources and knowledge.
- Throughout the effort, hazardous materials specialists evaluate the disaster site, and decontaminate rescue and medical members who may be exposed to hazardous chemicals or other contaminants. The MICE/AVSAR system allows the hazardous material specialists to use the autonomous vehicles to collect data on the hazardous releases. They can then collaborate with off-site hazardous material engineers using detection and dispersion models to identify and characterize the materials' releases. This collaboration is critical if chemical or biological weapons were used, as presented in the Emerald City scenario. In that case, the crews on the ground would need to rapidly deploy autonomous vehicles to sense the chemical or biological agents and coordinate with the off-site specialists to create a computer model of the migration of those agents. This would allow them to form an effective plan for mitigation and evacuation.
- Logistics specialists handle the vast quantities of equipment that can be used during a large search and rescue operation to support the search and extrication of the victims. The coordination of these logistics is handled by organizations such as the Disaster Response Network (DRN, 2002), which brings together corporations from the construction, transportation, and engineering industries to assist in the relief operations. The MICE/AVSAR system allows for smoother, on site, collaboration between the logistical umbrella organization and the search and rescue personnel.

While this overview is by no means exhaustive, it does provide some perspective of the enormous complexity of search and rescue operations, and how the MICE/AVSAR system can greatly enhance the ability of the search and rescue personnel to accomplish their tasks. Successfully integrating teams from multiple disciplines, geographic areas, and expertise levels is an enormous challenge, given that the operations typically unfold in the midst of stressful, hostile, and chaotic conditions.

### 4.2 Intelligent Software Agents

One of the most influential organizations dealing with the design of intelligent software agents is the Foundation for Intelligent Physical Agents (FIPA, 2002). The organization produces standards for the interoperation of heterogeneous software agents and its membership is composed of interested parties from both academia and industry. FIPA has created, and is continually refining, specifications that range from architectures to support agent communication, to communications languages and content languages, to interaction protocols which expand the scope from single messages to complete transactions. These specifications have largely become de facto standards for industry.

FIPA is primarily concerned with a business paradigm, where agents are assumed to reside in a stable, hard-wired networking environment. Therefore, consideration has been given to how the specifications can be adapted to function in an environment characterized by frequent loss of network contact and continuous reorganization of ad hoc functional groupings. For example, the FIPA specifications describe a centralized registration service that keeps track of all agents' location and services. Whenever a new agent enters the agent world, they register with the service so that other agents can locate them and their services. This centralized registry works well in the case of a stable business network but is unreliable and inappropriate for an unstable network where some form of distributed registries should be used. This issue is addressed in this paper as it applies to search and rescue operations involving civil infrastructures and remote civil engineering experts.

### 4.3 Autonomous Decision Making

The core of intelligent autonomous agents is their ability to combine known data with data taken from the environment in order to make decisions of various types. These decisions can include such things as deciding when to communicate, with whom, and about what.

Some decisions, such as an agent deciding what data to store in a persistent fashion, can be made using an objective function optimization. This technique involves maximizing or minimizing an objective function that is composed of relevant costs and benefits representing the various options involved in the decision, while being subject to some constraints ((Ragsdale, 1988), (Lasdon, 1970), and (Emslie, 2000)). In the case of data storage, each type of data may hold some value to the agent and incur a cost in storage equal to its footprint in memory. The agent could decide which data types to store by optimizing the objective function subject to the constraint of total storage space.

Other types of decisions that involve a series of sequential actions taken over a period of time are better modeled as Markov Decision Processes and can be solved using various methods, among them dynamic programming (Bersekas, 2000). In these situations, the object of interest is in a state, which is defined to be part of the possible state space, and the agent has a set of controls that it can apply to the object in order to change its state. Each state at the end of the time period has a value and each control that can be applied by the agent incurs a cost. If the state transitions are deterministic, the problem becomes a shortest path problem and all the controls applied by the agent over the entire period can be determined before the period starts. If however, the state transitions are stochastic, the agent can at best determine a policy that tells it what controls to take at each time step, given that the object of interest is in a particular state. This type of optimal decision-making applies to problems such as a guidance agent on an autonomous vehicle deciding over a period of time whether to direct the autonomous vehicle to scan a particular location of interest.

## 5 Autonomous Vehicle Search and Rescue planning system

The MICE system provides the infrastructure foundation that makes collaboration possible between distributed users, including structural engineers and first responders, operating in an unstable environment. With that infrastructure in place, an application layer is created that provides users, both human and machine, with interfaces and controls for the collaboration environment. This application layer is called the Autonomous Vehicle Search And Rescue (AVSAR) planning system and is designed to operate on multiple device types and in a platform independent fashion.

The AVSAR planning system includes applications to view and modify the world environment, both in the geospatial and temporal dimensions. It also includes mechanisms for autonomous vehicle self-assignment and algorithms for self-organization.

### 5.1 Plans and Waypoints

Search and rescue operations, like other large-scale campaigns, start with a stated goal, such as to save as many human lives as possible. The stated goal is initiated by a manager and represents the outcome of a process where risks and outcomes are weighed in order to determine the desired strategy. That goal is then projected onto a series of tasks that together make up the operation. The tasks may be order dependent and may be conditional on other tasks being accomplished prior to a specific task. In the AVSAR system, tasks are conceptualized as operation plans, consisting of waypoints, which are themselves ordered navigation subtasks. In other words, waypoints are the ordered components that make up a plan, and all the plans together make up the operation.

For example, with the search and rescue operation underway in the Emerald City scenario, structural engineers and first responders from many fields begin the process of breaking down the operation into plans. These plans are then further broken down to waypoints, or locations where some ordered task is to be performed. A structural engineer may be concerned about a historic building in the center of downtown but must also determine the structural integrity of other critical infrastructure next to the building. He/she creates a plan with waypoints for autonomous vehicles to stop and inspect the infrastructure next to the historic building. Once these subtasks have been completed, the structural engineer can create another detailed plan for inspecting the historical building itself. At this point, the plans exist but do not have any autonomous vehicles assigned to carry them out. If the engineer has command over a group of vehicles and some of them have the necessary equipment required by the plan, then he/she can manually assign them to the desired plans. In addition, any vehicles that are qualified and are operating in autonomous mode can assign themselves to the plans.

### 5.2 Dynamic Self-Assignment

Dynamic self-assignment is the process by which autonomous vehicles assign themselves to plans without human control. The vehicles have an internal model of the state of the world based on their known data and sensor readings of the environment, and take action according to an objective function optimization. Once a vehicle determines that it is most beneficial to join a certain plan, it can assign itself to this plan. As the state of the world changes over time, the vehicle continuously reevaluates its commitments and can re-allocate itself as necessary. In the currently implemented version of the AVSAR system, self-assignments are determined by a mapping of sensors possessed by the autonomous vehicles to the sensors needed by the existing plans. A vehicle assigns itself to the first plan that it is eligible to participate in. Current research (Craig, 2003) is investigating more sophisticated assignment mechanisms that take into account the value produced by certain assignments and the decreasing marginal value added for each additional asset allocated to a particular plan.

In the Emerald City scenario, dynamic self-assignment might arise if a structural engineer that wishes tohave some autonomous vehicles inspect the infrastructure adjacent up to a critical building does not have control over any vehicles. In this case, the structural engineer may create a set of plans, composed of waypoints at locations where inspection should take place, and autonomous vehicles can assign themselves to the plans that best match their goals and abilities.

### 5.3 User Applications

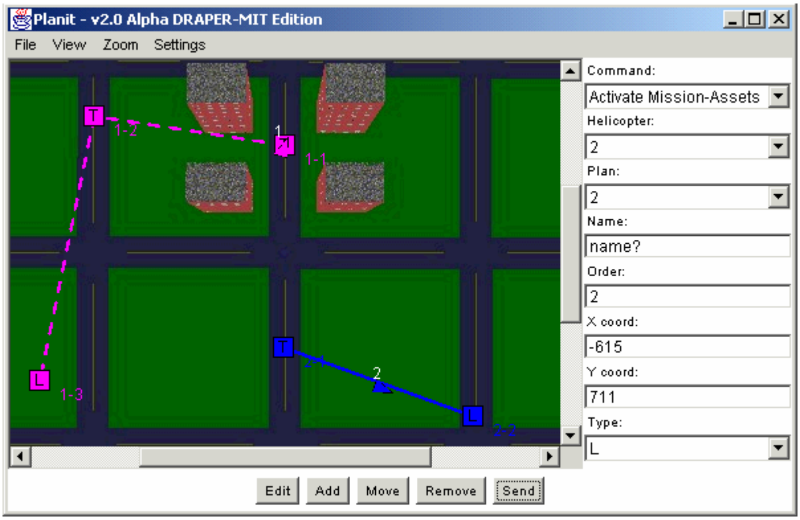
The system includes two primary user interface applications; a spatial one showing geographic information and a temporal one showing the time element of the missions. The power of the collaboration infrastructure is such that the various users interacting with these interfaces can share information and work together to achieve mission objectives. If for example, two operators are collaborating on creating a flight path for an aerial search vehicle, they can both view and manipulate the same plan on their respective interfaces simultaneously.

#### GeoSpatial / Controller

The geospatial controller is the primary user interface of the AVSAR planning system (Fig. 5). It provides a user with a bird's-eye view of the area of interest and shows vehicle assets, locations of interest, and vehicle plans. Plans, which are composed of sequential waypoints, can be created and modified either by a human user or by an autonomous vehicle, and dictate a travel path. For example, a structural engineer who is off-site can use this application within the AVSAR system to create plans to inspect damaged critical infrastructure. The engineer can specify where the autonomous vehicles should go and what action they should perform by placing specific waypoints in the plan. The geospatial controller also allows the user to add autonomous vehicles to the simulation by instantiating instances of the vehicle simulator. The vehicle simulator, developed at the Draper Laboratory (Kahan, 2001), handles the navigation of the autonomous vehicles and is very detailed.

#### *Vehicles Assets*

The functionality of the autonomous vehicles is defined by the sensors they possess, which can include infrared, biological, chemical, visual, and radar sensors, among others. A vehicle may possess none, one, or many of these sensors. The sensors that a vehicle possesses are a factor in determining what plans it is eligible to participate in. In our disaster scenario, an autonomous helicopter with all the sensors is brought in as a scout for locating areas of critical infrastructure that were



Command: Activate Mission-Assets
Helicopter: 2
Plan: 2
Name: name?
Order: 2
X coord: -615
Y coord: 711
Type: L

**Figure 5.** Geospatial-Controller user application

significantly damaged during the attack on Emerald City. The vehicle flies an automated route created by civil and structural engineers off-site, who have models to guide their plans to scan the particularly critical areas. The autonomous scout uses all of its sensors to document the damage in its path and to create a comprehensive map of the critical infrastructure. As the scout executes its plan, it detects locations where damage has been imparted to sensitive areas and creates more detailed search patterns around these areas. Any vehicles possessing the sensors needed to further study the damage are eligible to take on and execute the newly created, detailed search plans.

### *Plan Assets*

The plan assets are the sensors required in order to participate in a particular plan. While the sensor requirements could potentially be a complicated logical structure consisting of many “and”s and “or”s, the current implementation assumes that a vehicle is eligible for a plan if it possesses at least one of the listed sensors. These plan qualification requirements are important to ensure that the autonomous vehicles are most effective in serving the ground teams and the off-site specialists following attacks, such as the one in the Emerald City scenario. For example, while the scout vehicle is executing its reconnaissance mission over the attacked Emerald City, it uses its on-board camera and radar mapping sensors to detect a significant depression in a major highway artery. When it creates a more detailed search plan around this detected anomaly, it specifies that any autonomous vehicle wishing to participate in this detailed search plan must possess an on-board camera and/or a radar-mapping sensor.

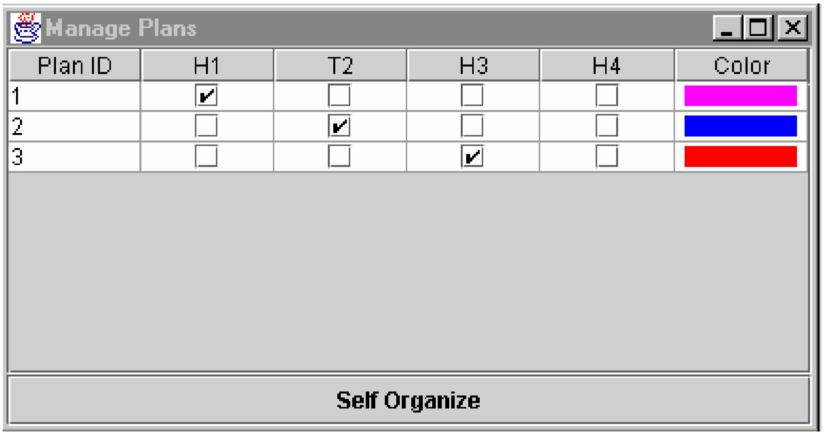
### *Resource Allocation Manager*

Once the vehicle and plan asset needs and requirements are defined, the task of resource allocation must be addressed. The issue is determining which autonomous vehicles should participate in which plans (Fig. 6). If an autonomous vehicle possesses the necessary sensor(s) required by a plan, a human user can simply assign the vehicle to participate in that plan. This might happen in the Emerald City scenario where the scout vehicle patrolling Emerald City detects a significant depression in a major highway artery and creates a detailed search plan around it.

Once the scout has created the new plan and specified the requirements in order to execute it, an off-site structural engineer may have control over a specific autonomous vehicle that he/she wants to execute the plan. The engineer can task the vehicle to take on the plan by specifying this relationship in the plan manager. It is also possible for autonomous vehicles with local planning authority to use the plan manager to specify the plans they intend to undertake based on their internal objective function optimization strategy.

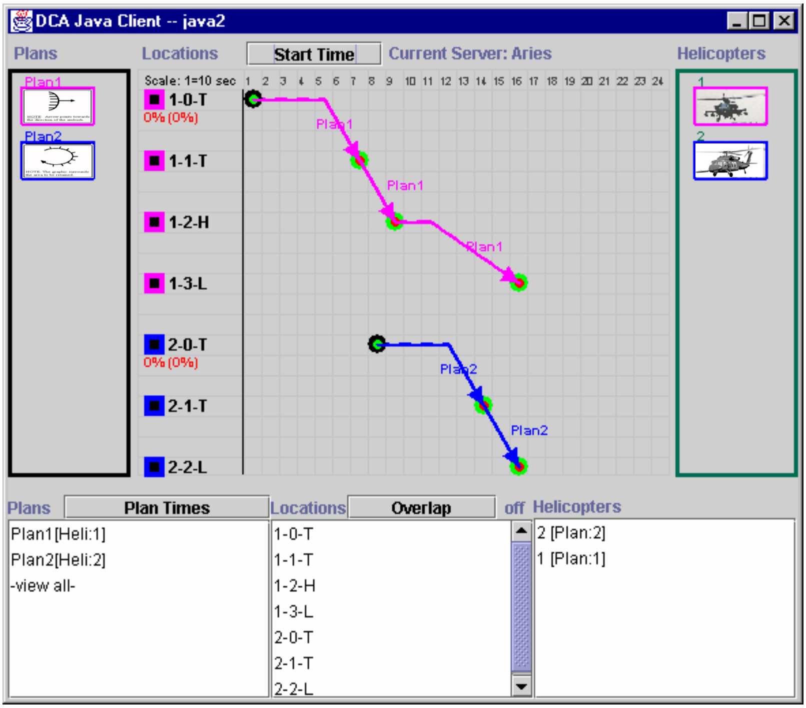
### Temporal

The temporal interface (Fig. 7) allows a user to view the mission plans in the time domain. It lists all the created plans and all the autonomous vehicles participating in the simulation and color codes associations between the two. Time is presented along the x-axis and the locations of the discrete plan waypoints are on the y-axis. To the right of the locations are strings of the form “1-1-T”. The first variable is a number that represents the plan number, the second is a number that represents the index of the waypoint in the plan, and the third is a letter that describes the action to take at the waypoint: ‘T’ is for fly through, ‘H’ is for hover, and ‘L’ is for land. The main graphical section of the interface shows two vectors for each plan. The first vector is made up of dashed lines and represents the intended arrival and departure times at all the locations. The second vector, which is constantly updated as the plan is carried out, is shown with a solid line and represents the actual progression of the plan. Ground personnel and off-site specialists can use the temporal application to coordinate departures and arrivals of the many autonomous vehicles at different locations. They can also use this application to monitor the progress of plan executions.



| Plan ID | H1 | T2 | H3 | H4 | Color |
|---|---|---|---|---|---|
| 1 | ☑ | ☐ | ☐ | ☐ | 🟪 |
| 2 | ☐ | ☑ | ☐ | ☐ | 🟦 |
| 3 | ☐ | ☐ | ☑ | ☐ | 🟥 |

**Figure 6.** Resource allocation manager



Plan1[Heli:1]
Plan2[Heli:2]
-view all-

1-0-T
1-1-T
1-2-H
1-3-L
2-0-T
2-1-T
2-2-L

2 [Plan:2]
1 [Plan:1]

**Figure 7.** Temporal user application

### 5.4 Information Policy and Control Architecture

The collaboration infrastructure allows for multiple users, both human and autonomous, to interact with other vehicles, as well as plans, simultaneously. The infrastructure does not, however, dictate who has permission to access what asset. The control of privileges and information access is the task of the information policy.

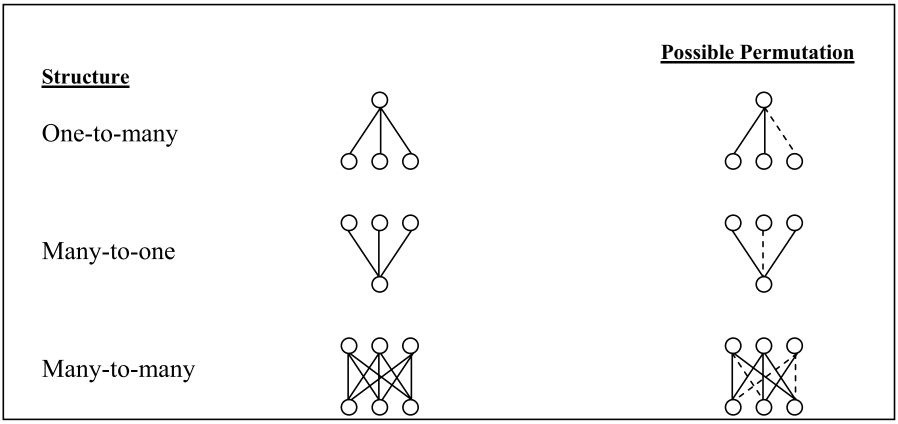


**Figure 8.** Control structures. The dashed lines represent cases where control is theoretically possible but is blocked either due to communication barriers or the application of an information policy

Search and rescue efforts generally have a hierarchical organization with different people and autonomous vehicles at different levels of control. While some users may have the ability to access information from and/or give orders to another user, they may not have that level of privilege within the organization. The information policy sets the guidelines for which users have access to which assets. Figure 8 shows the three types of control structure, one-to-many, many-to-one, and many-to-many, and illustrates how an information policy might be applied to them. The circles in the figure represent users, both human and autonomous, the solid lines represent lines of communication/control, and the dashed lines represent disallowed lines of communication/control.

### One-to-Many Control Structure

The one-to-many control structure is a generic hierarchical structure where a manager oversees several users. In the case of autonomous vehicles, this structure is typical of a human controller, such as a chief structural engineer, overseeing the activities of multiple autonomous and semi-autonomous vehicles. The chief structural engineer could guide the vehicles through the creation of a series of waypoints and can manage simple payloads on multiple vehicles. The applied information policy may establish a functional group that is a subset of the physical communication set. This can be the case when although off-site structural specialists have access to the autonomous vehicles, they do not have the authority to directly order them to participate in plans. In this case, the structural specialists may need to give their recommendations to their superiors who may have command privileges and can, in turn, assign the autonomous vehicles to the specified plans.

The one-to-many structure is also applicable in the case where a payload specialist operates the same payload on multiple vehicles. This occurs, for example, when an imaging specialist is charged with operating the camera payload on several vehicles. In the Emerald City scenario, the imaging specialist might receive one of the first SR tasks after Emerald City is attacked to survey the area of interest and document the destruction. She can use all of the vehicles under her command to build a composite image of the entire area. This image can then be used to aid the structural engineers in planning which areas need immediate attention, which areas show damage to the most critical infrastructure, and which areas are relatively unharmed.

### Many-to-One Control Structure

This control structure is useful when multiple human users out in the field are collaborating on designing a search and rescue operation. If each user has unique knowledge as to valuable search areas, they can collaborate and each contribute waypoints to create an effective search pattern. This type of collaboration can be particularly useful in the immediate wake of the attack on Emerald City, where a broad damage assessment perspective is needed. At this criticaltime, off-site structural and transportation engineers can team up with on-site search and rescue personnel to coordinate the information discovery stage. Between them, they can create a master search plan to include transportation infrastructure analysis as well as critical building stability analysis and personnel safety issues.

The many-to-one control structure is also applicable in the case where a group of payload specialists, each with a different specialty, operates on one vehicle. One specialist could be in charge of navigation, one of imaging, and one of communication systems.

#### Many-to-Many Control Structure

The many-to-many control structure is perhaps the most common for large search and rescue operations. It is a combination of the one-to-many and many-to-one structures and typically has each autonomous vehicle being controlled by multiple collaborating users and each user participating in the control of multiple vehicles. In the disaster relief operation, this would mean that many of the human users have some level of control over multiple autonomous vehicles participating in the search. If a human reaches an area that cannot be searched due to a health hazard or a physical obstacle, it can engage one of the autonomous vehicles within its command group to carry out the search. This control structure allows for the most amount of flexibility in that a coalition of off-site specialists and on-site personnel have the greatest number of options to address the situation optimally. The many-to-many structure also allows for interactions between a dynamic group of autonomous vehicles where they can communicate between each other and engage each other in activities.

#### Information Policy

An information policy schema was developed for the collaboration system. The information policy controls access to the client applications by a standard login procedure. When a user logs in, their rank in the disaster relief hierarchy is determined and their base set of privileges is established.

In the current implementation, the base privileges allow the user to manipulate the priority levels of the search and rescue missions, plans, and vehicles. Once the user is logged into the system, he/she is able to launch the various client applications as well as view the other users logged into the system and what applications they are running. In the Emerald City scenario, this information policy is critical in the aftermath of the attack for the purpose of maintaining order amongst friendly forces but also to guard against sabotage from enemy forces attempting to infiltrate and disrupt the search and rescue operations. Restricting access to resource allocation functionality to users who have properly logged into the system and have been properly identified by their security clearance helps prevent infiltration into the system.

### 5.5 Dynamic Self-Organization

Vehicle-to-plan assignments, control structures, and information policies are all part of the larger concept of an organization. An organization defines control relationships between its constituent members, as well as restricts information flow between them. Once these parameters have been established, an organization functions to allocate its resources in the most effective way possible to achieve its adopted goals.

The most recent work on the AVSAR system has been to integrate the concepts of self-assignment, control structures, and information policies into the singular concept of organizations and to shift the focus of autonomy to dynamic self-organization (Craig, 2003). Dynamic self-organization of autonomous vehicles is defined as the integration of five steps that include: goal evaluation and adoption, organizational strategizing, goal decomposition and aggregation, organizational design, and finally, organization adoption. These five steps allow autonomous agents to reason about creating organizations in order to pursue perceived goals.

In this paper, we focus on the fourth step of dynamic self-organization – organizational design. A four-phase systems engineering methodology has been developed for designing organizations by sequentially optimizing along multiple dimensions, which include boundary, size, structure, and membership. Algorithms that yield optimal solutions for each phase of the design methodology exist, but are not suitable for realistic scenarios, which are often characterized by large state spaces. Therefore, the design methodology has been implemented using heuristic algorithms that trade optimality for problem tractability.

### Four-Phased Organizational Design Methodology

The first phase of the design methodology defines an organization along the dimension of abstract membership. An organization's abstract membership is defined as the roles and resource types that the organization has in order to pursue its adopted goals. Each member of an organization contributes in some way to the execution of the adopted set of goals and in turn costs the organization something. The abstract membership problem is formulated as a minimization problem where the objective function is a weighted sum of goal processing time (T), organization cost (C), and goal failure probability (1-Z):

$$\min J, \quad where\ J = w_t \cdot T + w_c \cdot C + w_s \cdot (1\text{–}Z)$$

The second phase of the design methodology organizes the abstract members enumerated in the first phase into sub-groups. This division into sub-groups reflects the advantages an organization gains, in terms of operational efficiency, by limiting the communication links amongst its members. This gain in operational efficiency is at the core of why hierarchical organizational structures are so prevalent. The sub-groups are created using an agglomerative clustering algorithm. This is an algorithm whereby every member initiallystarts as a separate cluster. Selected clusters are then merged one by one until the desired number of clusters remains. The selection criteria for choosing which clusters to merge is based on the desire to minimize the maximum amount of internal and external information coordination (W) over all the clusters (d). Internal information coordination is defined as the number of members in a cluster and external information coordination is defined as the number of communication links that exist between a given cluster and all other clusters due to shared tasks. Once the final clusters have been determined, they are then assigned to manager roles – a specialized type of role – thereby forming the initial structure of the organization. Mathematically, the grouping problem is:

$$\min J, \quad where\ J = \max_d W_d$$

The third phase fully defines the structure of the organization by creating a hierarchical structure between the manager roles. This is accomplished by using a max-in algorithm, whereby the coordination links ($D_{ij}$) between manager roles with the heaviest use are used to generate a tree with no cycles. Formulated as a minimization, the problem takes the form:

$$\min J, \quad where\ J = \sum_{(i,j)\in E(T)} D_{\max} - D_{ij}$$

A manager role is then selected to be the organization's leader role, thereby creating a unique hierarchy among the manager roles and all the members of their corresponding clusters.

The fourth phase maps an organization's abstract members, in the form of roles and resource types, onto concrete entities, such as actual autonomous vehicles and physical resources. This is the step where the qualifications, such as experience and abilities, of actual autonomous vehicles and physical resources are considered in order to determine the mapping that leads to the most effective organization. The end result of the design process is a fully specified organization, with a defined membership, control structure, and information flow network, which is designed to pursue a specific set of goals or plans.

## 6 Simulation

The AVSAR system is designed to be deployed in real disaster relief search and rescue operations. However, search and rescue operations are high risk and any technology that is to be used in these operations must first be thoroughly tested. In order to provide a platform for system verification and evaluation, an autonomous vehicle simulation engine was integrated with the AVSAR system. The simulation engine also has the potential to serve as a training tool to teach SR team members including structural engineers and first responders, effective ways of collaborating during search and rescue operations.

The vehicle simulation engine takes input from users such as navigation points in a plan and continuously returns, as output, the updated state data for the vehicle. The autonomous vehicle simulator used is coded in C and C++ and is computationally intensive (Figure 9 shows the 3D vehicle simulator environment).

### 6.1 Architecture

The complexity of the simulation engine limits the number of simulations that can be run on one processor so a distributed simulator architecture was employed to spread out the computation load (Fig. 10).

The distributed architecture allows for vehicle instantiations, called for by the controllers, to be distributed over many nodes. The gateway processor acts as a middleware between the controllers and the nodes and balances out the workload over all the nodes. Once a node receives instructions to instantiate a vehicle simulator it sends a confirmation message back to all the controllers. The controllers then each make a direct connection to the simulator running on the node in order to send commands to, and receive state data from, the vehicle. This architecture allows multiple users to collaborate on the control of an autonomous vehicle. If for example, two off-site structural engineers were both investigating the same geographic area for damage to the critical infrastructure, they could use their individual controllers to collaborate on creating a search plan and on assigning vehicles to their joint plan.

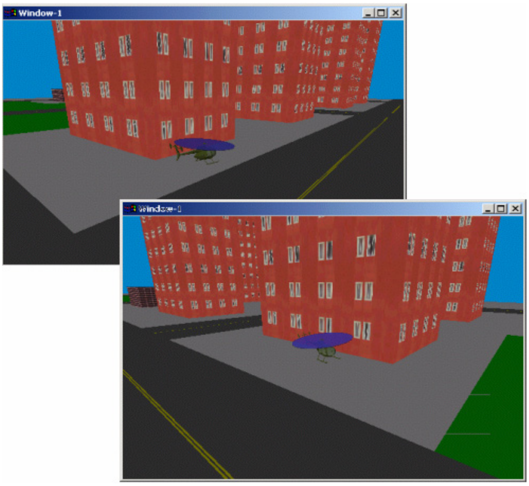


**Figure 9.** 3D representation of the simulation environment

### 6.2 Emerald City Simulation

A simulation of the Emerald City scenario was constructed in order to demonstrate the operation of the MICE/AVSAR system. In this simulation, the MICE/AVSAR system is deployed in an urban setting using three mobile servers and three mobile human SR clients, which include a structural engineer and two first responders (the client-server infrastructure view is shown in Fig. 3). As both the servers and the clients move around, their intercommunication links are dynamically established and severed. Each client has a local database in which it stores all of its data. When the clients are within communication range, they notify each other of data changes, and when they are isolated, they simply store all changes locally.

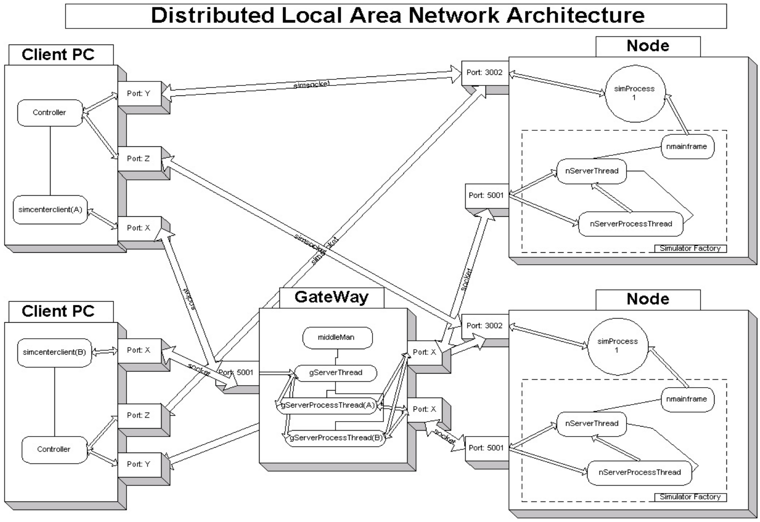
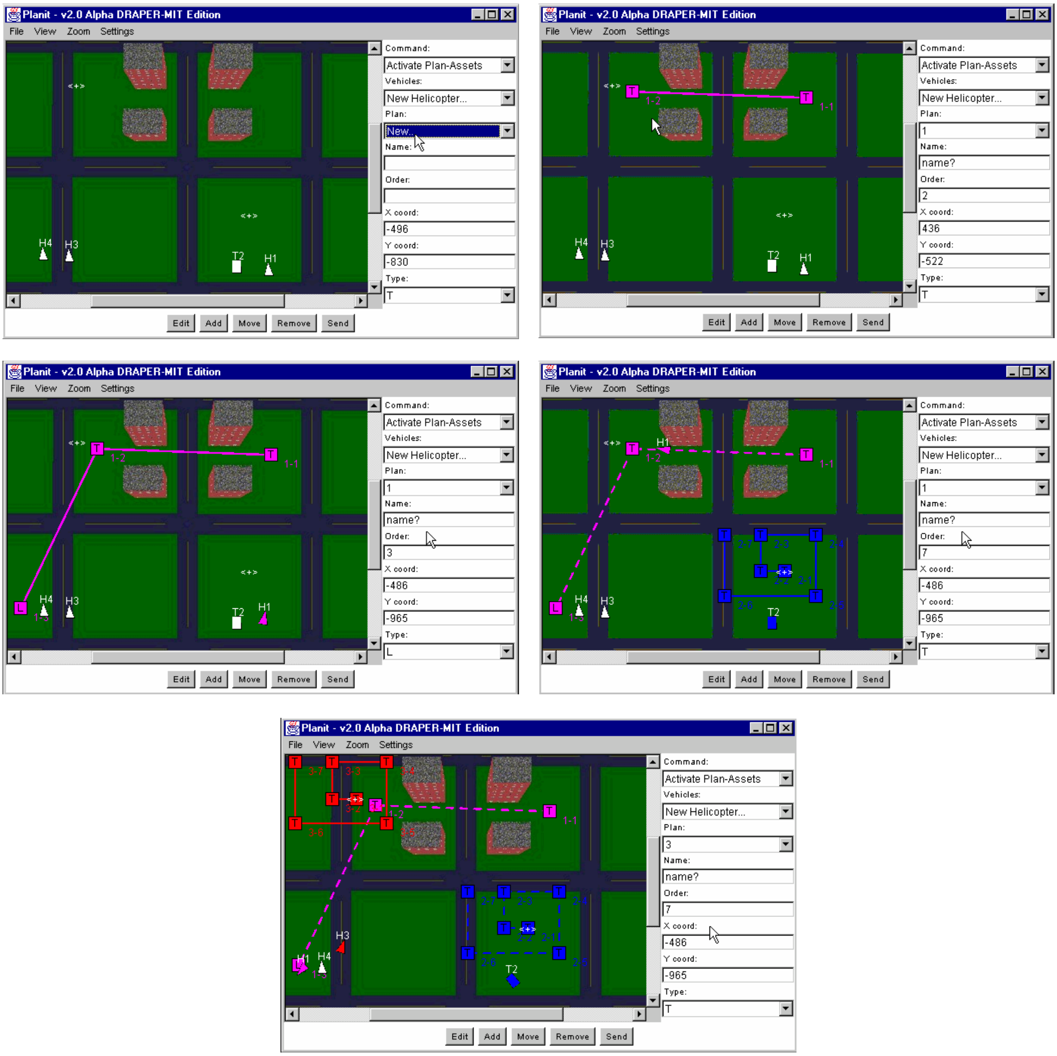


**Figure 10.** Distributed simulation architecture. Solid lines show class relationships and arrows show lines of communication

The human SR clients use the AVSAR system to instantiate multiple autonomous vehicles and place them around the disaster area (Fig. 11, top-left). Among the autonomous vehicles are ground vehicle simulations, and helicopter simulations, representing aerial vehicles. Two "locations of interest" are placed in the scene to represent objects wanting to be identified and explored by the search and rescue operation. Each one has a distinct signature that can only be detected by a particular sensor. One can be detected by a visual sensor and one can be detected by a radar sensor. This would be a typical situation where structural and transportation engineers deploy imaging and depth sensing equipment in fly-overs to detect significant damage to the critical infrastructure, such as underground craters in roads and damaged bridges. The autonomous vehicles are given an assortment of sensors and one of the helicopters is designated a scout and is given all of the sensors. Multiple human users, each working with their own controller in the AVSAR system, collaboratively create a search pattern for the scout by placing waypoints wherever they deem appropriate (Fig. 11, top-right).

Once the scout search plan has been created, the scout is assigned to the plan and is given orders to execute the plan (Fig. 11, middle-left). The scout helicopter takes off and begins to fly the designated route, scanning with all sensors as it proceeds. When the scout comes within range of the first "location of interest" it detects a visual signature and creates a new plan comprised of a spiral search pattern centered on the detected signature (Fig. 11, middle-right). It knows that the location has a visual signature so it designates the newly created search pattern as needing of a camera. Because the scout's primary role is to cover the greater area and look for new "locations of interest", it continues on its original flight path once it has created the search pattern and notified the other vehicles of its discovery. The scout performs the same function when it eventually detects the second "location of interest", but this time designates the new search pattern as needing of a radar sensor (Fig. 11, bottom).

When the other autonomous vehicles receive notification of the scout's detections, they perform a sensor mapping to the newly created search patterns



Command: Activate Plan-Assets
Vehicles: New Helicopter...
Plan: New...
Name:
Order:
X coord: -496
Y coord: -830
Type: T

Command: Activate Plan-Assets
Vehicles: New Helicopter...
Plan: 1
Name: name?
Order: 2
X coord: 436
Y coord: -522
Type: T

Command: Activate Plan-Assets
Vehicles: New Helicopter...
Plan: 1
Name: name?
Order: 3
X coord: -486
Y coord: -965
Type: L

Command: Activate Plan-Assets
Vehicles: New Helicopter...
Plan: 1
Name: name?
Order: 7
X coord: -486
Y coord: -965
Type: T

Command: Activate Plan-Assets
Vehicles: New Helicopter...
Plan: 3
Name: name?
Order: 7
X coord: -486
Y coord: -965
Type: T

**Figure 11.** Images of the Emerald City Simulation. Sequential order is top-left, top-right, middle-left, middle-right, and bottom Helicopters (triangles), transports (rectangle), locations (<*>), and waypoints shown

and assign themselves to appropriate plans (Fig. 11, middle-right and bottom). They then engage themselves and carry out the search operations.

Figure 12 is a sequence diagram that shows the process of the simulation in some detail. When the autonomous vehicles receive notification from the scout ship that a new plan has been created, they perform a sensor mapping to check their eligibility. If they are eligible, they assign themselves to the newly created plan.

### 6.3 Physical Simulation

The software simulation has been integrated with Lego MindStorm robots (Kuang, 2001) in order to demonstrate the system's ability to create plans for

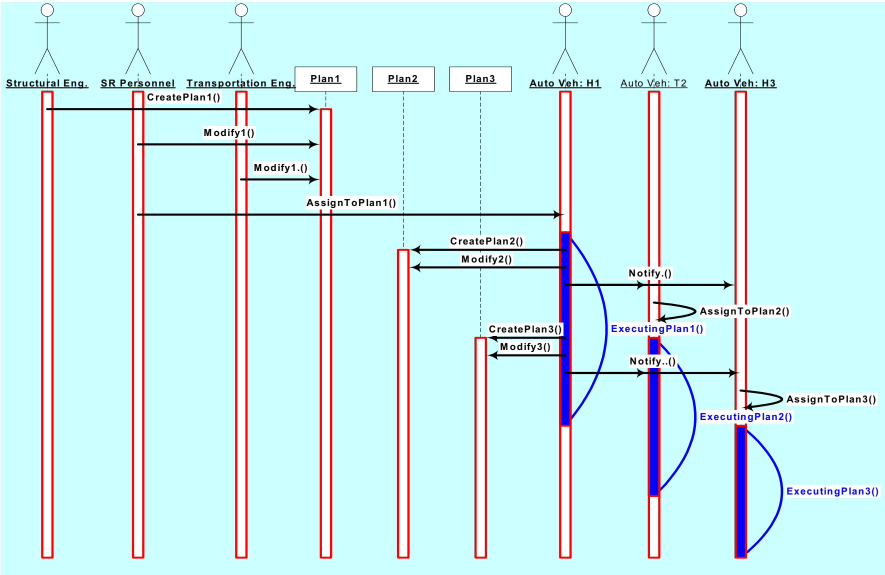


**Figure 12.** Sequence diagram of Emerald City simulation

physical participants. In this setup, search plans were created using the AVSAR planning system and the commands were sent to the MindStorm robots though infrared sensors. The robots were able to receive these commands and carry out tasks such as the navigation of waypoints and the blinking of a light. These simple robots (Fig. 13) provided a stepping stone in order to gain experience with hardware and demonstrate the system's ability to integrate with physical units.

**Figure 13.** MindStorm robot

## 7 Conclusion

This research effort set out to improve the collaboration of disaster relief search and rescue teams, involving structural engineers and first responders, operating in hostile, unstable, and chaotic environments, while simultaneously expanding their capabilities by integrating autonomous agents and autonomous vehicles into the collaboration space.

### 7.1 Research Methodology

The first step was to understand the interaction space of the distributed teams with their members coming from many fields, including structural engineering and first response, and geographic locations, both on-site and remote. This came by defining the needs of the various collaborators such as access to remote specialists, security, and data reliability, as well as defining the constraints imposed on them by their situation or duties. Once a solid understanding of this collaboration space was generated, the research focused on enhancing the traditionally human elements of disaster relief search and rescue operations with the developing field of intelligent autonomous software agents and autonomous vehicles. Frameworks were developed for search and rescue personnel to interact with the autonomous agents and vehicles as well as for the agents and vehicles to collaborate amongst themselves by forming functional groups. In order to demonstrate these developments, a simulation environment was created and presented.

### 7.2 Intellectual Merit

This research represents a powerful merging of two developing fields that are closely tied to disaster relief search and rescue operations involving structural engineering and first response: distributed team collaboration and autonomous multi-agent and multi-vehicle systems. Both fields are fertile areas of academia and the advances made through their integration present assuredly large benefits to the field of disaster relief operations as well as many others within the military and commercial sectors.

### 7.3 Impacts on Research and Industry

The merging of the fields of distributed team collaboration and autonomous multi-agent and multi-vehicle systems has created a need to develop more protocols for the complex interactions between humans and autonomous vehicles working in disaster relief search and rescue operations on critical physical infrastructure. The development of these high-level dialoging protocols that will allow for distributed plan formulation and organizational design negotiation represents a new research frontier.

The tremendous potential of autonomous systems to displace human casualties while simultaneously expanding capabilities will lead to significant military-industrial impacts. However, the greatest impact of this research will be on the disaster relief search and rescue industry. By providing a robust and flexible collaboration system, this research will help bring the actors in disaster relief search and rescue operations, such as structural engineers, autonomous vehicles, and first responders, together in ways that maximize their effectiveness.

### 7.4 Future Work

The work presented in this paper represents a significant research effort that takes the first steps towards the development of an operational system for integrating autonomous vehicles into disaster relief search and rescue operations. There are many opportunities for future research in order to increase the sophistication and functionality of the system.

#### Infrastructure Flexibility

Data storage is a significant issue to be tackled in order to make the infrastructure more flexible. The next generation collaboration infrastructure will allow the user to analyze the benefit received from, and the costs associated with, each data storage technology and choose the most appropriate one.

Network communication protocols are another area where more flexibility can be introduced. The collaboration infrastructure is currently set to use TCP/IP only. Some users may benefit from being able to select other network communication protocols such as UDP/IP (User Datagram Protocol) and RTP (Real-time Transport Protocol).

#### Dynamic Self-Organization

In order for true self-organization to be realized by autonomous vehicles in disaster search and rescue operations, involving structural engineers and first responders, a considerable amount of research must still be done on the many components of self-organization. Some of the most significant components are: goal evaluation and adoption, organizational strategizing, goal decomposition and aggregation, and organization adoption.

## Acknowledgements

The authors would like to thank John Turkovich, Milton Adams, and Jeff Cipolloni, all of Draper Laboratory, for their contributions to this work. This research was supported in part by Draper Laboratory Inc. Project#6892331, as well as the US National Science Foundation CAREER Award, and the US White House PECASE Award CMS-9875557.

## REFERENCES

[1] (Agarwal, 1995) Agarwal, S., Keller, A., Saraswat, K., Wiederhold, G., *“Flexible Relation: An Approach for Integrating Data from Multiple, Possibly Inconsistent Databases,”* Int. Conf. on Data Engineering, Taipei, Taiwan, March 1995.

[2] (Aldunate, 2002) Aldunate, R., Peña-Mora, F., Nussbaum, M., “Ad Hoc Distributed Shared Memory System for Collaboration between Critical Physical Infrastructure Experts and First Responders in Disaster Relief Situations,” IESL Internal Report, MIT, 2002.

[3] (Bertsekas, 2000) Bertsekas, D., “Dynamic Programming and Optimal Control, 2nd Edition,” Athena Scientific, Belmont, Massachusetts, 2000.

[4] (Craig, 2003) Craig, M., “A Design for Dynamic Self-Organization in Autonomous Distributed Operations,” M.S. Thesis, MIT, Cambridge, Massachusetts, 2003.

[5] (DRN, 2002) Disaster Response Network (DRN), World Economic Forum, www.weforum.org/drn (last visited 12.15.2002)

[6] (Emslie, 2000) Emslie, P.D., “Multi-Objective Evaluation of Target Sets for Logistics Networks,” M.S. Thesis, Air Force Institute of Technology, Wright-Patterson Air Force Base, Ohio, 2000.

[7] (FIPA, 2002) Foundation for Intelligent Physical Agents (FIPA), www.fipa.org (last visited 11.10.2002)

[8] (Garrett, 1999) Garrett Jr., J., D.M. Sackin, D.P. Siewiorek, A. Smailagic, J. Small, J. Sunkho., “Design of Computer Support for Field Operations,” Proceedings of the Annual Congress of the International Association of Bridge and Structural Engineers, (abstract on pp. 266–267 of Abstract Volume, full paper on CD), Rio de Janeiro, Brazil, August, 1999.

[9] (Howard, 1994) Howard, H., Keller, A., Gupta, A., Krishnamurthy, K., Law, K., Teicholz, P., Tiwari, S., Ullman, J., *“Versions, Configurations, and Constraints in CEDB,”* CIFE Working Paper #31, Center for Integrated Facilities Engineering, Stanford University, April 1994.

[10] (Kahan, 2001) Kahan, M., “Planit Documentation,” Draper Laboratory Document, August 2001.

[11] (Katz, 1990) Katz, R., “Toward a Unifying Framework for Version Modeling in Engineering Databases,” ACM Computing Surveys, Vol. 22, No. 4, pp. 375–407, December, 1990.

[12] (Krishnamurthy, 1994) Krishnamurthy, K., Law, K., “A Versioning and Configuration Scheme for Collaborative Engineering,” First Congress on Computing in Civil Engineering, Washington, D.C., ASCE, 1994.

[13] (Kuang, 2001) Kuang, C., “Multi-Server Collaboration System For Disaster Relief Mission Planning,” M.S. Thesis MIT, Department of Civil and Environmental Engineering, June 2001.

[14] (Lasdon, 1970) Lasdon, L.S., “Optimization Theory for Large Systems,” MacMillan Publishing Co., Inc., New York, 1970.

[15] (LaValla, 1987) LaValla, R., Stoffel, S., “Search in an Emergency,” The Emergency Response Institute, Inc., Olympia, Washington, 1987.

[16] (Law 2001) Law, K., Lynch, J., Kiremidjian, A., Kenny, T., and Carryer, E., “The Design of a Wireless Sensing Unit for Structural Monitoring,” Proceedings of The Third Workshop on Structural Health Monitoring, September 12–14, 2001, Stanford, CA, CRC Press, Boca Raton, FL, pp.1041–1050?

[17] (Liu, 2002) Liu, D., Cheng, J., Law, K., Wiederhold, G., "Ubiquitous Computing Environment for Project Management Services," Computing in Civil Engineering, November 3–7, 2002, Washington, DC, ASCE, Reston, VA, pp. 273–285.

[18] (MEOPS, 2000) Massachusetts Executive Office of Public Safety, "A Plan for Civilian Search and Rescue in Massachusetts," Submitted to Governor Paul Cellucci and Lieutenant Governor Jane Swift, January, 2000. http://www.state.ma.us/eops/publications/civilian_sar.htm (last visited 11.10.2002)

[19] (Noel, 1996) Noel, R., "Scale Up in Distributed Databases: A Key Design Goal for Distributed Systems," 1996. http://www.cs.rpi.edu/~noel/distr_scaleup/distributed.html (last visited 11.25.2002)

[20] (NSARC, 2000) National Search and Rescue Committee, "United States National Search and Rescue Supplement to the International Aeronautical and Maritime Search and Rescue Manual," Washington, D.C., May, 2000. http://www.uscg.mil/hq/g-o/g-opr/nsarc/nsarc.htm (last visited 11.10.2002)

[21] (Prieto, 2002) Prieto, R., "Vulnerability of Public Infrastructure: A System's Perspective," Presentation to FIATECH, Workshop on Homeland Security for Capital Projects, Landsdowne, Virginia, November 13th, 2002.

[22] (Ragsdale, 1988) Ragsdale, C.T., "Spreadsheet Modeling and Decision Analysis," South-Western College Publishing, Cincinnati, Ohio, 1988.

[23] (Sen, 2001) Sen, S., "Globally Dispersed Project Teams: Interaction Space Management," M.S. Thesis MIT, Department of Civil and Environmental Engineering, May 2001.

[24] (Yusuke, 2001) Yusuke, M., "Collaborative Environments for Disaster Relief," Msc. Thesis MIT, Department of Civil and Environmental Engennering, Boston, June 2001.

[25] (Zondervan, 1998) Zondervan, Q., Lee, A., "Data Synchronization of Portable Mobile Devices in a Distributed Database System," Lotus Technical Report 98–02, Lotus Development Corp., 1998. http://domino.watson.ibm.com/cambridge/research.nsf/2b4f81291401771785256976004a8d13/c71ebac11ec6e54f8525661600797829/$FILE/mobile.pdf (last visited 11.25.2002)