Robocup Rescue: A Proposal and Preliminary Experiences

Ranjt Nair*, Takayuki Ito**, Milind Tambe*** and Stacy Marsella†

USC Information Sciences Institute, Marina del Rey CA 90292

Abstract. RoboCup Rescue is an international project aimed at applying multi-agent research to the domain of search and rescue in large-scale disasters. This paper reports our initial experiences with using the Robocup Rescue Simulator and building agents capable of making decisions based on observation of other agent's behavior. We also plan on analyzing team behavior to obtain rules that explain this behavior.

1 Introduction

Effective search and rescue during large-scale disasters is an issue of critical social importance. It is highly relevant in Southern California (given our tragic familiarity with Earthquakes and Fires), in Japan (where the 1995 Kobe earthquake killed over 6,000 people, injured 300,000, and led to Property damage exceeding 300 billion dollars) Orissa, India (where the cyclone killed at least 10,000 people), in Taiwan (where the earthquake claimed more than 2000 lives) and indeed all over the world.

As in most natural disasters, better preparation prior to these disasters and better coordination of search and rescue efforts afterwards would have lessened its impact. The case of Kobe is well documented in [1]. In the aftermath of the Kobe earthquake, information about the disaster was not communicated immediately due to damage to the information infrastructure. Efforts to fight the rapidly spreading fires were hindered by disrupted water supplies. Rescue teams could not reach the area due to damaged roadways and the flow of refugees. Accurate information as to where they were needed was also lacking. Aerial surveillance by helicopter finally provided that information but at the cost of drowning out the faint sounds of victims trapped in collapsed buildings. Roads and open areas, which had been designed as fire breaks, actually became combustion pathways allowing collapsed wooden structures to be exposed to air.

Effective, large-scale response to such rapidly unfolding events requires the formation of plans and infrastructure prior to the event, as well as training in how to execute the plans and use the infrastructure. And since any fixed pre-existing plan will in some ways be inadequate in such dynamic situations, training is also

* nair@ISI.edu
** ito@ISI.edu
*** tambe@ISI.edu
† marsella@ISI.edu

required in how to assess the rapidly unfolding situation in the face of limited, uncertain information and then how to appropriately modify the plans. Further, the training must involve the diverse groups involved in the relief operation so that they help, as opposed to interfere with, each other.

RoboCup Rescue has been undertaken to put large-scale simulations in use in the domain of search and rescue for large-scale disasters. In the immediate term, the initiative will provide a general software-based simulator platform for simulating a wide range of disasters, including earthquakes, tsunamis, fires, etc. This will result in a common test bed that will promote research in such areas as multi-agent planning and execution for rescue operations as well as the realistic modeling of refugees and victims for the simulation system. The longer-term goals include applying the work to the engineering and training of real search and rescue operations, as well as development of deployable robotics and simulation systems for use during actual disasters. A fundamental concern is that the research has a clear social benefit. To facilitate these goals, the development team in Japan is designing the simulator so that it can incorporate real data that models the city of Kobe.

2 Our Research Goals

The following are the research issues we intend to explore in RoboCup Rescue:

1. Effective teamwork between heterogeneous organizations with differing goals
2. Rapid situation assessment
3. Decision-making under uncertainty
4. Analysis of performance of teams.

3 Current Status of RoboCup Rescue's Simulator

In the current version of the simulator (Version 0.23), an agent establishes a connection with the kernel by first sending out a request. If the kernel receives this information successfully it responds by sending the agent a confirmation of the connection along with information about the world before the disaster. The agent acknowledges the receipt of this packet by sending the kernel a packet with the header AK_ACKNOWLEDGE.

Different simulators like traffic simulator, collapse simulator, fire simulator and blockade simulator have been implemented. Each of these simulators model a different aspect of the simulation. The kernel compiles the result of their calculations and sends each agent a packet with the updated information about the agent and objects near it.

Based on this information the agent has to decide on what command it must issue. The various possible commands an agent can issue are move, open, rest, load, unload, say, tell, extinguish and stretch [2]. At the time of writing this paper the commands, say and tell have not been implemented and so there is no means of inter-agent communication.In order to solve this problem the initial sample civilian code provided to us uses a common shared memory pool used by all civilian agents. Each agent then simply updated this memory pool with all the information that the kernel sent it.

4 Initial Problem

To familiarize ourselves with the simulator we chose an initial problem consistent with the current state of the simulator. We decided to build police agents that can collaborate to effectively clear blocked roads. But in order to build multi-agents that can cooperate it is important that we have a means by which agents can communicate with each other. Since the scheme for inter-agent communication was not finalized at the time of writing this paper we decided on building agents that would observe other agents and infer their plans.

Although the sample civilian code provided with the simulator uses a common shared memory, this is clearly an unrealistic approach to communication and collaboration between distributed agents. Agents located in different areas of the world should not have access to the same information instantly as it would be like assuming that each agent is omniscient. In our implementation each agent has its own memory.

While collaboration based on plan recognition is a difficult challenge, it is extremely useful. Imagine a situation where there is a total communication break-down as a result of the earthquake or a partial communication break-down that prevents agents from different teams from communicating. In such cases, the ability to recognize the plans of other agents in crucial. Even in the event of no communication break-down it is realistic to assume that it may not be possible for rescue agents to communicate easily with civilians. Therefore these plan recognition techniques are vital even after inter-agent communication has been implemented in the simulator.

4.1 Current State of our Sample Police Agents

The primary function of police agents is to restore blocked roads via the open command. Since each agent can observe only the area in its vicinity based on what the kernel tells it, it is possible that blocked roads may be present that our police agent cannot see.

We came up with an initial algorithm for inferring the presence of blocked roads not in our sample police agent's range of vision:

If the movement of other agents within our police agent's range of view appears restricted the agent assumes that their movement is being impeded by a blocked road.

Else if another police agent in our police agent's range of view is moving our agent decides to follow it in order to help it clear up the road faster.

If neither conditions are satisfied then our police agent chooses its move randomly. As of now road blocks don't impede the flow of traffic so we don't use the first condition but this seems a reliable method of detecting blocked roads.

In order that the entire environment get systematically traversed it is important to ensure that the agent doesn't move around in circles. We do this by detecting cycles and by always choosing the road the agent has traversed the least.

Fig. 1 is a snapshot of the output of the simulation just before eight police agents begin to clear blocked roads. The police agents are shown with ids 361 to 368. Blocked roads are shown with thick black lines and roads that are clear with thick white lines. Fig. /refsnapshot2 shows a snapshot of the world after the police agents were able to clear all of the blocked roads.

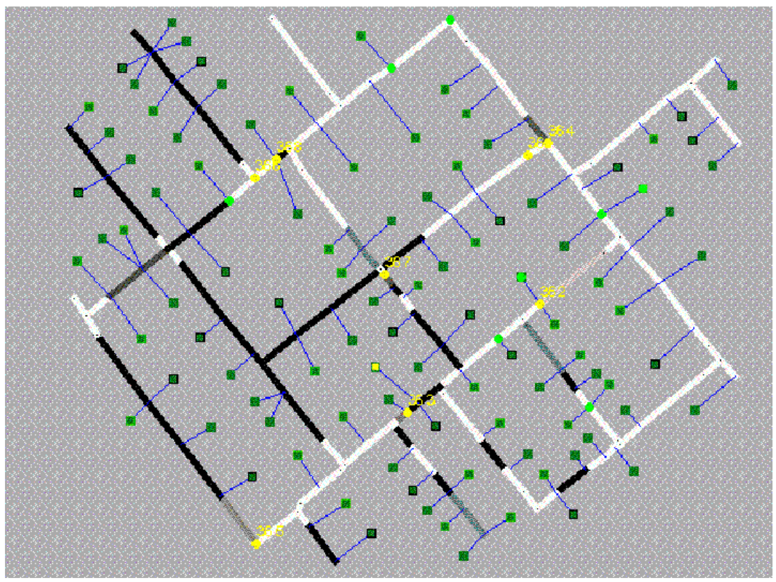

Fig. 1. Snapshot of Viewer before Road restoration.

4.2 Sample Problems

Once communication is implemented various interesting coordination and decision-making problems will arise in this simulator environment. Consider that A (ambulance) is instructed to go to help the injured civilian C.

1. A is now going to help an injured civilian C. A finds another injured civilian D. The possible choices for A are:
   (a) A helps the injured civilian D, and informs another ambulance B to go to help the civilian C.
   (b) A goes to help the injured civilian C, and informs B to go to help the civilian D.

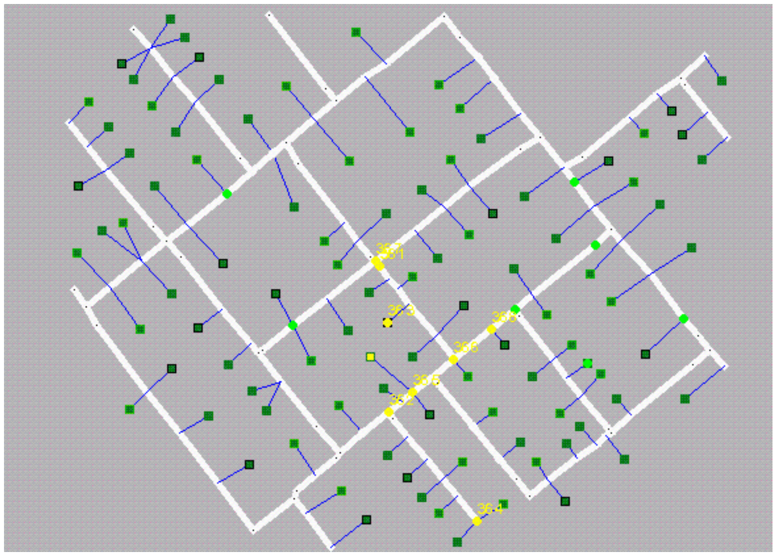

Fig. 2. Snapshot of Viewer after Road restoration.

2. A arrives at an injured civilian C.
   (a) If C is surrounded by fire, A needs a help of B-1 (fire engines). A needs to inform this situation to a fire station.
   (b) If there exist other injured civilians, A needs a help of A-2 (ambulance). A need to inform this situation to a ambulance Center.

To effectively decide upon and execute a course of action in such a situation, an agent must rapidly assess the state of the civilians and the surrounding environment. Critical decision must be made and acted upon in concert with other teams.

5 Analysis

Apart from getting the agents to work as a team we will also attempt to analyze team behavior. For this we intend to write a program for extracting rules that explain reasons for the teams success or failure at a particular task.

For RoboCup Soccer we developed a program called ISI Soccer Automated Assistant Coach(ISAAC) [3]. This uses data mining and inductive learning techniques to isolate the key issues determining the successes or failures of teams at multiple levels of behavior, including shots on goal, passing and assists, and overall performance. A snap shot of ISAACs output is provided in fig. 3.

We intend to use a similar approach to analyze the success of teams in terms of putting out fires in time, rescuing the maximum number of people, saving the maximum number of lives, etc.

We will use a two-tiered approach to the team analysis problem. The first step is acquiring models that will compactly describe team behavior, providing a basis for analyzing the behavior of the team. This involves using multiple models at different levels of granularity to capture various aspects of team performance. The second step is to make efficient use of these models in analyzing the team and presenting this analysis to the user.

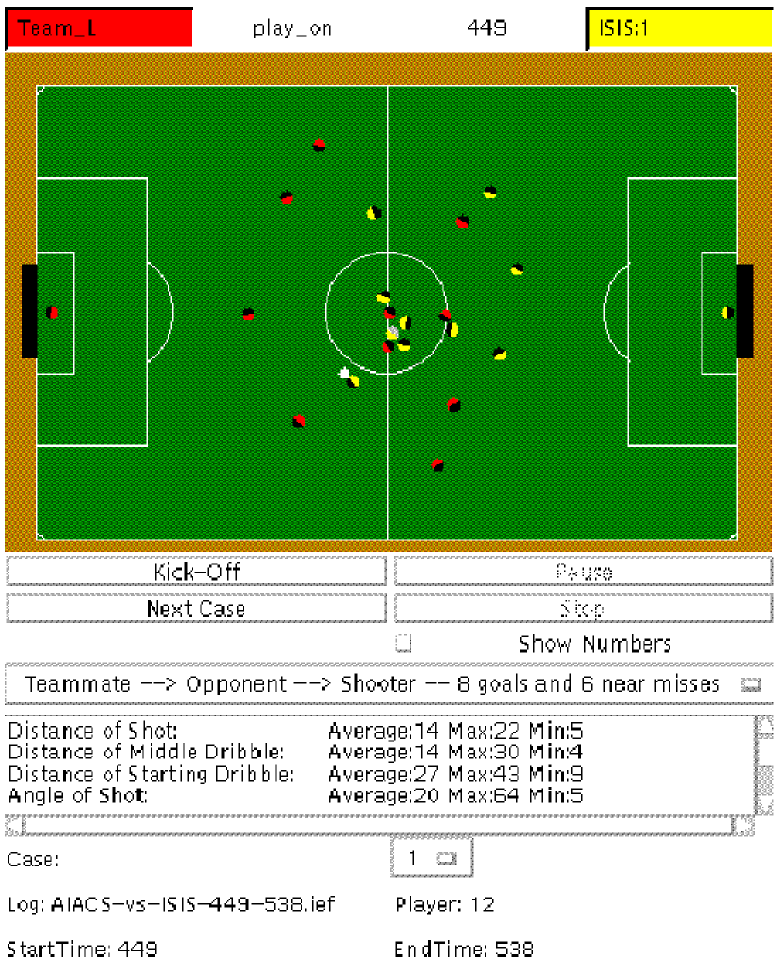

Fig. 3. Snapshot of ISAAC's output

6 Conclusion

Search and rescue in disasters is an ideal test-bed for research into multi-agent collaboration among agents with differing goals. The highly dynamic, uncertain environment and the need for speed to save more lives, will also present a challenge for the design of the agent's decision making. And owing to the complexity in the environment the analysis of team behavior will be critical to coming up with better agent designs.

To initiate our research effort we have built agents that are active in the simulation environment based on their goals and the information they receive from the kernel.

We hope that the eventual outcome of our efforts will not only benefit multi-agent research but also be useful to the people actually responsible for search and rescue in real-life disasters.

References

1. Kitano et al: RoboCup-Rescue: Search and Rescue for Large Scale Disasters as a Domain for Multi-Agent Research. Proceedings of IEEE Conference, **SMC-99**, (1999)
2. Takahashi, T., Chobu, U.: RoboCup-Rescue Simulator Manual. **version 0 release 3** (03/27/2000)
3. Raines, T., Tambe, M., Marsella, S.: Automated Assistants to Aid Humans in Understanding Team Behaviors. Fourth International Conference on Autonomous Agents (to appear), Agents 2000

This article was processed using the LaTeX macro package with LLNCS style