The Black Sheep Team Description

Jonathan Teutenberg and Cameron Skinner

The University of Auckland
New Zealand
{cam,jteu004@cs.auckland.ac.nz

1 Introduction

This short paper describes the functionality of the Robocup Rescue agents built by Team Black Sheep[1] at the University of Auckland[2]. We have implemented agents for each of the four components of the simulated environment: fire brigades, police forces, ambulance teams and emergency centres (fire stations, police offices and ambulance centres).

Our mobile agents are called Fireman Sam[1], PC Plod and Dr Ropata[2].

We begin by describing software components that are shared by all agents. Section 2 describes the sharing of knowledge, followed by a description of our path planning architecture in Section 3. The strategies used by each type of agent are then described in Section 4. Section 5 concludes with some areas that could be improved and directions for future work.

2 Knowledge Sharing

2.1 Messages

The Robocup Rescue Simulation competition allows platoon agents to send and receive up to four messages of 256 bytes every timestep. The various centre agents (fire station, police office and ambulance centre) are allowed to send and receive $2n$ messages where n is the number of platoon agents under that centre's control. To prevent the centres getting overwhelmed with messages we only allow platoon agents to send two messages per timestep.

Messages are encoded using a set of message types, a timestamp and zero or more bytes of message-dependant data. Agents queue and prioritise messages internally during a timestep. When it comes time to send the messages, they are packed into 256-byte chunks with highest priority messages in the earliest chunks. Up to two of these chunks are then sent to the kernel. If there are more messages than can be sent, the remaining messages are stored for sending in a later timestep.

[1] http://www.cs.auckland.ac.nz/~rescue
[2] http://www.auckland.ac.nz

2.2 Messaging

Agents are only allowed to read four messages per timestep. This is enforced by each agent. When a 256-byte chunk arrives the agent unpacks all the components of the chunk and looks at each component in turn. The data contained in the message is used to update the agent's model of the world, or to change the agent's behaviour if instructed by their centre. Platoon agents only listen to messages from their centre, whereas centre agents listen to their own platoon agents and all other centres.

A list of all message types and their data follows.

Dig this target A command from the ambulance centre telling all ambulance teams to rescue a particular target. Data: ID, position, time.

Need rescue A request from a trapped rescue agent to be rescued. Data: ID, position, buriedness, time.

Request road clear A request from an agent to have a particular road cleared. Data: ID.

Road update Notification that a road's blockedness has changed. Data: ID, blockedness, time.

Building update Notification that a building's fieryness has changed. Data: ID, fieryness, time.

Humanoid update Notification that a humanoid's status has changed. Data: ID, position, buriedness, damage, hp, time.

Building searched Notification that a building has been searched for trapped civilians and does not need to be searched again. Data: ID.

Extinguish this target A command from the fire station telling all fire brigades to extinguish a particular target and any buildings nearby. Data: ID.

Patrol assignment A command from a centre instructing a platoon agent to search buildings around a particular object on the map. Data: ID.

The various notification messages are sent when an agent notices a change in the environment. In addition, the centre agents will pass on these notification messages to all other agents if the notification resulted in a change in the centre's world model.

3 Path Planning

Path planning for all agents is based on an A* algorithm using the Euclidean distance heuristic. Roads that are known to be blocked are considered impassable. If no paths to a target can be found, the shortest route with blocked roads is chosen and a request for the roads to be unblocked is made. The exception is PCPlod, who always chooses the shortest path to the goal and clears any blockages.

3.1 Reachability Sets

Only paths that are known to succeed are actually planned by our agents. To assess whether an agent can in fact reach a given target we use *reachability sets*. Reachability sets are sets of mutually reachable nodes, i.e. all pairs of nodes in the set have at least one unblocked path between them. When a road changes state (i.e. we see a new blockage or a blockage is cleared) these sets are adjusted. If a blockage is removed and its head and tail node are in different reachability sets we merge the two sets. If new blockage is found then a breadth-first search is performed on the set containing the nodes of that road, and the set is split if necessary. These computations are performed once at the beginning of each timestep and are guaranteed to run in time linear in number of nodes.

3.2 Traffic Jams

In addition to basic path planning, a simple traffic jam resolution scheme is included. If another agent is known to occupy a location on the path they are assumed to be creating a traffic jam. In this case, the location is considered impassable during path planning for this timestep. In addition, if an agent attempts to traverse a path but stays at the same location for more than one timestep then it is considered to be in a traffic jam and that road is marked as impassable for the rest of the timestep. This allows for an alternative path to be planned immediately, while the road can be used again in future without requiring further observation.

4 Agent Strategies

All three platoon agents are implemented as finite state machines. The agent logic is built as a hierarchy of states, with each state delegating to its first child that is able to execute. These states vary in complexity from simple wrappers for commands to states using high level spatial reasoning to decide which child state to query.

4.1 Fundamental States

The leaf nodes of the state tree hierarchy simply wrap an agent-kernel command, we call these the fundamental states.

Move State This state contains a path from an agent to a location. After each execution it will determine the agent's progress on the path so far and crop the path accordingly. Failure to execute is broken down by this state into three possible reasons: either we already reached the goal, we encountered a blocked road or we hit a traffic jam.

Rescue State This state checks the buriedness of a civilian and ensures that the location of agent and civilian are correct before executing a rescue command.

Load State Executes a load command on a given civilian. Falure to execute due to another agent loading this civilian first is detected here.

Unload State Unloads a civilian from the agent.

Extinguish State This state checks the distance between the agent and a building before extinguishing. Currently this state also checks the fieryness of the building to ensure we are not wasting water on an irrelevant target - this will change when the new simulator is used.

Clear State If a road needs to be cleared and the agent is at that location then this state will execute a clear command. There are two versions of this state: the Partial Clear State which ensures that a road has at least one lane free and the Total Clear State which ensures that all lanes are free.

4.2 Intermediate States

Above the fundamental states are a series of states that provide extra functionality or further reasoning.

Fixed Targets Move This state is initialised with a list of potential targets. At any point it can be queried and will set up its child Move State to move to the nearest reachable target. A typical use of this state is for moving to a refuge before unloading a civilian.

Random Walk This state generates a random path from the agent's current location and assigns it to its child Move State.

Buried If the agent is buried this sends a distress call to other agents.

Damaged If the agent is damaged this state delegates to a Fixed Targets Move state to take it to the nearest refuge.

Refill If the agent has no water left it will move to the nearest refuge using a Fixed Targets Move state. If the agent is already at a refuge it will remain there until it has sufficient water.

4.3 High-level States

These states provide high-level abstract behaviour for the agents.

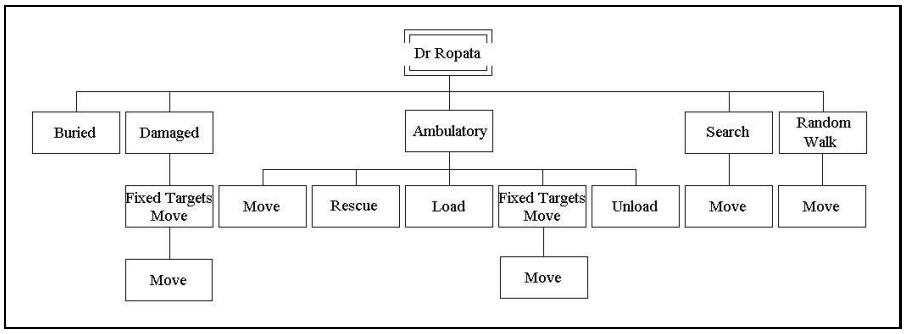

Fig. 1. Dr Ropata's state hierarchy

Ambulatory This state manages moving to a buried civilian (Move), rescuing (Rescue) and loading (Load) them, moving to the nearest refuge (Fixed Targets Move) and finally unloading them (Unload).

Search This state keeps track of all buildings searched by any agent. When it is asked to execute it moves to the best unsearched building. The selection of best building is based on its proximity to burning buildings and distance from the agent. This has a single child Move State.

Firefighting This state manages the extinguishing of a single burning building. This includes moving to a nearby unburnt or burnt out building (Move), extinguishing the target building (Extinguish) and refilling when low on water (Refill).

4.4 Dr Ropata

Dr Ropata is the implementation of the ambulance platoon agent. The primary duty of Dr Ropata is to decide which civilian to make the target of his Ambulatory State. Civilians that cannot be rescued by the combined efforts of all ambulance agents before death or timestep 300 are not considered valid targets. From amongst all known, valid civilians Dr Ropata chooses the most damaged civilian that he can reach and instructs his Ambulatory State to rescue that agent.

4.5 PC Plod

Unlike Dr Ropata and Fireman Sam, PC Plod does not have a high-level state to perform most of his reasoning. As the implementation of the police agent he merely has to decide where to move to, and his Clear state will clear any blocked roads as he reaches them. He will first move to any unreachable fireman or refuge, thus guaranteeing access to a refuge for all firemen. Next he will do

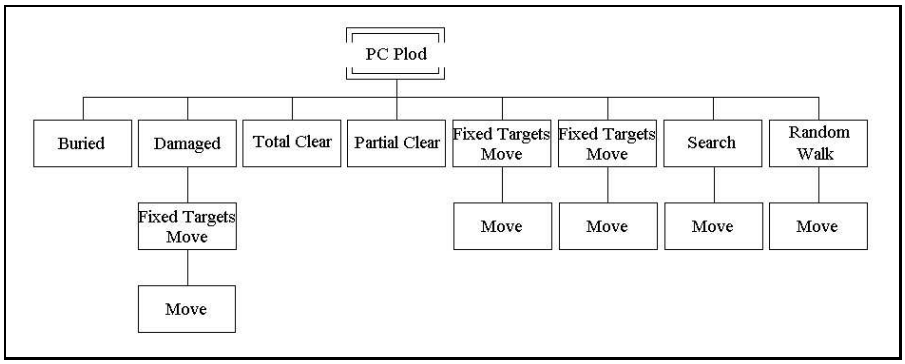

Fig. 2. PCPlod's state hierarchy

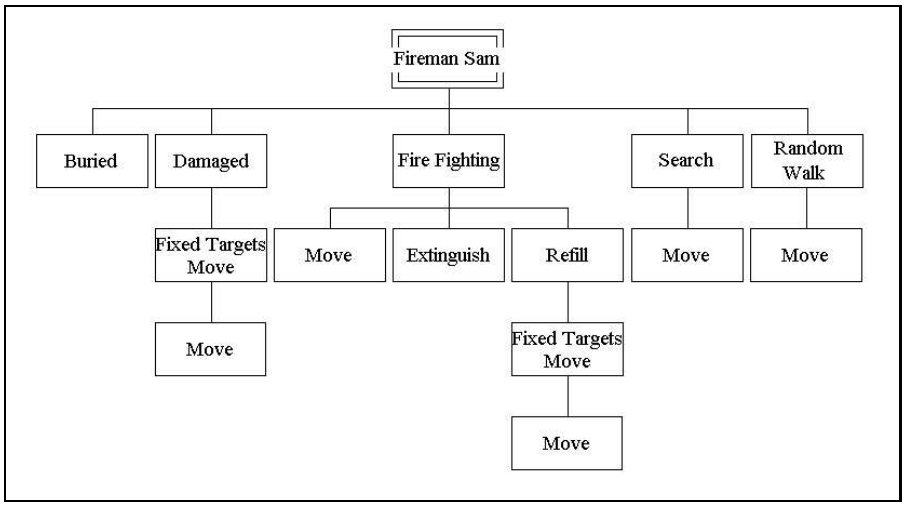

Fig. 3. Fireman Sam's state hierarchy

the same for fires. Once these important locations are accessable he will respond to specific requests from agents. When no roads are in need of partial clearing, PC Plod will switch to using a Total Clear State.

Each PC Plod is assigned a *centre of operations* which is a location within the city. Requests and clearing of important points are weighted towards those that are close to this centre. This helps prevent multiple agents from attempting to deal with the same targets and becoming inefficient.

4.6 Fireman Sam

Fireman Sam is the implementation of the fire brigade agent. This agent has the largest amount of reasoning outside its state hierarchy, all of which is used to select the next fire to fight. Fires are first clustered into *perimeters*. These

are connected sets of burning buildings with at least one unburnt neighbour. All unburnt buildings in the city are assigned a perimeter, given as the perimeter that will first expand to burn this building. The best perimeter is then selected based on the total floor area of its buildings (an approximation of the difficulty to extinguish) and the number of buildings it will first effect (an approximation of its deadliness). Once a good perimeter is chosen Fireman Sam will persist until either it is extinguished or another perimeter becomes significantly more important. Once a perimeter is selected the building with most firebreak (extinguished or burnt out) neighbours that is smallest and reachable is chosen. By considering those with most neighbouring firebreaks the agents will tend to create "walls" of extinguished buildings. By choosing the smallest buildings we attempt to reduce the size of the perimeter as fast as possible as these are easy to extinguish.

5 Conclusions and Future Work

The agents as described here still have some shortcomings. For example, there is no coordination between ambulance teams and fire brigades. It would be beneficial for the ambulance teams to work together with the fire brigades so that civilians who are near fires that are known to be out of control can be rescued first. Better coordination between the police agents and the other two types of platoon agent would also be desirable.

The problem of agents spending a large amount of time exploring different paths to distant targets also needs to be solved. An agent will prefer to go to a distance, high value target than to a near, low value one. Unfortunately, if there are many blocked roads between the agent and its target that we do not know about it is possible for the agent to waste several tens of timesteps trying every alternative route before finally giving up and returning to the low value target which by this time might have spread (if it is a fire) or died (if it is a trapped civilian).

Finally, it is possible that an efficient technique for planning paths to groups of targets might result in better computational performance by the agents. This technique is under development.

References

1. Fireman Sam. http://www.firemansam.co.uk (2003)
2. New Zealand Film Archive.
   http://www.filmarchive.org.nz/collections/collections_images.html (2003)