RoboCupRescue 2012 - Rescue Simulation League Team Description Ri-one (Japan)

Masatoshi Tsushima, Keisuke Yasuda
Sho Okazaki, Takahiro Nakagawa, Yuichiro Kurose

Ritsumeikan University, Japan
is0077er@ed.ritsumei.ac.jp
http://www.pj.is.ritsumei.ac.jp/ri-one/

Abstract. This paper describes the agents which Ri-one developed for RoboCup Rescue Simulation (RCRS) in 2012. RCRS is one of the multi-agent systems to simulate disaster reliefs. RCRS generates models of humans, buildings, roads, communications, fire and blockades, and reproduce virtually. The agents which we have to develop are called Ambulance Team Agent, Fire Brigade Agent and Police Force Agent. They have skills to use in rescue works. Therefore, we have to deal with the virtual situation by developing agents implemented efficient logic and algorithms.
The main idea of our rescue agents is determining priorities to entities. These agents act according to the priorities. The information for determining priority is given as the visual field from server, and the agents share with other agents using communication through the voice and the radio. The information which the agents having is managed as consolidated model called World Model. We must develop the World Model to make agents more efficient. Most of our changes in this year are related to the World Model. The first is the graph structure composed of lines and joining to calculate route. It also makes possible to reduce blockades that Police Forces clean. The second is partitions of the areas. It is used to generate efficient positions for Fire brigades. The third is a rough estimate of influence fires. This paper describes these and the application of these models to each our agent.

1 Introduction

RoboCup Rescue Simulation (RCRS) is one of the multi-agent systems to simulate disaster reliefs. RCRS servers simulate situations of various disasters. There are modeled human actions, communication, collapse of buildings, the occurrence of blockades and fires caused by the earthquake. The purpose of the agents that we have developed is to prevent the fire spread and rescue civilians. We have developed agents to solve this situation. The main problem of last year is the accuracy of the route search. Therefore, we have developed a new graph for the route search in this year. This graph is composed of points and lines. It is necessary to convert mutually the graph used for searching routes when we construct this graph in RCRS. In Section 2.1, we introduce the algorithm to solve this problem. In addition, we have developed Area Partition to treat adjacent buildings as a mathematical set. We quantify adjacent buildings by the distance between buildings, and create a set. Therefore, fire brigades can generate effective deployment. In Section 2.2, we introduce this Area Partition. In Section 2.3, we think about estimation of the fires. In Chapter 4, we introduce the use of these on the agents.

2 WorldModel

The role of World Model retains information to make better actions. We have developed Point of Visibility Navigation Graph, Area Partition and Energy Flow in this year.

2.1 Applying Point of Visibility Navigation Graph

When the agents move anywhere, they must send the list of areas to server. Information given to agents about areas is two-dimensional closed shapes and their adjacencies. Information about blockades blocking routes is also two-dimensional closed shapes and their belonging. Therefore, the agents must search routes in limited time. Point of Visibility Navigation Graph is the data structure which is generated with placing important points and their adjacency in area for searching[1]. We think it is possible to search accurate routes if the agents know whether the line segment in the graph has intersection point with shapes of blockage. However, it is problem to generate automatically to deal with various maps. Then, we developed the way to convert mutually.

In order to apply this graph to RCRS, we have to consider the relation between points and areas. As a result, start and end of path are connected with points which can determine the belonging area because the path to move is expressed the list of areas. However, it has not been solved yet. The line segments which must not become route come out when connect with points in areas simply, such as Fig. 1. Then, we add nodes to graph to solve this problem. These nodes do not have any areas and cannot become start or end. Therefore, an end result is the undirected graph which have two kinds of nodes, Finish-able node and Un-finish-able node. Finish-able nodes can become start and end of the path, and they must belong to an area. Un-finish-able nodes cannot become start and end of the path, and it does not have to belong to any areas.

The algorithm to generate Point of Visibility Navigation Graph is as follow:

1. Set a Finish-able node to all areas
2. List all pair of adjacent areas
3. For all pair of areas, define the middle point of the shorter line segment of two (refer to two buildings as A and B, edge of A to B and B to A) as Un-finish-able node
4. Relate Finish-able nodes and the Un-finish-able node mutually

This method makes new line segments from relation of nodes. Fig. 2 expresses these lines.

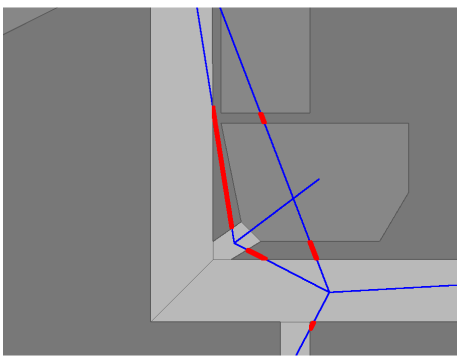

Fig. 1. Connecting areas simply

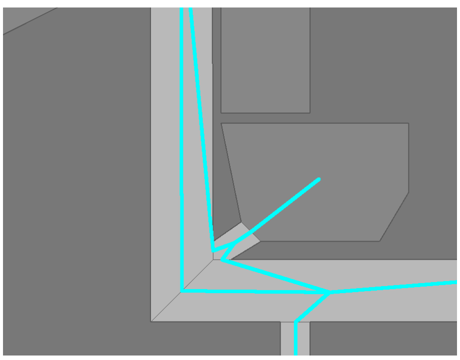

Fig. 2. Generated graph

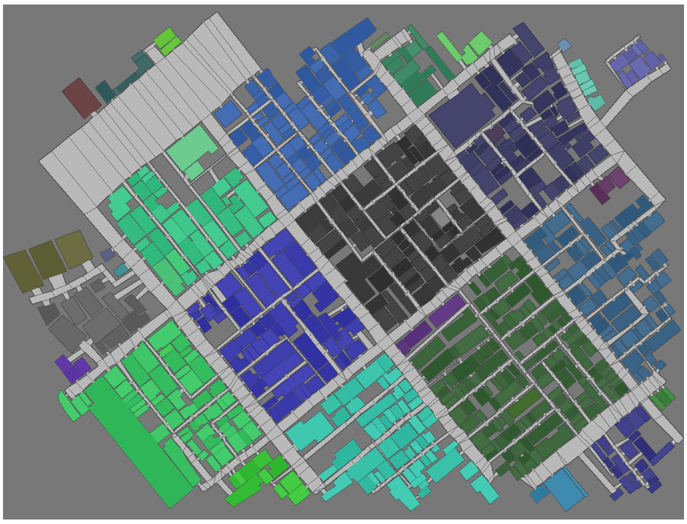

Fig. 3. AreaPartitions

2.2 Area Partition

In RCRS, information of the buildings is given as a set of individual buildings. Area Partition is intended to partition this set reasonable and to hold as Parted Area. Parted Area is constituted by own ID, a set of buildings, states and neighboring Parted Areas. This allows agents to work out efficient strategy for fire. In addition, this can communicate more information of fire with less traffic, and search civilian effectively. Fig. 3 shows the map which is color-coded by Area Partition. Each parted areas have different colors, coloring with Parted Areas color to buildings.

2.3 Energy Flow

Energy Flow is a part of World Model to estimate fires. This estimation suppose two things. One is that influence of other building's temperature is relative to temperature of the building. The other is heat spread in the form of concentric sphere, centering on the burning building equally. When a building whose temperature is t affects other building r meters away from the center of the building, and a surface area of sphere radius r is defined to be S, and a constant defined to be k, the influence I have relation because of a supposition as follow:

$$\oint_S I dS = kt \tag{1}$$

However, this I is the influence on inside of sphere. Then, an angle formed by the line from the burning building to intercept of affected building and effected edge is defined to be θ. An influence I of an infinitesimal surface of the sphere is as follow:

$$I = \sin\theta \frac{kt}{4\pi r^2} \tag{2}$$

The Fig. 4 shows this idea. We think it is a factor of priority to make more efficient actions and it will make possible to estimate whether a building is on fire by comparing with temperature of the building.

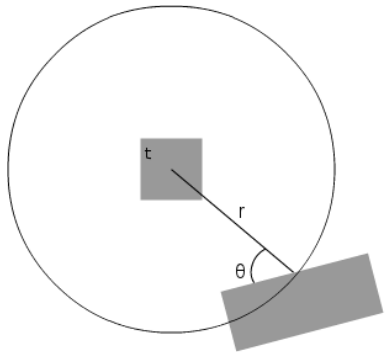

Fig. 4. Energy Flow

3 Communication

Communication with each agent is through server as voice and radio in RCRS. Our purpose of communication is avoiding redundant actions of agents. Therefore, we must consider what information should be share. Until last year, we divide communication to reading and writing such as Fig. 5. However, it is inconvenient to consider that if we use same architecture as last year because the module of communication depends on the whole World Model. Therefore, we have improved software architecture. In this year architecture, a pair of reading and writing exists per one lower model in the World Model such as Fig. 6 because this pair can become a unit to decide whether information about a lower model is should be share. The module for communication invokes methods of this pair at the right time, and then the agents generates section of bits for communication or applies the bits to model which receiving agents is managing. The header of bits of communication includes an assigned number which corresponds the lower model when the agents generate message. Conversely, it determines the lower model from a number in the header when the agents received a message. This change makes considering the communication more convenient.

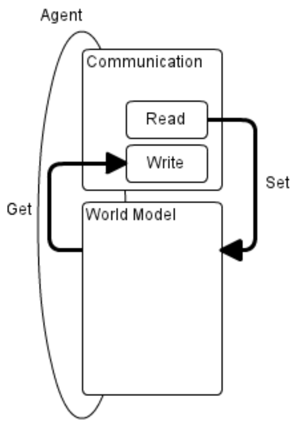

Fig. 5. Our legacy communication system

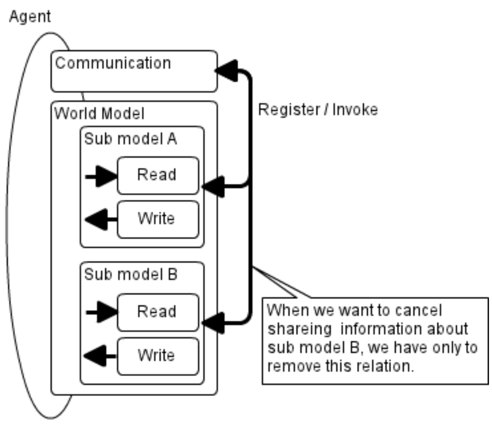

Fig. 6. Our current communication system

4 Agent Skills

4.1 Ambulance Team

Ambulance teams rescue victims buried in a building, and carry them refuge. All agents communicate location of the victims buried in blockade to Ambulance teams. Ambulance teams use the following information:

- Injured civilians
- Buried condition of the victims
- Status of the building fire
- The distance between a victim and an ambulance team

Ambulance teams use information of the above items as follows. First, victims contained in following items are excluded from candidates for rescue.

- The building with a victim is on fire
- A victim will be dead before ambulance teams carry to Refuge
- Ambulance teams are unable to rescue victims by blockades

Second, ambulance teams go toward the location of a victim who is in the shortest distance among them by using the Dijkstra method.

Then, there are cases such that multiple agents and victims existed in the same area. At that time, it is necessary to assign tasks to ambulance teams for efficiently rescue. The algorithm to assign we have developed is as follows:

1. If it is possible to load victims, assign an ambulance team in order of ID
2. Assign an ambulance team in order of ID to buried victims in shallow order
3. If it is impossible to load the victim at next cycle, assign an ambulance team again until ambulance teams end in the same area.

For example, we consider the case which is five ambulance teams and two victims exist in the same area, and each buriedness[1] of victims are $B_1 = 3$ and $B_2 = 4$, and ambulance teams can reduce 1 buriedness every cycle. Ambulance teams are assigned to the order by ID, $N_1 = 3$ and $N_2 = 2$. After this cycle, the each buriedness becomes $B_1 = 0$ and $B_2 = 2$. Then, next assignments are $N_1 = 1$ and $N_2 = 2$. Remaining ambulance teams go to search for other victims. Fig. 7 illustrates this example.

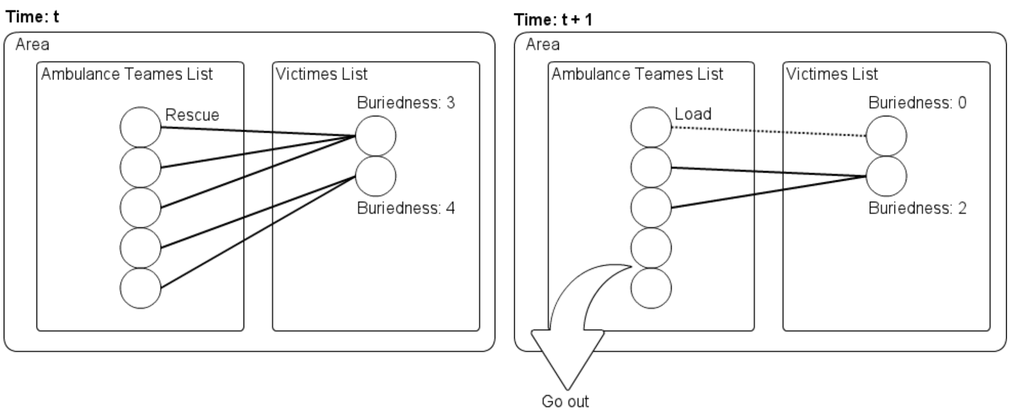

Fig. 7. Example of rescue

[1] buried condition rate of the victim

4.2 Fire Brigade

Fire brigades extinguish fires caused by disaster. They must protect more buildings from fires and reduce damage from fires. Then, it is necessary to extinguish all of the fire from limited information by sight. In other words, fire brigades need to confirm burned buildings around whether there are fires or not effectively. We solve this problem by using Area Partition in Section 2.2 and Energy Flow in Section 2.3. Specifically, fire brigades make enclosing a Parted Area burning and extinguish buildings. It can confirm more burning buildings around. Thereby, fire brigades get more accurate Energy Flow. Consequently, it gains them accurate extinguishing targets. Therefore, it can make fires spreading from burning buildings smaller. Fig. 8 is extinguishing Fire brigades with enclosing a Parted Area.

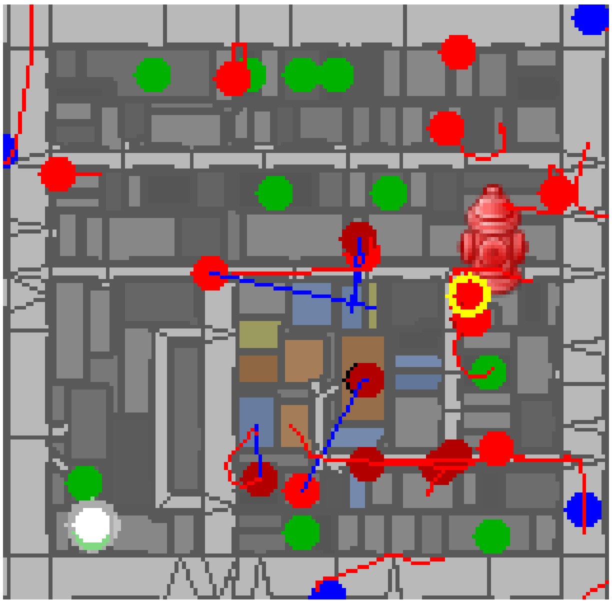

Fig. 8. Extinguishing Fire brigades with enclosing a Parted Area

Therefore, to extinguish using Area Partition and Energy Flow is smaller buildings damage than setting priority of fire buildings in last year. In addition, agents can move to Parted Area to have the oldest fire information for reducing the blind spot. Fire brigades have split into groups, they set a Parted Area for firefighting. It can allot fire brigades firefighting clearly. As a result, agents have in charge of their own place, and fire brigades can extinguish more efficiently to suppress the spread of fire.

4.3 Police Force

Police forces clear the blockade caused by the disaster. They must clear the blockade efficiently to help actions of other agents. The police forces of Ri-one set priority to entity, and move to entity of the highest priority with clearing the blockade. The method to decide priority is same as team of Ri-one in 2011[2]. We newly develop a method to clear the blockade efficiently on the path.

Police forces have to choose a blockade and decide amount to clear efficiently. In order to solve this problem, we use Point of Visibility Navigation Graph in Section 2.1 to decide a blockade which police forces clear. First, a police force computes the shortest path to target entity without considering blockades. Second, the police force considers the line segments which consist of the shortest path in cleared range. Then, the line segment belongs to the single area because of creating the graph, and it is judged whether intersections of the line segment and the shape of all blockades expanded fixed amount exist or not. When an intersection exists on the path, they clear the blockade. When an intersection does not exist on the path, they move along the path. After that, an intersection appears on the path and they clear the blockade likewise when they discover new blockade on the path. By this repeating method, they can reach target entity without clearing blockades which do not need to clear. Fig. 9 is the result of the algorithm applied. The line segments which do not have any intersections on the path are expressed by blue lines, and the line segments which have some intersections are expressed by red lines.

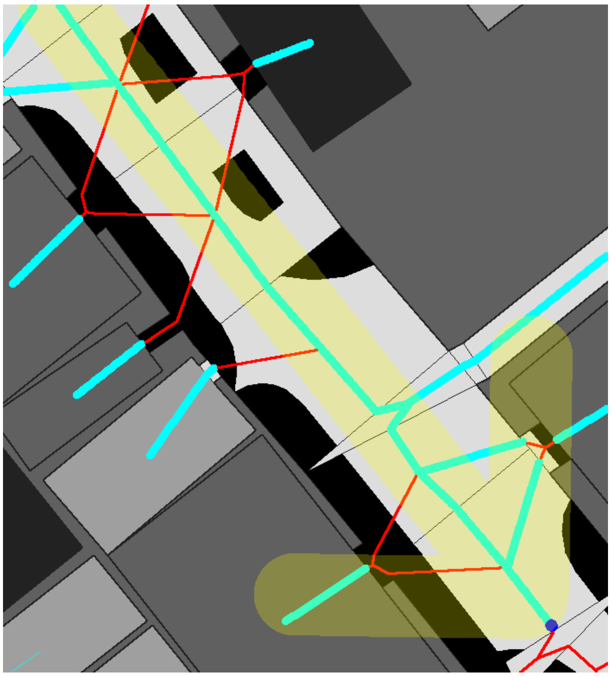

Fig. 9. The result of using Point of Visibility Navigation Graph

5 Result

Table 1-3 are the result of comparison with the agents in Ri-one 2011 at RoboCup 2010 Rescue Simulation League competition. The scores in some maps are increasing. We think it is caused by efficient rescues, extinguishes and cleanings. However, the some scores are decreasing almost slightly.

Table 1. Comparison of Preliminaries Day 1

	VC1	Paris1	Kobe1	Berlin1	Istanble1
Ri-one2011	16.44	110.78	90.17	22.69	7.34
Ri-one2012	28.19	138.91	124.95	22.13	7.25
Change	+71.47%	+25.39%	+38.57	-2.47%	-1.23%

Table 2. Comparison of Preliminaries Day 2

	Kobe2	Paris2	Istanbl2	Berlin2	VC2
Ri-one2011	158.10	60.45	74.60	47.78	70.29
Ri-one2012	145.47	61.17	75.43	31.16	96.78
Change	-7.99%	+1.19	+1.11%	-34.78%	+37.69%

Table 3. Comparison of Semifinals

	Paris3	Istanbul3	Berlin3	Kobe3	Istanbul4	Berlin4	VC4	Paris4
Ri-one2011	4.58	41.58	104.81	92.52	30.33	127.40	81.65	82.43
Ri-one2012	4.21	51.98	115.39	140.94	35.78	125.70	97.11	76.36
Change	-8.08%	+25.01%	+10.09%	+52.33%	+17.97%	-1.33%	+18.93%	-14.46%

6 Acknowledgment

We are deeply indebted to Gonkichiro Sumidai, Naoko Tanibata, Naoya Hikima and Kanako Watanabe for their advices.

References

1. M. Buckland: Programming Game AI by Example, 2005.
2. K. Watanabe, G. Sumidai, N. Tanibata N. Hikima, M. Tsushima, K. Yasuda, N. Suzuki, Y. Nakanishi and K. Kamei: RoboCupRescue 2011-Rescue Simulation League. Team Description. Ri-one(Japan), 2011.