RoboCupRescue 2011 – Rescue Simulation League

Team Description

<SEU_RedSun (P.R.China)>

Guan Daqi, Ouyang Jin, Wang Huijun

School of Mechanical Engineering, Southeast University

knight.guan@gmail.com

http://me.seu.edu.cn/sfzx/rescue

Abstract. After RoboCup 2010 Singapore, a lot of work has been done in our code. In this paper, a task-state based decision making method will be introduced. Meanwhile, a method based on minimum spanning tree for police force agent will be covered. As the appearance of new server, some close and relative research work has been done in the decision method of agents. Although certain features still need fully test, SEU_RedSun has gained 5rd place in RoboCup Open 2010 based on these new features.

1. Introduction

RoboCup Rescue Simulation System (RCRSS) is a large-scale Multi-Agent System (MAS) of urban disasters. In such a dynamic, partially observable environment, the action decision making is always the primary problems which need to be effectively solved. Our code structure is shown in the graph below.

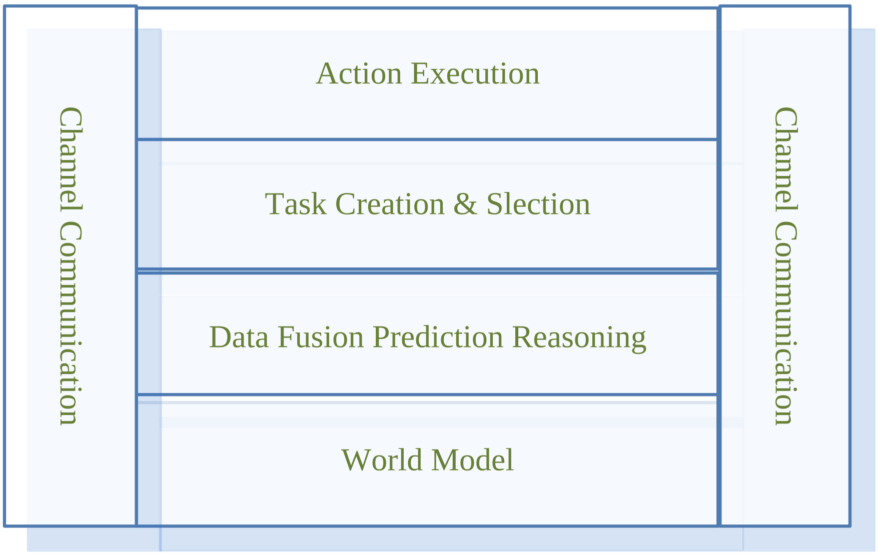

Figure 1. Code Structure of SEU_RedSun

The effectiveness of decision making needs a complete and accurate world modeling. So, we established different channel based communication models in diversified disasters for information sharing: the typical communication model and communication model under no center conditions. The latter model has some profitable characteristics such as adaptability, minimum time delay and virtually equally distributed channels. These characteristics especially enable us to build a more realistic world model under certain sharp conditions. As for decision making, we use task-state based decision method. A simplified decision process is shown in Fig.2. Basic low level action of moving is addressed to fulfill different needs of our agents in such a dynamic and uncertain system.

Task selection method of fire brigade is both distributed and centralized. We use particle filter method to predict the building temperature and then the center decide how many fire brigades needed to extinguish the fire of a fire site or a fire zone. As inspired from the real forest fire fighting, a fire zone based approach is used to slow fire spreading. The fire brigade receives tasks from the center and executes them. If it has completed the tasks and hasn't received new tasks, it makes decision itself.

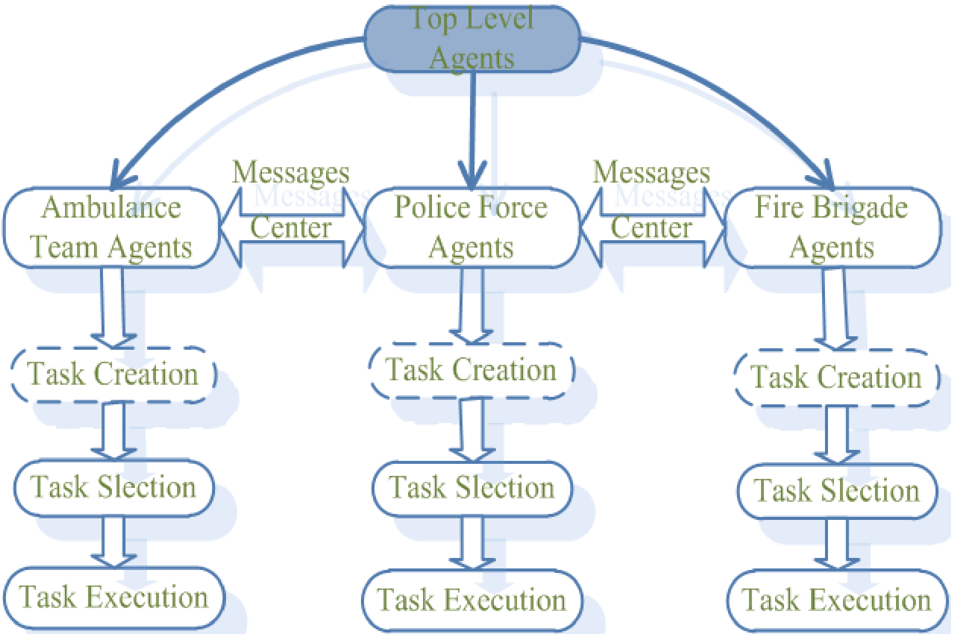

Figure 2 Module Structure of SEU_RedSun

Ambulances adopt totally centralized decision making approach which is treated as a dynamic assignment procedure: how to assign limited civilians to given number of ambulances in this dynamic environment. It can be very easy to understand that the total time for ambulances performing each successful rescue task is spent on two aspects: the time on road and the time on rescuing civilians. Besides, if we could estimate when civilians will be dead (Death Time), we will know the maximum ambulance needs for each civilians. Thus the assignment would be achieved. In order to estimate civilian death time, a particle swarm optimization (PSO) method is then illustrated. In the new scoring rule (Score Vector), the factor affect the result of rescue work has been divided into 6 different score vectors. We should integrate these factors to get an overall best solution.

Our main code structure is as fig.3. There are 12 modules in our code. The most important modules include communication module, world model update module, path plan module and top agent module. They are the basic modules to construct all the code. The knowledge base of task-state decision module is updated by the world model update module. The communication module helps to update the world and execute the agents' command. Path plan module is a basic module that every kind agent must use it to get a path to the destination. The BFS method is low efficiency, so we do some efforts to improve it. We use traditional a star method to explore the path. The top agent model is the agent task manage center. The 3 kind agents' common task is done in this module and the world model update module is also called in this module.

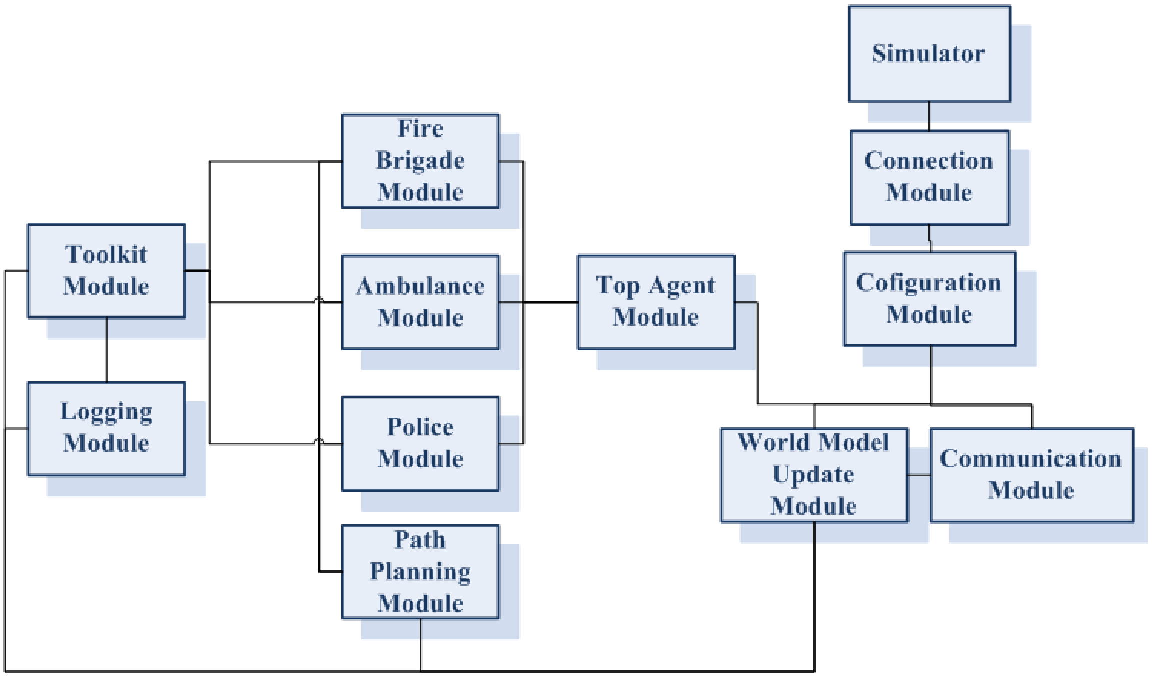

Figure 3. Basic Decision Process of Agents

2. Task-State Decision System

We adopt a task state system to help us to do the decision of agents' activities. A task include task object and task flag. Task object is what to do the task. Task state is a kind of flag that indicate what kind of activity the agent wants to do. Task flag indicates the task's attribute.

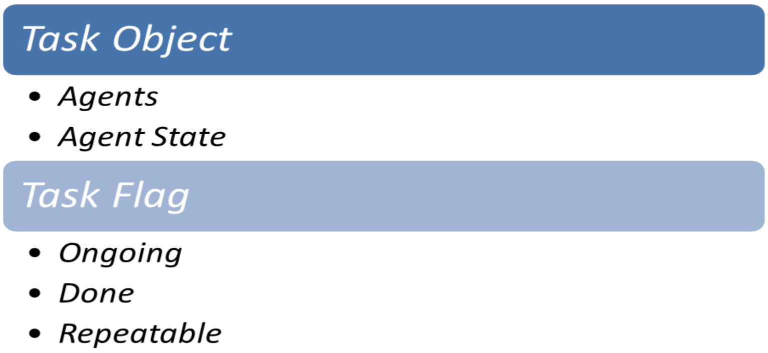

Figure 4. The Structure of a task

There are many kinds of agent state. Table 1 shows the common agent state.

Table 1. Common agent state

Common Agent State	
State	***Description***
Exploring	Map Exploring
Exploring Random	Random Map Exploring
Moving To Refuge	Return to refuge
Waiting	Waiting for task allocation & saved
Inactive	No task

SavingMyself	Save oneself
BuriedResting	Buried & waiting for saving
StuckResting	Stuck & waiting for saving
SavedByOtherAgent	Saved by other

Table 2 shows fire brigade state.

Table 2. Fire brigade state

FB State	
State	***Description***
GoingToRefugeToRefillWater	Going To Refuge To Refill Water
FillingWater	Filling Water
Extinguishing	Extinguish fire
MovingToFire	Moving to fire site

Table 3 shows ambulance team state.

Table 3. Ambulance team state

AT State	
State	***Description***
WillSaveHuman	Willing to save human
MovingToHuman	Moving to human
LoadingHuman	Loading human
UnloadingHuman	Unload civilian
RescuingHuman	Rescuing Human
TakingCivilianToRefuge	Take civilian to refuge

Table 4 shows police force state.

Table 4. Police force state

PF State	
State	***Description***
WillClearPath	Will to clear a path
ClearingPath	Clearing a path
MovingToBlockedRoad	Moving to the blocked road
ClearingCurrentRoad	Clear current road
ClearingBlock	Clear Blockade
ClearingRefugeEntrance	Clear entrance of refuge
ClearingAllBlocksAtLocation	Clear all blockade at some place

The task flow of task-state system is showed as fig.5 below.

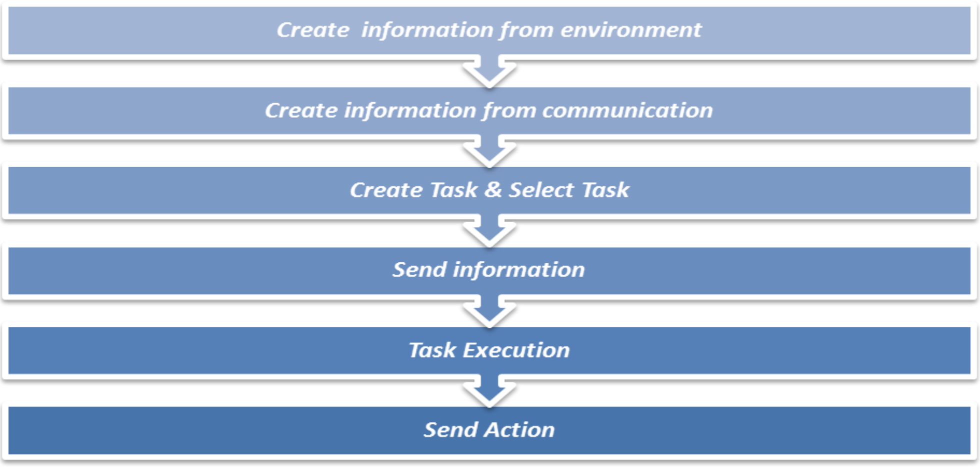
Create information from environment
Create information from communication
Create Task & Select Task
Send information
Task Execution
Send Action

Figure 5. The agent task flow

3. Agent Task Selection Method

3.1 Fire Brigade Agent

Prediction of fire spreading is crucial for fire brigade agents controlling fires. By the estimation of buildings temperature, we can get a more reasonable task allocation plan.

The building temperature update process building temperature include update of building fuel, update of building energy, update of building water quantity and radiation process. The specific process is as follows:

1) *Update of building fuel*

$$CF(i) = IF \times f(i) \times Rnd \tag{1}$$

$CF(i)$ is the cost of fuel in time step i, IF is the initial fuel of the building, $f(i)$ is the cost coefficient, it is not less than 0.005. Rnd is a random float number between 0.8 and 1.2. $f(i)$ is calculated as follows:

$$f(i) = \frac{T(i)}{1000} \times \frac{F(i)}{IF} \times 0.2 \tag{2}$$

$T(i)$ is the temperature of building in time step i, $F(i)$ is the fuel of building in time step i. If $f(i)$ <0.005, $f(i)$ =0.005.

The initial fuel is calculated by:

$$IF = \rho_f \times V \times Rnd \tag{3}$$

2) *update of building energy*

Energy update process of building includes the decrement of building's fuel and the increment of building's energy. The update process is as follows:

$$E(i) = E(i) + CF(i) \tag{4}$$

$E(i)$ is the energy of building in time step i, like equation 8, the fuel of building decreases by:

$$F(i) = F(i) - CF(i) \tag{5}$$

If $CF(i)$ is bigger than $F(i)$, then $CF(i)$ equals $F(i)$.

3) *update of building's water quantity*

The effect of water is related to the energy owned by the building. If the effect of water $effect(i)$ is smaller than the energy difference of building $\Delta E(i)$, the water will evaporate; When $effect(i)$ is bigger than $\Delta E(i)$, the real effect of water and energy consumption should multiple a coefficient $pc(i)$.

$$\Delta E(i) = \left[T(i) - T_i\right] \times \Gamma \tag{6}$$

T_i is the initial temperature of the world and equals $20^\circ C$. Γ is the heat capacity of the building.

$$pc(i) = 1 - (effect(i) - \Delta E) / effect(i) \tag{7}$$

If $effect(i)$ is smaller than $\Delta E(i)$, The water quantity of building is zero. The energy of building is calculated by:

$$E(i) = E(i) - effect(i) \tag{8}$$

If $effect(i)$ is bigger than $\Delta E(i)$, the energy of building is zero. The water quantity is calculated by:

$$WQ_B(i) = WQ_B(i) \times (effect(i) - \Delta E) / effect(i) \tag{9}$$

$WQ_B(i)$ is the water quantity of building in time step *i*.

4) radiation process

The radiation energy of building can be calculated by Stefan-Boltzmann law:

$$T_{rad} = \sigma \times (T(i) + 273)^4 \times S \tag{10}$$

σ is Stefan-Boltzmann constant, S is the area of buildings' wall.

Particle Filter (PF) base on Monte Carlo method. We can use the frequency of an event instead of the probability of the event. Here is the state equation:

$$x(t) = f(x(t-1), u(t), w(t)) \tag{11}$$

$$y(t) = h(x(t), e(t)) \tag{12}$$

Equation 11 is state transfer equation and equation 12 is the observation equation. *x*(*t*) is the state of time *t*. *u*(*t*) is control value. *w*(*t*) and *e*(*t*) are model noise and observation noise. *y*(*t*) is the observation value of time *t*. Goal of Particle Filter is from observation value *y*(*t*), state *x*(*t*-1*)* of time *t*-1 and control value *u*(*t*) to filter out the true state *x*(*t*).

First of all, we should generate a large number of sampling based on the state *x*(*t*-1*)* of time *t*-1 and its probability distribution. The sampling is called particles. These particles distribute on state space with the probability distribution of *x*(*t*-1*)*. We evaluate all the particles with conditional probability $P(y \mid x(t)^i)$. It means the probability of getting observation value y when you choose particle *i* in true state *x*(*t)*. The conditional probability is the weight of particle *i*. We filter particles with low weight with re-sampling method. Particles left will be the true state *x*(*t)*.

3.2 Ambulance Team Agent

The performance of ambulances could greatly affect the final scores. We treat the process of rescuing civilians as a dynamic assignment problem: given *m* ambulances and *k* civilians. We use Particle Swarm Optimization (PSO) to complete the allocation. The whole algorithm description is in our team's team description 2010. The new scoring method (Score Vector) adds a lot of factors into decision process. From the table 1 behind, we can see score vectors and their influence on ambulance teams' decision factor.

Through table 5, we can see that civilians' lives have been divided into 4 levels. Two positive levels and two negative levels are divided by 4000 HP(Health Position).

Table 5. Civilians' Lives and Their HP

Factor	HP	Influence on The Score
Healthy Humanoid	$7001 \le HP \le 10000$	+
Humanoid with average health	$4001 \le HP \le 7000$	+
Humanoid with critical health	$1001 \le HP \le 4000$	-
Dead humanoid	$HP \le 1000$	-

So the decision objective is not only the most civilians, we but also need to rescue more healthy civilians. We find that civilian whose HP is lower than 4000 in the situation using score vector is just as the civilian whose HP just over 0 in the old situation not using score vector. We are using the same algorithm as 2010.

4. Communication

Communication module is a very important module to support world model update and the agent action execution. In the latest server, the RCRSS use variable channel count to replace fixed channel count. If the communication mode is fixed channel count, the distribution mode is based on the number of specific center. Now the channel count is variable, center is as an ordinary agent.

We can give a definition of the communication problem. To connect agents by P2P channel, to achieve message sharing and task allocation, minimize message quantity and maximize information sharing rate. The constraint condition include agent number, center number, channel count, available channel count, message size, utter number, false rate and dropout rate.

We use available channel count to allocate agents and channels. The definition of available channel count is as equation 13. The channel distribution method is described as below:

$$Available\ \ Channel\ \ Count = Channel\ \ Count \times (1 - Dropout\ \ Rate) \times (1 - False\ \ Rate) \qquad \textbf{(13)}$$

$$Group1 : Group2 : Group3 = C(Channel1) : C(Channel2) : C(Channel3) \qquad \textbf{(14)}$$

The proportion of group 1, group 2 and group 3 is equal channel 1, channel 2 and channel 3. It is showed as equation 14. The distribution method is as fig 6.

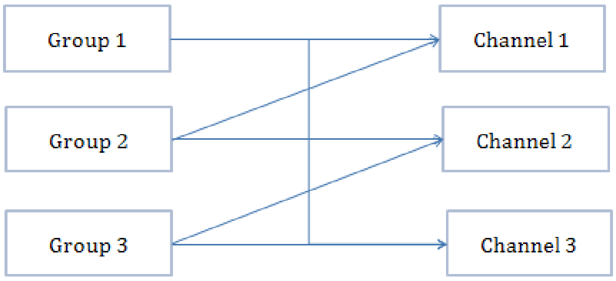

Figure 6. Group and Channel Distribution

a. Groping agent by the available channel count proportion

b. Creating the relationship between channel and agent group

c. According the channel quantity agent can use, send information to the corresponding channel and the upper channel

Some important messages are listed as table 6. These messages are important to support agent task creation and selection, but they are not all.

Table 6. Some important messages

Type	Description
Position	Current agent position
LockedByBlockade	Blocked agent information
UnLockedByBlockade	Blocked agent saved information
Fire	Building's fire information, including Fieriness, temperature and material.
Road	Road state, including
EmptyBuilding	Exploring buildings
Victim	Victim information
FoundCivilian	Found civilian information
HeardCivilian	Heard civilian information

5. Conclusion

In this paper, we presented a brief overview of the structures and approaches designed and implemented in SEU_RedSun after RoboCup 2010. First of all, we want to build a complete and accurate world model via communications among different agents. Second, we adopt a new task-state decision system to improve our agents' decision. Finally, various techniques have been tried or implemented in our code to deal with noisy, varied, real-time and dynamic disaster environments.

For the future, we plan to thoroughly test our code, modify minor bugs and use other Artificial Intelligence methods in order to establish an effectively cooperative team of agents in such a complex multi-agent domain to diminish the side effects of urban disasters.

6. Reference

1. RoboCup-Rescue Simulation Manual, RoboCup Rescue Technical Committee (2000).
2. Guan Daqi, Chen Ning, Jiang Yinhao. SEU_RedSun 2010 Team Description.
3. Meng Qingfa, Guan Daqi. SEU_RedSun 2009 Team Description.
4. Guan Daqi, Xu Yingqiu, Tan Yingzi. Particle Filter Based Study of Fire Forecasting in RCRSS.
5. Jiang Yinhao, Xu Yingqiu, Tan Yingzi. MST & SP Based Traffic Clearing and City Search in RCRSS.
6. Chen Ning, Xu Yingqiu, Tan Yingzi. Ambulance Team Algorithm Based on Multi-Evaluation Index in RoboCupResce.