RoboCupRescue 2016 – Rescue Simulation League
Team Description
<SEU_Jolly(P.R.China)>

Wei Niu, Jie Wu, Cheng Qian
School of Mechanical Engineering, Southeast University

Abstract: In this paper, we will show different strategies of ambulance in different period of stimulation to save civilians. Meanwhile, an approach of fire brigade is put forward to protect the important area when the fires are beyond control. A simple log commands is adopt to write down the job of agent. Finally, something about Multi-agent is put forward.

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 needs to be effectively solved. Our code structure is shown in the graph below.

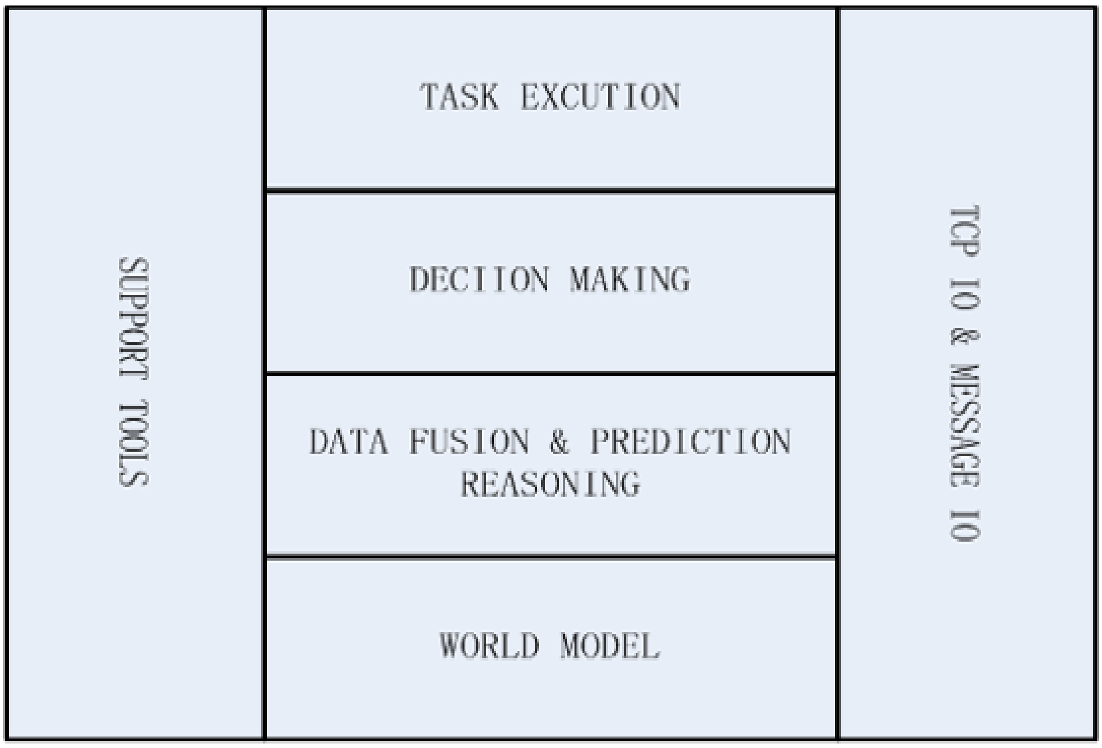


Figure 1 Code Structure of SEU_Jolly

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.

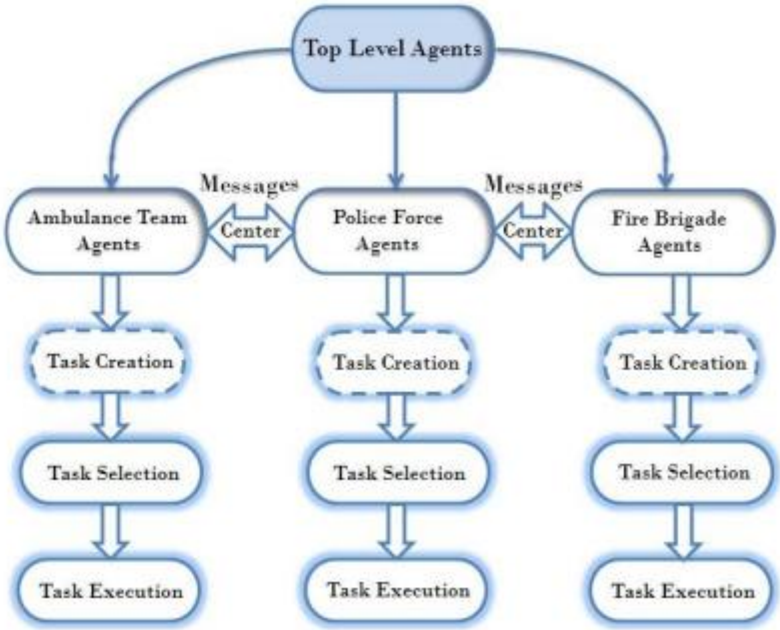


Figure 2 Module Structure of SEU_Jolly

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 help s 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.

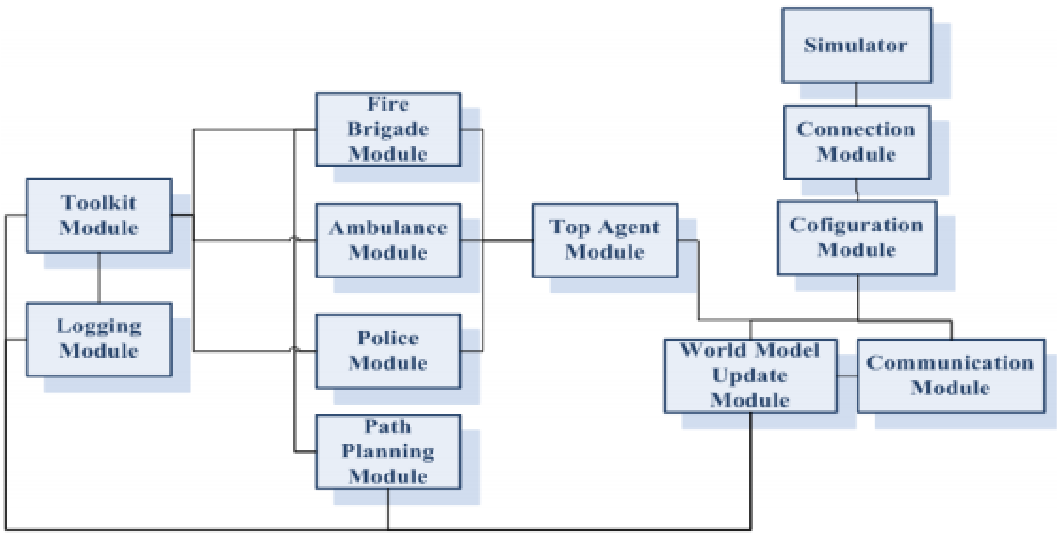


Figure 3. Basic Decision Process of Agents

Ambulance Team

ATs tend to work within partitions. The simple partitioning method used before has now been replaced by K-means++ algorithm. The realization of K++ algorithm for finding K clusters is as follow:

1) Select one point as the first initial centroid randomly.

2) Calculate the distance between this centroid and any other point.

3) Choose a new initial centroid. A centroid with farther distance with any other point is of higher probability to be chosen.

4) Repeat step 2 and 3 until all initial centroids are selected.

5) Use these k initial centroids and run the basic k-means algorithm.

Normally, AT works separately, based on the assumption that each civilian can only be rescued by one agent. But, in the latter period of simulation, the health blood of the buried civilians will be low that one AT cannot save a civilian, because the rescue-time will be much longer than their left-time. So we adopt another strategy of selecting tasks in the latter period of simulation.

The procedure of selecting tasks:

1. Take all the wounded who is being rescued into consideration; if there is not enough ambulance for the target, then move there;
2. If there is no such wounded who is being rescued or there has been enough ambulance for the wounded, then reevaluate all the (other) wounded according to the injury and the distance to refuge. Choose the most valuable casualty as the new target.

In this way, in the latter period of simulation, they will tend to handle the same civilian. Thus ATs can gather together gradually as expected. This method can not only improve the rescue efficiency but also avoid occupying too much channel resources. As a matter of fact, it does not need any extra message as well. What’s more, as the method is refuge based, agents’ traveling time will be rather short.

Fire Brigade

The firefighting task can be decomposed into some subtasks, and the two most important subtasks are:

1- Have a clear understanding of the fire.

2- Allocate limited resources (fire brigades) to various fire sites.

During recent years, we set up our own simulator modeled on the mechanism of the fire simulator to estimate the spread of fire. Meanwhile, a cluster algorithm is introduced. In this way, we could implement an advanced thinking strategy based on fire cluster. For that the fire brigade would decide itself whether the fire should be put out or be controlled, or be ignored according to the condition of fire cluster.

We put out fire when fire cluster is controllable. When the fire cluster is large that we cannot put out, direction based strategy should be taken. In this way, we will split the map into 9 sections in a way that the fire site locates on center with another 8 sections surrounding it. Then give a value to each cluster based on its total area of unburned buildings, number of refuges and number of gas stations. This method is presented by MRL in 2014.

Sometimes, fires in the whole map are beyond control, in this situation the defensive strategy will be activated. Then FBs are demanded to gather together to one area as many as possible to stop the separation of fire in this area. The steps of defensive strategy are listed as follows:

1) Divide the map into 9 parts with K-means++ algorithm
2) Evaluate each part.
3) Select the nearest fire cluster to the most important area and implement the direction based strategy.

The effect of defensive strategy is shown in figure 4.

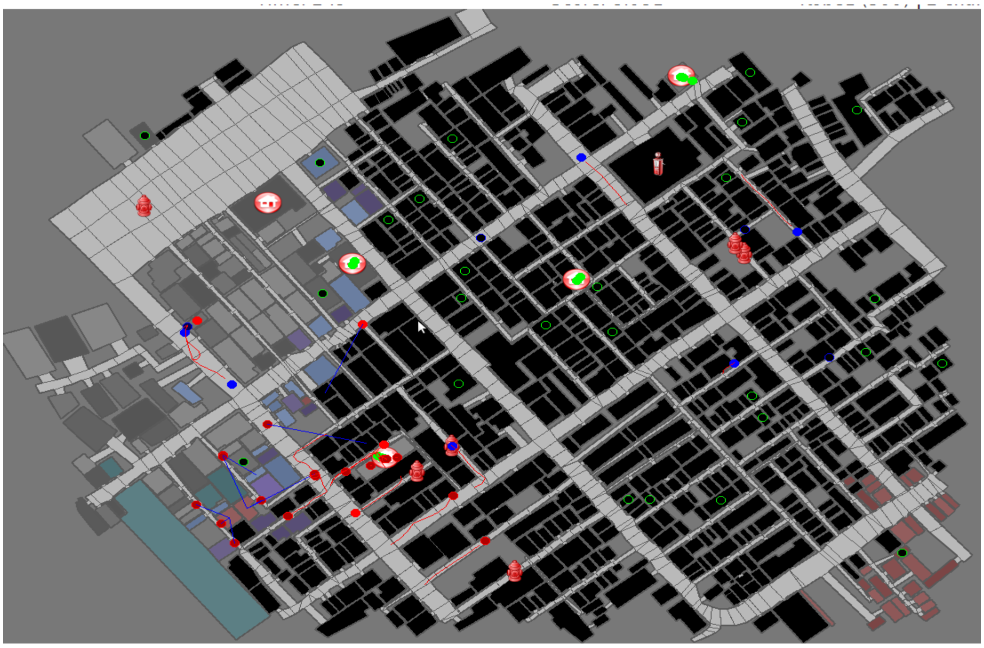
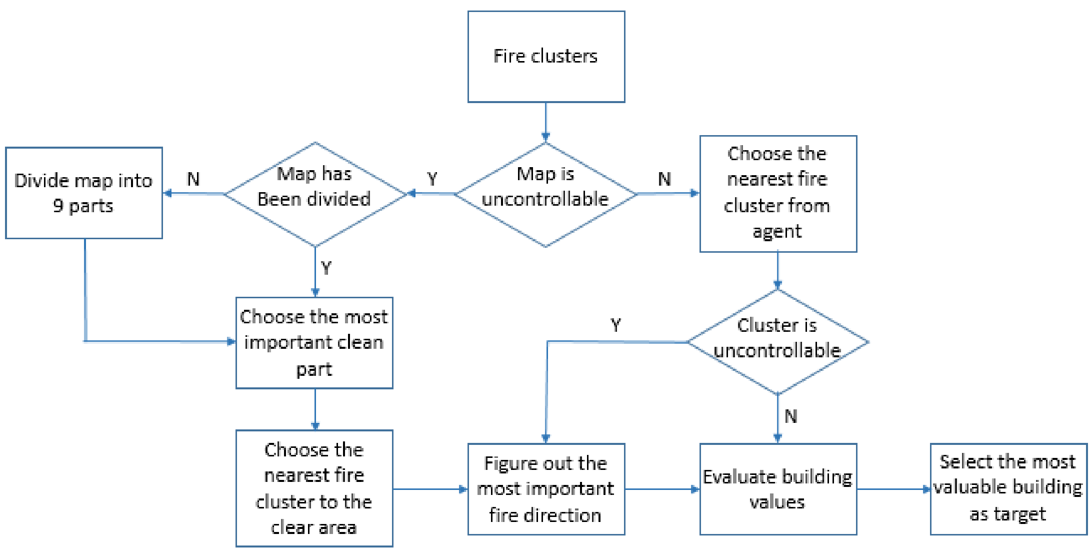


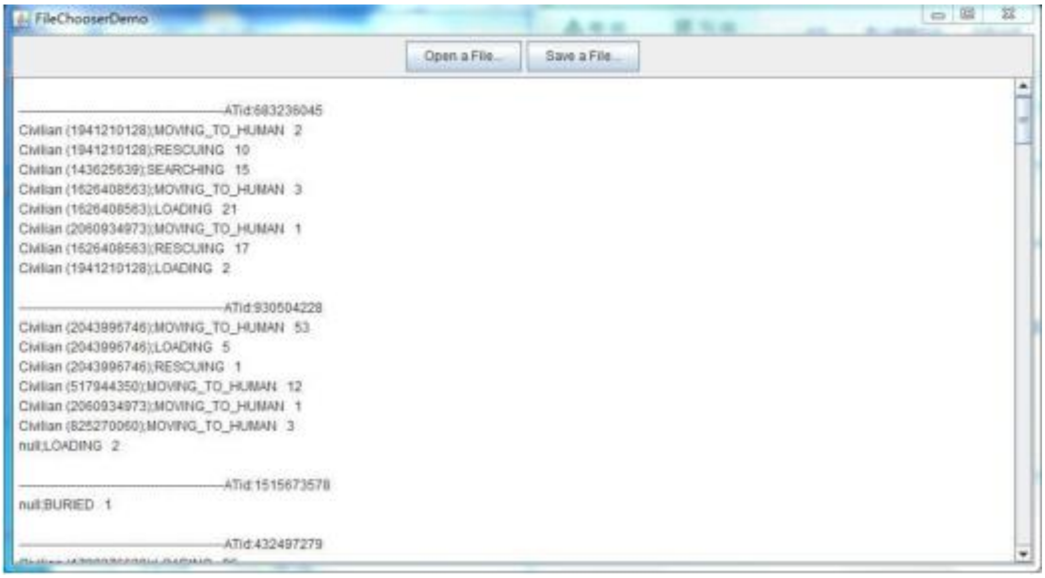
Figure 4

The total procedure of fire brigade is show below:



The logger

As described in the previous content, our agents' activities are based on what its task at that time. Sometimes, an agent may spend so much time on one task, which means it cannot handle some more important task, making the circumstance worser. However, as we know, it is difficult for us to know what exactly time for an agent spent on a task by just watching the viewer. That's why we need to write a logger to write down what the agent were doing at that time.



Thus, we put the log commands in the program and it can print out the agent's activities at each time cycle. Then we need to do some data analysis based on the log file. We also wrote a simple jar program to read the file and print out the data analysis result, based on the time that each agent spent on the task. Although it is a simple program, it helps us a lot to decide the strategies on the agent. After running a full cycle, we can summarize and evaluate each agents' performance and change some codes based on the results.

In the future, we are going to improve our programs and add more functions, such as ploting the summary, doing more complicate analysis.

Conclusion

In this paper, we presented a brief overview of the structures and approaches designed and implemented in SEU Jolly after RoboCup 2014. 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 domain to diminish the side effects of urban disasters.

Reference

1. RoboCup-Rescue Simulation Manual, RoboCup Rescue Technical Committee (2000).
2. Wei Niu, Shuang Cai, Jie Wu. SEU_Jolly 2014 Team Description.
3. Wei Niu, Jie Wu,Hangzhou Wan, Changwang Yuan, Kaikai Wang. SEU_Jolly 2015 Team Description.
4. Huang Zhourong, Guan Daqi, Shi Fabin. SEU_Jolly 2012 Team Description
5. Guan Daqi, Xu Yingqiu, Tan Yingzi. Particle Filter Based Study of Fire Forecasting in RCRSS.
6. Jiang Yinhao, Xu Yingqiu, Tan Yingzi. MST & SP Based Traffic Clearing and City Search in RCRSS
7. Pooya Deldar Gohardani, Peyman Ardestani, Siavash Mehrabi, Mehdi Taherian, Mostafa Shabani. RoboCup Rescue 2014 – Rescue Simulation League Team Description,MRL (Iran)