

Team Description of NITRescue02

Yoshiki Asai,Isao Higashide,Taku Sakushima,Nobuhiro Ito,Koichi Wada

Department of Electrical and Computer Engineering,
Nagoya Institute of Technology
Gokiso-cho,Showa-ku,Nagoya 466-8555,JAPAN
yoshikia@phaser.elcom.nitech.ac.jp

Abstract. Our team has so far participated in the RoboCupRescue[1] simulation competition. In the course, we created the class library of Java which supports development of an rescue agent[2].If an agent designer uses this library,it is expected that the burden which is needed for creating an agent can be lessened as much as possible. In our rescue agent team “NITRescue02” , using this library , the agent who performs cooperation between heterogeneous agents is creating.

1 Introduction

In the RoboCupRescue simulation, since the specification of a simulator being complicated and the system of a simulator were large-scale, the great labor was needed for agent manufacture. Then, we created like ADK[3] the class library by Java which supports agent manufacture. There are a method which deals with the complicated communication between kernels with an agent, and a method which can acquire various information in order that an agent may perform action determination in this class library. Therefore, an agent designer can do an agent’s creation at a few burden by using this.

In our rescue agent team “NITRescue02” , using this library , the agent who performs cooperation between heterogeneous agents is creating.

2 Java class library for RoboCupRescue agent

2.1 Outline

The composition of the Java class library which we created is shown in Fig. 1. The function of each module is as follows:

Router ... It is the module which collected the operation about movement. If the starting point ID and a terminal point ID are given , a route is returned.

Object Database ... It is a database for storing the object in rescue simulation space. The information which the agent has can be acquired by accessing this module.

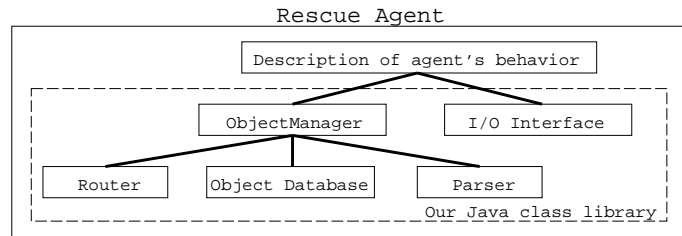


Fig. 1. The composition of the Java class library

Parser ... It is the module which consists of a function which extracts each object information from the data received from kernel and is stored in Object Database, and a function which interprets communication between agents and is held for every word.

Object Manager ... Based on the information acquired from three modules, Router, Object Database and Parser, the method which acquires the information which is needed when manufacturing an agent is offered. For example, the following methods are offered.

- goToDest(src,dest,blocked) ... The move course to src to dest is returned.
- getBlockedRoad(self,time) ... The information on the blockaded road is returned.
- getBuriedCivilianList(self,time) ... The information of the buried civilian is returned.

I/O Interface ... There is a method which can perform command transmission and reception with kernel without being conscious of the specification of a simulator.

An agent designer should just describe behavior of an agent using the module explained until now.

2.2 Example

The example of description of agent's behavior is shown. An agent designer can create easily the agent who acts according to the present situation using the method in ObjectManager. Moreover, when an agent designer wants to transmit a command to kernel, they can transmit a command easily using the method of I/O Interface.

```

// Is there any buried civilian or not?
int[] buriednessID = objectManager.getBuriedCivilianList(self,time);
if(buriednessID != null){ // There is a buried civilian.
    if(actionRescue(self.getPosition(),buriednessID)
        return;
}
  
```

```

boolean actionRescue(int pos,int[] civilList){
  
```

```

for (int i=0;i<civilList.length;i++){
  if (objectManager.hasCivilianObject(civilList[i])){
    Civilian c = objectManager.getCivilianObject(civilList[i]);
    if(pos == c.getPosition()){ // here is in the destination.
      io.rescue(c.getID());
      return true;
    }
    else{// here is not in the destination.
      int[] route = objectManager.goToDest(pos,c.getPostion(),true);
      io.move(route);
      return true;
    }
  }
}
return false;
}

```

3 Cooperative Action of Heterogeneous Agent

In RobocupRescue simulation, agents attempt in order to minimize the damage of disaster. However, depending on the degree of a disaster, it may be unable to be solved only by one agent's capability. For example, a fire company cannot arrive at the building which is burning by road blockade, and fire-extinguishing activities may not be able to be performed. In this case, if police force do not clear the road blockade immediately, a fire will spread further and a possibility that damage will be expanded will become high. Therefore, cooperation of a fire brigade and a police force becomes important.

In our team NITRescue02, using our Java class library, it aims at establishing cooperation work using an agent's communication. Moreover, capability to give suitable directions to an agent is implemented in fire station, ambulance center, and police office, and it aims at making damage of a disaster into the minimum.

References

1. RoboCup-Rescue Official Web Page,
<http://robomec.cs.kobe-u.ac.jp/robocup-rescue/index.html>
2. NITRescue Web Page,
<http://adam.elcom.nitech.ac.jp/NITRescue/>
3. Agent Development Kit,
http://www.tryllian.com/body_products.shtml