# SUNTORI:RoboCupRescue2011-Rescue Simulation Infrastructure Communication Library between agents and centers

Takefumi Ohta, Fujio Toriumi

Graduate School of Information Science,
Nagoya University
{takefumi@kishii.ss., tori@}is.nagoya-u.ac.jp
http://www.kishii.ss.is.nagoya-u.ac.jp

**Abstract.** The aim of the RoboCup Rescue Simulation is to find effective strategies for dealing with disaster situations. The system provides a platform for exploring ideas for a multi-agent system. However, it requires high-level programming skill to develop agents on RCR Simulation because of its complexity.

In this paper, we designed a communication library that supports communication between agents and a command center. By using this library, it is possible to develop a command center that gives orders to agents of various teams.

## 1 Problem in RCRSS

The RoboCup Rescue Simulation System (RCRSS) is one of the most complicated multi-agent systems, and presents an interesting area of study [1]. However, it requires high-level programming skills and knowledge of detailed rules. Therefore, it is difficult to operate agents (move to the destination on the shortest path, extinguish fire in a building, and so on) and to conduct cooperative work between agents (for example, help police force coordinate movement of fire brigade).

Though the aim of RoboCup Rescue Simulation is to find effective strategies for dealing with disaster situations, agent developers require certain skills to develop the agents freely before considering a rescue strategy. This is one of the reasons newcomers to the field are scarce.

Moreover, in disaster relief settings, it is possible that the control room must command not only subordinates who are usually leading but also workers from other regions (for example, a fire brigade from a neighboring town). For this reason, when considering real-life disaster relief, the command center must be able to lead any agent, and agents must be able to react to commands.

In RCRSS, most teams use their own protocols to achieve communication between centers and agents. Thus, when we try to use agents developed by other


**Fig. 1.** Examples of strategy and tactics

teams, we have to analyze the protocols of communication used in certain teams before developing new centers. However, generally, it is too difficult to analyze protocols used by other teams. Thus, it is impossible to develop a command center that can lead these agents.

## 2 Communication Library

To solve the problem described above, we designed the communication library.[1] By using the library, it becomes easy to separate the development of the command center and development of the agent. In other words, we can easily separate strategy and tactics. "Strategy" here means the overall policy in the virtual disaster space, and "tactics" means details for executing the strategy (for example, which fire should be extinguished). **Figure 1** shows examples of strategy and tactics.

By separating them, we can easily replace command centers and agents without considering communication protocols.

[1] Available at http://sourceforge.jp/projects/rcrscs/releases

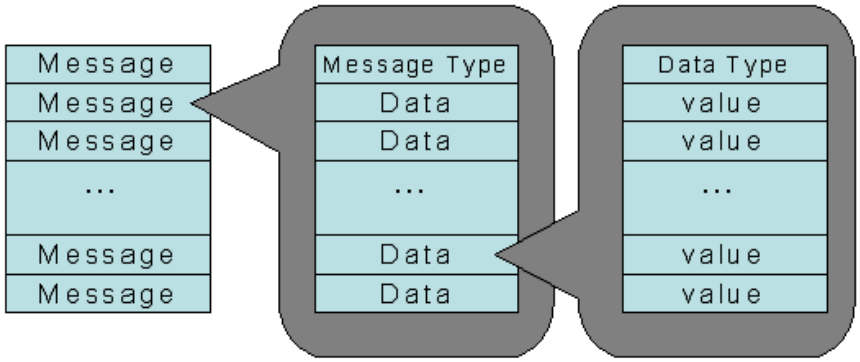


**Fig. 2.** Data Structure

### 2.1 Command Center

The command center gives the agent the directionality of action. For example, the command center gives an order to extinguish a fire at a certain site to some fire brigade agents. Notice that the command center NEVER gives orders about how to extinguish the fire, but only the location. This means that each agent has discretion about how to execute the orders.

The command center must integrate the information coming from each agent, and send back that integrated information to the agents.

Because command centers need to send both the strategy and integrated information, efficient communication protocols are required.

### 2.2 Agent

Agents determine their actions based on their perception and information sent by centers.

In a communicationable situation, agents must follow the strategy set forth by the command center. Even if the command center has conceived good strategies, if the agents execute the strategies poorly, the response effort will deteriorate. Therefore, it is important to develop agents that can perform orders efficiently.

However, in a non-communicationable situation, agents must be implemented to take action without any orders or integrated information.

### 2.3 Protocol

A conceptual diagram of the data structure in the library is shown in **Fig. 2**. The transmission data consists of some **Message**, and each Message consists of some **Data** with **Message Type**. Moreover, each Data is composed of **Data Type** and some **value**.

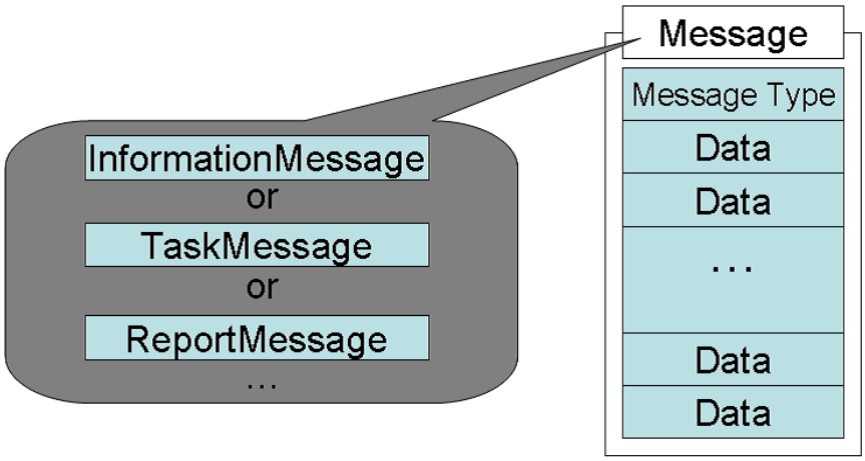


**Fig. 3.** Message

**Message**

The Message shows transmitted information (**Fig. 3**), and the Message Type expresses what kind of information the Message is.

Each Data shows the element of information included in the Message. Different types of Message have different numbers of Data. There are three main types of Message (**Information Message**, **Task Message** and **Report Message**).

The Information Message is the Message to transmit information obtained from the disaster space, such as brokenness and fieryness. The Information Message is used to share the information between agents and command centers.

The Task Message is used to send the order to agents from the command center. The agent that receives the Task Message acts according to the order included in the message. When an agent is instructed to do different tasks at the same time by two or more command centers, higher priority is given to the task sent by the center to which the agent belongs (for instance, if the agent is in a fire brigade, the command center is a fire station). Moreover, each agent is also able to send a Task Message to other agents for coordination in a command center-less situation. However, when command centers are communicationable, the Task Message sent by the command center always gets higher priority.

The Report Message represents a report that contains the result of the instructed strategy. For example, there is a Message named the Exception Message, which reports that the instructed strategy was impossible to achieve in the agent's situation.

**Data**

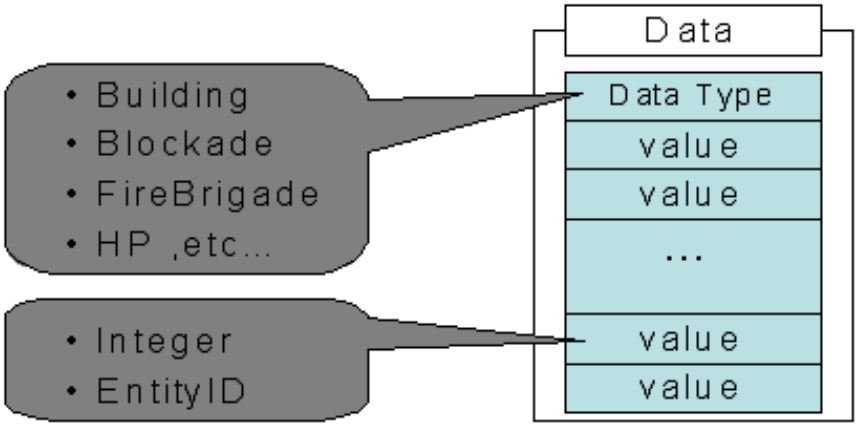


**Fig. 4.** Data

The Data shows the element of information included in the Message (trapped civilian, building on fire, and so on)(**Fig. 4**).

The **Data Type** expresses what kind of element the Data is, and **value** shows an actual value of the Data. Only Integer or EntityID can be substituted for value. Different types of Data have different numbers of value. For instance, some Data have one element of an integer that shows physical strength, and other Data have some element of EntityID in the Data, which show tracks of movement.

### 2.4 Types of Messages

In this section, we describe the details of each Message defined in this library.

**Information Message**

The Information Message shows information obtained from the disaster space. It doesn't include static information (outline of roads, gross area of building, etc.) to reduce the size of data. **Table 1** shows the types of Information Message and the elements included in it.

The Building Message represents information on the building that includes brokenness and fieryness. The Blockage Message shows information on blockages. The Victim Message contains information on civilians. The Position Message represents location information of each agent. The Transfer Message shows the movement locus of each agent. The FireBrigade Message, PoliceForce Message and AmbulanceTeam Message show information on each kind of rescue agent. Here, the items with an asterisk(*) in **Table 1** are optional items. The developer can include the item or not.

| Information Message | Elements contained |
|---|---|
| Building Message | fieryness,brokenness |
| Blockage Message | road ID,barycentric coordinate* ,repair cost |
| Victim Message | area ID,HP,buriedness,damage, position coordinate* |
| Position Message | agent ID,position coordinate |
| Transfer Message | agent ID,IDs of some area |
| FireBrigade Message | agent ID,HP,buriedness,damage,water quantity, area ID |
| PoliceForce Message | agent ID,HP,buriedness,damage,area ID |
| AmbulanceTeam Message | agent ID,HP,buriedness,damage,area ID |

**Table 1.** Definition of Information Message

| Task Message | Elements contained |
|---|---|
| Clear Route Message | pf ID,ID of work beginning area,ID of end of work area |
| Rescue Area Message | at ID,rescue work area ID list |
| Extinguish Area Message | fb ID,extinguish work area ID list |

**Table 2.** Definition of Task Message

### Task Message

The Task Message gives the order (e.g. help civilians in area). However, this Message does not order agent behavior directly. This type of message provides only the direction of the behavior. The type of Task Message and the included elements are shown in **Table 2**.

The Clear Route Message is the Message for police forces to clear blockages in a road connecting two areas. The Rescue Area Message is the Message for an Ambulance Team to find and rescue civilians in a specified area group. The Extinguish Area Message is the Message for Fire Brigades to fight fire around a specified area group. These Messages add flexibility to the actions the agent can take to execute the order.

### Report Message

The Report Message reports the results of each strategy. The types of Report Message and the included elements are shown in **Table 3**.

| Report | Elements contained |
|---|---|
| Done Message | agent ID |
| Exception Message | agent ID |

**Table 3.** Definition of Report Message

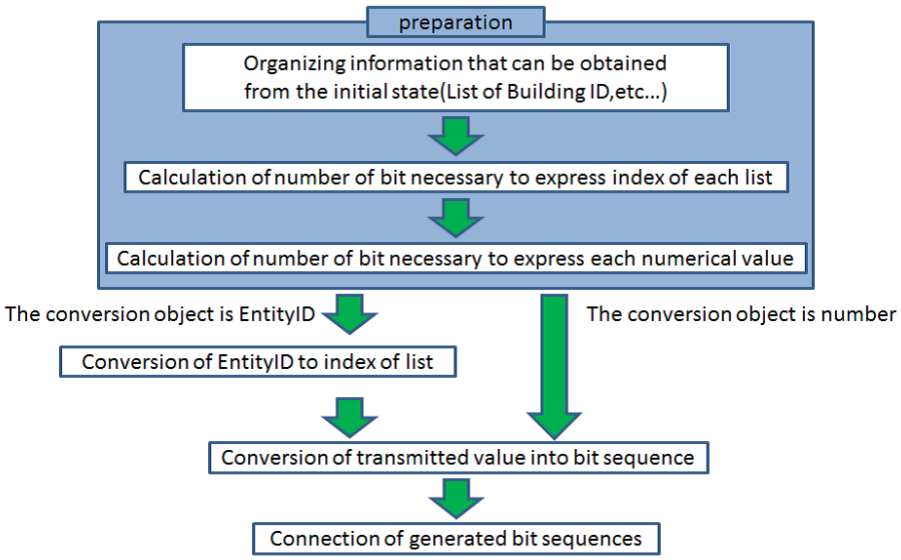


**Fig. 5.** The Process of CompressCompression Process

The Done Message reports that the given strategy is completed. The Exception Message reports that the ordered strategy was impossible to complete at that moment, e.g. Could not go to specified location and extinguish fire.

### 2.5 Data Compression

To send the Message defined in this section, how to compress the data is one of most important problems because of the restriction of transmission capacity [2]. If data compression is insufficient, it becomes difficult to send all Messages to share information sufficiently. The transmitted data compression process in this library are is shown in **Figure .5**. In this library, the following processes are applied as a preprocessing.

1. Make lists of EntityID for various Entities in disaster space such as buildings and rescue agents.
2. Calculate number of bits necessary to express index of the each lists.
3. Calculate number of bits necessary to express the value that whose limit is predictable (HP, damage, etc.)

The information which used for the preprocessing can be obtained from initial states, and the same result can be obtained for all agents and centers in this process. Compression and the restoration of the transmission data can be done smoothly by these this processing.

Additionally, the following process are is applied when each information are is compressed in this library.

1. If the conversion data is EntityID, it is converted into an index of the list which that is already stored.
2. Express each numerical value by bit sequence of the necessary length calculated in the preparation.
3. Connect bit sequences that are obtained, and send the result.

Because the total numbers of the agents, the buildings and so on do not change while during the simulation, it becomes possible to compress efficiently by theseefficient compression becomes possible with this procedure.

## 3 Conclusion

In this TDP, we proposed the a Communication Library during for the agent and the command center to separate the strategy and the tactics. When the team is developed by using the library, the command center plays the role to direct the policy of the action of giving orders to the agent. Thus, command centers can lean determine the most important region in the disaster space at that moment. We think it's possible to use learning algorithms such as SVM and the Bonanza Method for the command center [3, 4]. From the learning result, the command center can send more effective orders to the agents.

## References

1. *RoboCup Rescue Simulation Wiki*. http://sourceforge.net/apps/mediawiki/roborescue/.
2. *RoboCup Recue Simulation League Agent Competition 2010 Rules and Setup*. http://sourceforge.net/2010/rules.pdf, 2 2010.
3. C.J.C. Burges. A tutorial on support vector machines for pattern recognition. *Data mining and knowledge discovery*, 2(2):121–167, 1998.
4. K. Hoki. Optimal control of minimax search results to learn positional evaluation. *Joho Shori Gakkai Shinpojiumu Ronbunshu*, pages 78–83, 2006.