Proposal and Implementation of new framework to RoboCup Rescue Simulation NAITO-Rescue 2015(Japan)

Kazuo Takayanagi[1], Takuma Kawakami[2], Shunki Takami[1], Yusuke Kitagawa[1], Shivashish Jaishy[1], Nobuhiro Ito[1], Kazunori Iwata[3]
rescue-2015@maslab.aitech.ac.jp

[1] Department of Information Science, Aichi Institute of Technology, Aichi, Japan
[2] Department of Electrical and Computer Engineering, Nagoya Institute of Technology, Aichi, Japan
[3] Department of Business Administration, Aichi University, Aichi, Japan

Abstract. In recent years, the new entrants for the RoboCup Rescue Simulation (RCRS) are seen to have reduced. It can be believed to have occurred as for some existing problems in the Agent Development of RCRS. Hence, we present a design and its implementation in the agent framework as a development environment of the agent, known as Agent Developed Framework (ADF), which would also attract the new entrants.To carry out the development of the agent in a development environment(ADK:Agent Developer's Kit) that incorporates the ADF.

Keywords: Rescue Simulation, Multi agent, Agent development framework, Common Communication Protocol,

1 Introduction

Recently, the new participants for the RCRS have decreased. It is believed that this is a result of some problems that exist in the simulation (RCRS). Below are some of the problems described.

- The new entrants find the simulation difficult as there is very less information regarding the development of the agents.
- The simulator provided needs to implement the various algorithms in order to verify them.
- The time consumed for the comparison of algorithms is much longer.

Therefore, the aim of this paper is to solve the abovementioned problems designing and implementing the agent framework as a development environment of the agent, referring it as Agent Developed Framework (ADF). Creating manual and module of the agent. And a questionnaire was conducted to know the opinions about its practical use.

2 Problems of RCRS

To begin with, in RCRS there are three problems at the time of the development of an agent.

- The information in the manual lacks.
  The information regarding build of servers and agents is very less. The new applicants have almost no way to get any knowledge and are left with the options of javadoc or the materials that have already been published. Thus, this less choices of agent-building knowledge have been reducing the number of new partakers.

- It is a problem to share the source code.
  At present, in RCRS, each team implements its own communication protocol and every agent has own notation and different execution making it incompatible with other teams. As a result, even if the source code of an excellent agent is published on the Internet, it gets hard to be fully utilized in RCRS.

- Verifying the algorithm is a big problem.
  The problems like route search, research allocation, message sharing and real time planning with the complex issues are yet to be solved in RCRS, which makes it difficult to bring out an individual problem further challenging the comparison of algorithms. Further, various implementations of an algorithm are necessary in order to develop the algorithm with the agent as a base.

3 Design of an agent development support technique.

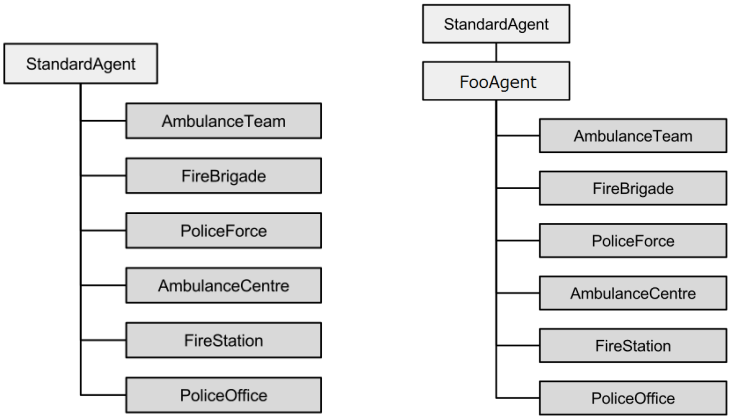
To resolve the problems, as mentioned in section 2, we would like to introduce the agent framework design as described below.

- The creation of a manual to match the current RCRS system.
  We hereby attempt to create the design and implementation of a manual for RCRS system, which would overcome the problem of lack of information in the current manual. We would require the information about the RCRS system to perform the designing and implementation of the agent framework. In addition, in order to prepare the sample source using the framework, it is necessary to have information about how to use the actual classes and methods.
- Design of the agent with high readability.
  It is known that the redundant parts of the agents have been an issue in the past. StandardAgent is a class of agent, which has a structure using generics[1] to obtain a versatility. The following Fig.1.(a) represents the class of the basic structure of the agent.
  The advantage of this structure is that summarized common method as Fig.1.(b) However, basically shareable part is only code which does not

[1] Function for associating particular type to a generic classes and methods.

depend on the type of agent, such as route searching and communication method.Furthermore, they can be independently as a class. In addition, due to the impact of generics the PostConnect method performs its own initialization of the Think method that determines performance of the behavior itself. Moreover, it is necessary to describe the method indicated by the agents themselves such as types of agents. Therefore, the specification using generics for new entrants can be considered difficult to understand. Because StandardAgent was repeatedly extends class by the specifications of the RCRS, available variables is not explicit. In contrast, the design of the framework in Fig.2. is composed of two parts.

In Fig.2., Tactics is a class that describes the behavior algorithm of agent. By using the Tactics, the redundant part of the agent can be separated from the behavioral algorithm, providing the readability. In addition, the route search and resource allocation in Fig.2. shows the interface for implementing each algorithm. This interface prevents the complexity of the source code of Tactics and is also possible to increase the variable readability.



(a) StandardAgent Configuration was utilized directly

(b) configuration using FooAgent with additional generic methods

Fig. 1. Block diagram of the agent of RCRS class

– The unification of specifications.

  In the current RCRS, source code compatibility is low and portability of algorithm is low. In contrast, the interface is used to implement the agent framework in each algorithm in Fig.2. As a result, to solve the problem related to sharing of source code and to have a readability and portability to algorithms is gained. Regarding the communication protocol between agents, previous studies refer to the existing Communication Library. In this study, we actively utilized Comlib, which performs shared communications.

– Design of agent that perform the basic operation.

  It is necessary to develop a verification algorithm for the comparison and

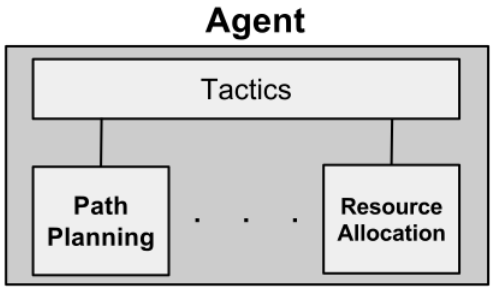


Fig. 2. Agent configuration diagram of the framework

verification to a particular problem. It requires enormous time to verify the algorithm. Thus, the basic behavior algorithm to implement the Tactics, to simplify the verification of the algorithm.

4 Implementation of the framework

Hereby, we perform the implementation of the suggestions and models mentioned in Chapter 3.

4.1 Creating a manual

Design and implementation of framework, and we were create a manual. To match the system of the current RCRS, manuals were described with regard to the construction of agent development environment.
In addition, indicated by the flow of the development of the agent the use of classes and methods is shown. As a result, the agent developer shall benefit with a variety of information.

4.2 Implementation of Tactics

The implementation of Tactics designed in Chapter 3 was performed. The main points that are summarized and shown in Fig. 3. The AbstractComponent or AbstractAgent, in class inherit from StandardAgent. StandardWorldModel holds a variety of disaster spatial information. The variables and methods defined for each class, it becomes explicitly by summarized as in Tactics in Fig. 3. Also, sendMove calls in the think method by combining the methods, such as class Action, keeping the readability of the Tactics.
The Tactics, TacticsAmbulance,TacticsFire and TacticsPolice is implemented. The explanation regarding the interface that implements the algorithm is given in section 4.6.
Also,by bringing together the sendMove method Action class is maintained readability of Tactics.

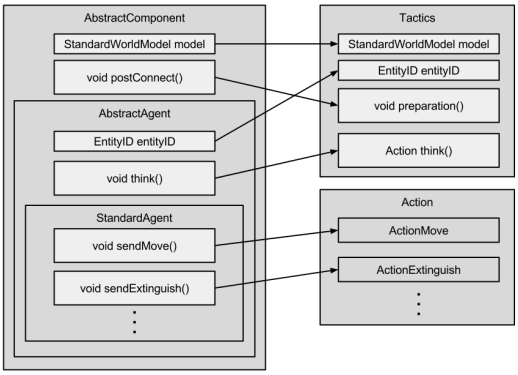


Fig. 3. Framework agent configuration diagram

4.3 Improvement and utilization of ComLib

ComLib, used by unified communications protocol, was applied to the current RCRS. Also attempted to improve the convenience by which the event-driven.

4.4 Implementation of Interface for Implementation of Algorithm

We have implemented about target selection algorithm and path planning algorithms to interface. The target selection algorithm is to determine the target rescue efforts based on resource allocation and damage prediction.

RouteSearcher

The RouteSearcher interface is that which implements the routing algorithm. In the RouteSearcher, algorithms such as Dijkstra and A* are possible to be implemented. In the ADF, implements the A * algorithm as a sample.

TargetSelector

In TargetSelector, Auction algorithms such as task assignments can be implemented by using the k-means algorithm. In the ADF a sample is provided implement the auction algorithm.

4.5 Implementation of the Tactics (basic behavior)

As an implementation example of Tactics, we have implemented a BasicTactics.RouteSearcher and TargetSelector has been incorporated into the BasicTactics.Also, for each interface implementation samples are provided, comparison algorithm is easier.

4.6 Package Configuration Framework

In this framework, shown in Fig. 4, the agent developers such as Launcher requires no part or portion required to develop such Tactics. Sample parts can

be used as the reference and comparison. We have realized that in the future for the development of the framework, we should keep in mind the manifestation and maintainability of the same.

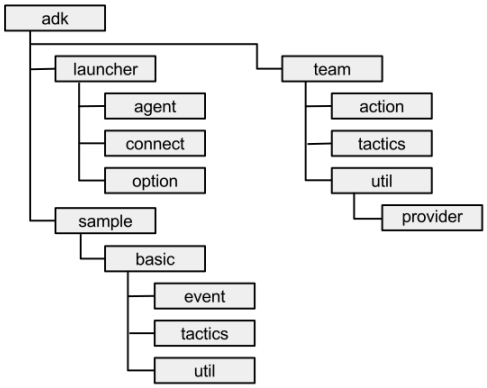


Fig. 4. Framework agent configuration diagram

Launcher package

This package contains class Launcher package that loads the agents.

Team package

In the ADF, the team Tactics of each agent is the system put together implementing a Team task. Team task is an abstract class, corresponding to each agent instance of the Tactics to generate a method or team name.

Sample package

In the current sample package the basic packages that include a description of the behavior of basic Tactics are included.

5 Surveys and Consideration

To confirm the validity of the framework questionnaire was investigated. Survey subjects were requested aiming at newcomers and other organizations.

5.1 The main opinions obtained by the questionnaire.

Survey results shown below are the main opinions obtained by the questionnaire.

- RCRS System Overview in manual, helped in understanding the it.
- When the agent development, framework of variables and methods were easy to understand.
- Because there is no prelude, I do not know the movement of the entire agent If I do not read until the end.
- World model is difficult to understand.

– Less information about the Entity.
– I want you to implement the algorithm of the past world champion team in the framework.
– Share the sample code in wiki, and I want you to be able to implement algorithm roughly by combining a sample code.

5.2 Consideration

About Tactics

It was possible to obtain the opinion that easy to understand by summarizes the variables and methods to Tactics.From this result, approach is considered to be effective..

About manual

Because some lack part was pointed out, improvements are needed.

About source sharing

We ware not able to obtain an opinion about

6 Summary

This paper describes how to build a development environment and to show how the agent develops. The aim was to facilitate the new entrants with a better Agent Framework for the design and implementation of an agent.
In this ADF, by eliminating the redundant part the current RCRS system, aimed at environmental construction that developers can focus on the development of algorithms. We also tried to create a manual with as much information on the development of how to use the classes and methods.
In the questionnaire regarding our RCRS agent development for beginners, we were able to obtain the opinion that the Tactics could be valid for the RCRS system. However, manual could not as a overall to obtain satisfactory results. From these, if it is possible to expand the information of manual, it becomes easy to use RCRS, it is expected that the environment that can focus only on the agent development can be realized.

References

1. Dai Obashi, Yuki Kawaguchi, Shogo Horibe, Takehumi Ota, Nobuhiro Ito, Toriumi Hujio, "Information shared library for RCRS".