

# RoboCup Rescue 2021 TDP Agent Simulation Ri-one (Japan)

Takafumi Horie 1\*, Kota Takehara, Hidetake Kondo, Takatsugu Nakayama, and Kosuke Yano

Ritsumeikan University, Japan  
rione.rescue@gmail.com  
<https://sites.google.com/view/ri-one>

**Abstract.** We focused on Agent development and improvement this year. Police forces (PFs) were implemented the removing blockade process to avoid agent's being stuck. In ambulance teams (ATs) and fire brigades (FBs), we introduced *triage*; classifying and prioritizing civilians, we realized flexible responses in selections of injured civilians and transportation to refuges. Furthermore, we implemented a dynamic target determination mechanism called *dynamic task management* in all agents. This mechanism led to a significant improvement in efficiency and a reduction in the amount of calculation. As a result, the score increased compared to the implementation in 2019.

## 1 Introduction

One of the most important issues in Robocup Rescue Simulation (RRS) is the flexible management of tasks by agents. Improving a single module alone is not enough to solve this problem. For evaluation and selection of appropriate targets, the following are required; accurate behavior in Action, flexibly evaluation criterion in Detector, and integration of the results through all evaluation criteria.

Ri-one 2019 focused on optimizing each of these evaluation criteria. In PFs, cooperation with ATs were realized by removing blockades near the refuge. In ATs, the collaborative rescue of civilians was implemented by expanding the priority search range over time [2]. However, as mentioned above, implementations on various scales were necessary to improve scores.

This year, we intensively developed the Agent and made improvements on various scales. Firstly, PFs were improved the action module, which is the most microscopic part. For making efficient and stabilizing behavior of PFs, the efficiency of blockade removal was improved by changing the conditions for clearing. We also added a process to modify the PFs' behavior by interrupting inefficient action. Secondly, ATs and FBs were improved on the scale of evaluation criteria. They were adjusted to take into account the priorities of civilians. That improved selection of targets and processing of unloading civilians. Lastly, for a macroscopic scale, *dynamic task management system* was implemented. This system separated the target determination mechanism and the candidate evaluation mechanism. It also made agents retain calculation results from multiple selection criteria.

Section 2 explains these detailed theories and implementations. In section 3, the results of the experiment are shown. There are no major changes to Modules this year.

---

\* Corresponding author.

## 2 Strategies

### 2.1 Police Force

**2.1.1 Purpose** The main role of PFs is to remove the blockades generated on the map due to the disaster. It is particularly essential to secure a route for other agents such as ATs and FBs to go through. Until 2019, PFs determined their target with the Road Detector and removed blockades. However, this method caused the situation that PFs regarded blockades which were unreachable as their targets of Action modules.

**2.1.2 Proposed Approach — Efficient clear** PFs could not remove the blockades efficiently, if the decision to remove blockades was made at the largest distance at which it can be removed. This caused the PFs not to remove blockades as efficiently as their primary capacity of removing.

As shown in Figure1, removing blockades was made efficient by removing blockades when they were within not only the removable area but also the certain range.

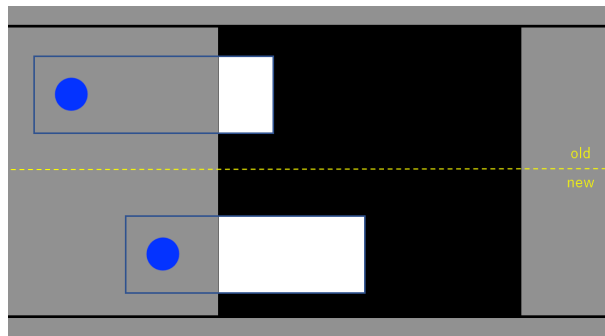


Fig. 1: Comparison of the amount of blockades that can be removed

**2.1.3 Proposed Approach — Interrupt processing** The detail of the process are as follows.

1. When a target cannot be moved due to blockades, an interrupt process ignores the specified target and defines a new target. The activity diagram of the interrupt process is shown in the figure 2.
2. PF keeps track of its coordinates and calculates the distance it has traveled in each cycle.
3. If there is no movement, PF assumes that it is stuck in the blockade and removes the blockade at the shortest distance.

In addition, by comparing the coordinates with those of two cycles ago, it detected the problem that the targets of PFs changed whenever they went back and forth in the same place. In this case, interrupt processing was performed in the same way.

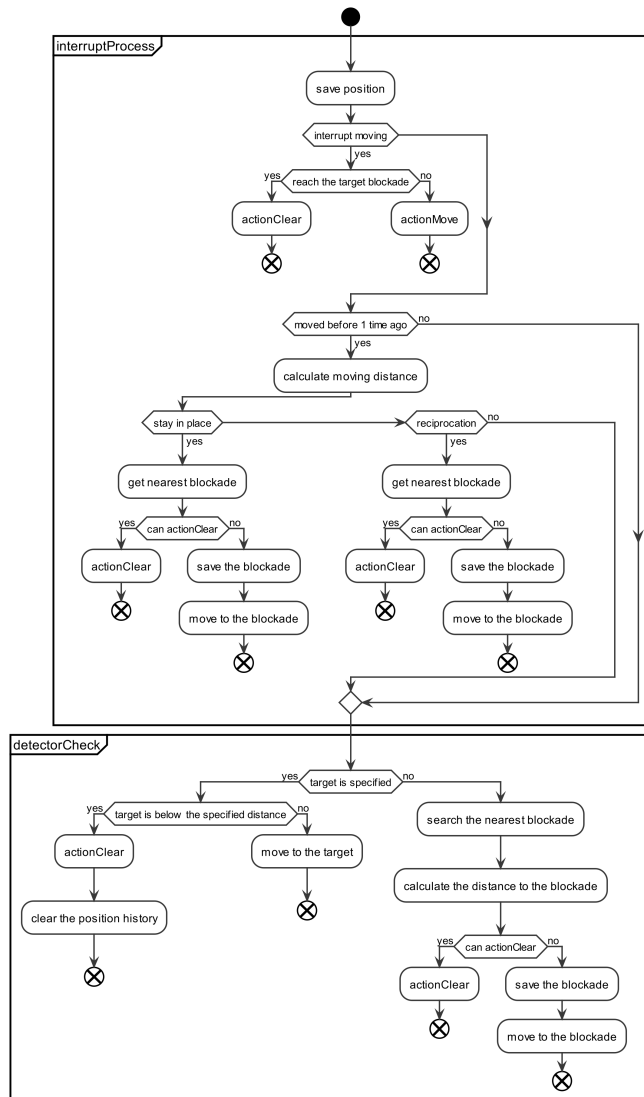


Fig. 2: Interrupt processing

**2.1.4 Pros and Cons** This system made not only PFs but also other agents more active all the time due to efficiently removing blockades.

**2.2 Ambulance Team and Fire Brigade**

**2.2.1 Purpose** ATs and FBs can rescue only a part of civilians. Therefore, they have to prioritize civilians based on some criteria. In rescue activities at actual disaster scenes, the process of determining the priority of civilians treatments is introduced and it is called *triage* [1]. The priority of the civilians is determined based on their states. The aim is to save as many civilians as possible with limited medical resources. This year, triage was adapted to RCRS: *triage system*.

*triage system* is a method that divides civilians into several groups based on their damages. With reference to this RCRS, *triage system* classified civilians into five groups as the following procedure.

Moreover, until last year, ATs had frequently stuck due to blockades on the paths to refuge. Therefore, we implemented new system for ATs to unloaded the civilians on the road and rescue other ones if the paths to refuges were blocked (hereinafter referred to as *release on refuge system*).

### 2.2.2 Proposed Approach — Triage

1. The time until civilian  $i$  die (hereinafter referred to as *life period*) is defined. The *life period*  $P_i(t)$  of civilian  $i$  at the time  $t$  is represented:

$$P_i(t) = \frac{D_i(t)}{L_i(t)} (D_i(t) > 0),$$

where  $L_i(t)$  represents HP of civilian  $i$  at the time  $t$ ;  $D_i(t)$  represents the damage of civilian  $i$  at the time  $t$ , and each agent gets  $D_i(t)$  in each cycle.

2. All civilians were classified into five groups depending on their life periods as shown in Figure 3.

The Life Period	The Priority
0~5	Not Rescued(Fifth)
5~50	First
50~100	Second
100~150	Third
150~	Forth

Fig. 3

3. In addition, all civilians which are not damaged were added to the *do not rescue group*.

**2.2.3 Proposed Approach —release on refuge** Unloaded civilians were classified *do not rescue group* of *triage system* (as mentioned above), and this enabled ATs to avoid loading and unloading the same civilian repeatedly. After PFs removed the blockades on the road that civilians were unloaded, the priority of the civilians are recalculated and finally ATs will carry these civilians to refuges.

**2.2.4 Pros and Cons** In *triage system*, mass rescue activities were carried out more efficiently by introducing a clear classification of civilians. For example, the number of ATs and FBs to rescue civilians was determined by referring to the priority of the civilians that determined in section 2.2.2. This helps ATs and FBs to rescue as many civilians as possible with the limited medical resources. *Release on refuge system* enabled ATs to avoid doing waste tasks such as going to carry dead civilians.

However, there are some issues. *Triage system* considered information for only civilians. In actual rescue activities, other information such as the number of free ATs and FBs which are around damaged civilians is also necessary. It can be expected that ATs and FBs will cause rescue

more civilians if they consider these factor. In the *release on refuge system*, ATs determined whether they unloaded civilians depending on only blockades on the shortest paths to refuges. This caused ATs to unload civilians even though there were other paths which were not blocked. In addition, the numbers of free beds in refuges were not considered. This caused the situations that some civilians which carried to refuges could not use beds. Therefore, we should implement better systems to make rescue activities more efficient. Therefore, it is required to solve these problems in order to make rescue activities more efficient.

## 2.3 Dynamic Task Management

**2.3.1 Purpose** We created a system called dynamic task management to develop flexible and lean Detector modules. This mechanism separated evaluation and target determination for flexible evaluation. The introduction of *dynamic task management system* greatly improved performance.

Prior to 2020, Detectors in Ri-one evaluated candidates of targets along with the decision-making process. However, this implementation had the following problems.

- Because of the sequential evaluation, target determination depended on a single strategy
- All data related to not selected target candidates lost because the purpose was to determine the target
- Detector codes was bloated and difficult to maintain and update

To solve these problems, we introduced a method of dynamic task management.

**2.3.2 Proposed Approach — Algorithm reorganization** In the old algorithm, the judgment was made for each evaluation criteria. Specifically, the purpose was only to find the most suitable candidate from the candidates by division into case in the calc function. On the other hand, in *dynamic task management system*, evaluation is carried out for each candidate. In addition, the target determination mechanism and the candidate evaluation mechanism are executed separately.

In *dynamic task management system*, flexible evaluation is realized by the cooperation of the following three elements.

- **Agent**
- **Evaluator**, which corresponds to an evaluation criterion
- **TaskManager**, which controls evaluation criteria and manages target candidates

The algorithm follows the procedure below. At the start of the simulation, pass the IDs of all potential targets to *TaskManager*. Next, each of candidates given two indexes called *priority* indicating the suitability as a candidate and *unstablensness* indicating the degree of necessity of re-evaluation. Then, *re-evaluate* and *determine the target* are performed for each cycle. In the re-evaluation, the candidate with the highest degree of instability is extracted. In that time, for the selected candidate, *priority* and *instability* are recalculated. To reduce the amount of process, calculation is performed only for one candidate. Likewise, in determining the target, the candidate with the highest priority is taken out and passed from *TaskManager* to the agent.

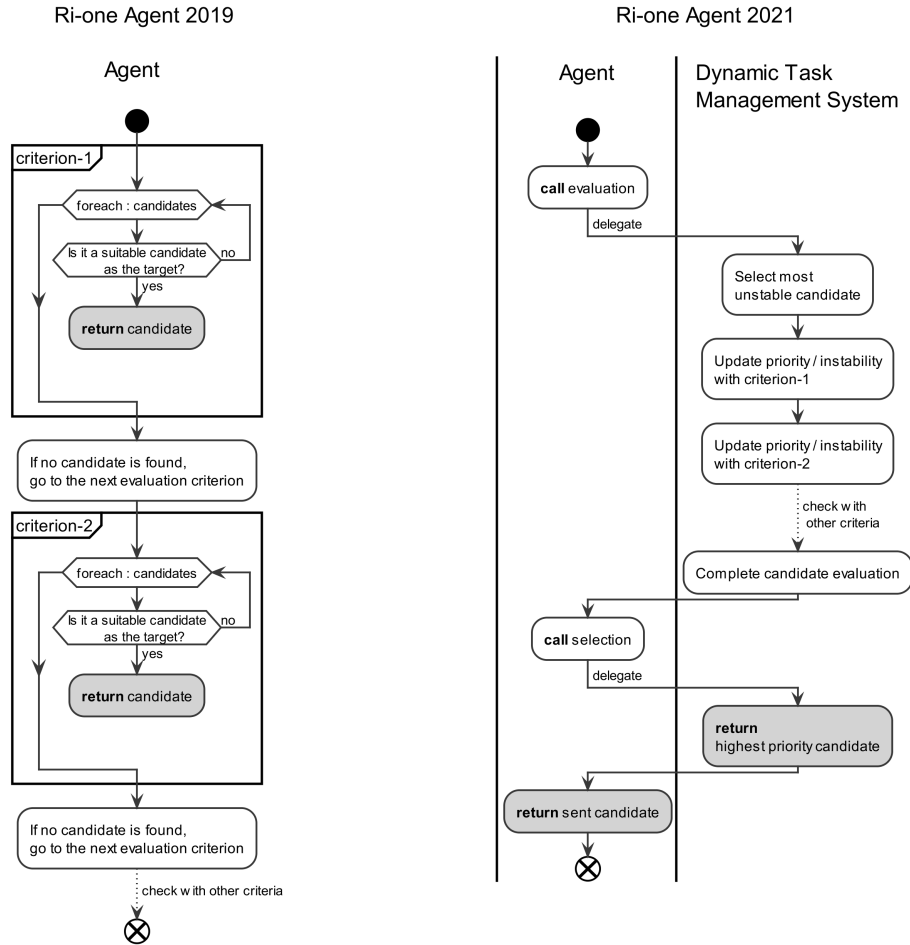


Fig. 4: Algorithm comparison between 2019 and 2021

**2.3.3 Proposed Approach — Ensuring modularity by Delegating** Each agent creates *TaskManagers* as private fields, which are contact points for evaluation. Then, *TaskManagers* manage *Evaluators* as private fields too, which are the evaluation criteria. During the cycle, each *Evaluator* is used to calculate the priority and instability, and the *TaskManagers* assigns target for the Agent as needed.

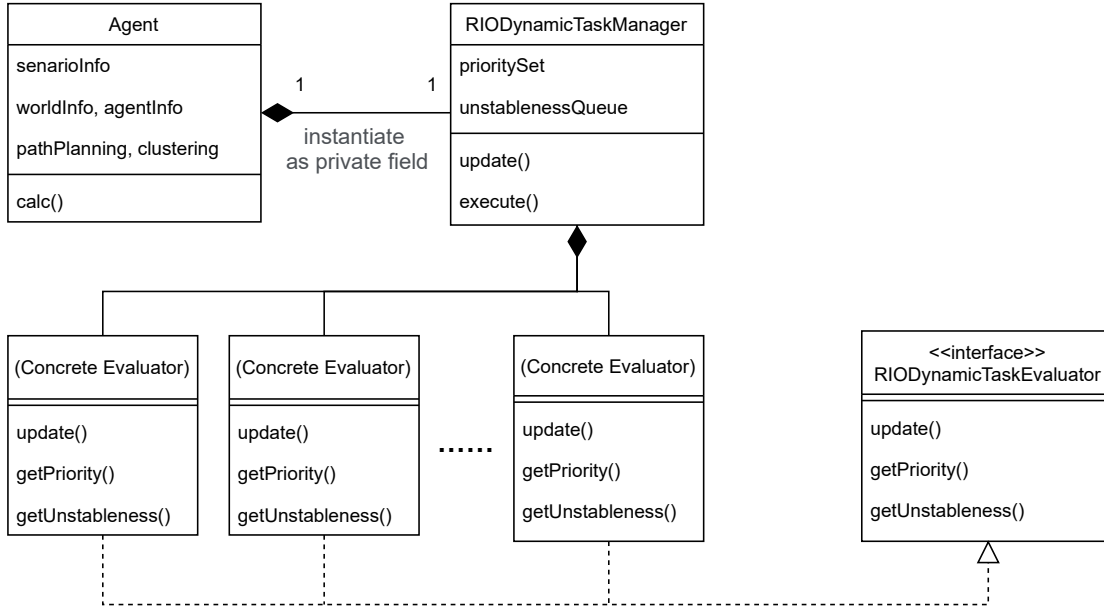


Fig. 5: Dynamic task management class diagram

**2.3.4 Pros and Cons** Experiment confirmed a significant difference. It is seemed because of two reasons. Firstly, calculation result was effectively used without being discarded. In addition, there is no possibility of returning a meaningless value.

In the old method, the process was performed only to determine an optimum target at that time. Therefore, the result of the calculation in calc was discarded. It means the time required for the calculation was wasted. On the other hand, in *dynamic task management system*, calculation results are retained and used for target determination continuously. The score increased by breaking away from the dependence on a single evaluation criterion and making effective use of computer resources.

In addition, in the old method, if an appropriate target could not be found, there was a risk that the process would end without the target decision. On the other hand, in *dynamic task management system*, the target determination and evaluation mechanisms is separated. It enables to a valid value for an target. The elimination of potential bugs is also one of the reasons for the increase in score.

On the other hand, this method has some limitations. In the current implementation, dynamic task management is realized by equally assigning priority / instability between 0 and 1 to each evaluation criteria. From the above experiment, it is shown that this implementation is also effective. However, it is considered that the target search can be performed more effectively by adjusting weights in depends on the evaluation criteria. Moreover, for target selection, in order to alleviate duplication and bias between agents, we are considering a method of probabilistic selection using priority.

### 3 Preliminary Results

In this section, we describe the results of the experiments. These experiments aimed to confirm the effectiveness of implemented functions.

### 3.1 Experiment Conditions

In the experiments, Ri-one 2019 and Ri-one 2021 were compared. Avoiding the influence of the simulation system’s specification changes, the experiments were conducted on the simulation system in 2019. Ri-one 2019 refers to the implementation of 2019. Ri-one 2021 refers to the modified version of Ri-one 2019. It was implemented the functions described in the section 2. The simulations were carried out 20 times for each condition. The specifications of the used computer followed the table below.

Table 1: specs of a computer

OS	Ubuntu 18.04
CPU	Intel Core-i9 10850K
Memory	DDR4-2666 32GB

### 3.2 Score and Analyze

The experiment results followed the table below. The *p test* in the table indicates whether or not a significant improvement appeared in the p tests. The p tests were two-sided test with a significance level of 5%. “P(Positive)” means that significant improvement appeared. “N(Negative)” means that significant deterioration appeared.

Table 2: Results of experiments

Map	score		p test
	Ri-one 2019	Ri-one 2021	
berlin	72.50989287	71.5285817	N
eindhoven	189.7062651	189.3507651	
joao	70.11983339	70.48484166	P
kobe	176.3522562	177.8400341	P
montreal	36.21247646	36.22393002	
paris	84.88469485	83.23055563	
sakae	29.28488181	56.2559197	P
sf	89.80645641	96.79971609	P
vc	260.9711487	251.8328633	N

These results showed the effectiveness of improvements under the rules in 2019. The improvements of this year solved the problems common to the 2021 version of the rule. Therefore, the implementations seemed to be valid even under the new rules.

## 4 Conclusions

We improved the operations of PFs, introduced flexible evaluation criteria for FBs and ATs, and developed *dynamic task management system* for each type of agents. In particular, *dynamic task management system* improved target determination with controlling the evaluation criteria. The performance improvement was as shown in the section 3.



However, these improvements are only the basis for breaking away from instruction-based implementations and sequential processing. We are going to consider advanced usages of existing evaluation criteria, and promote improvement of *dynamic task management*. Those developments will lead to further performance improvement.

## References

1. Christopher A. Kahn, Carl H. Schultz, K.T.M.C.L.A.: Does start triage work? an outcomes assessment after a disaster. In: Annals of Emergency Medicine Volume 54, Issue 3, Pages 424-430. American College of Emergency Physicians (2009)
2. Higashi, Y., Ijichi, T., Joichiro Amada Soichiro Komura, Akinori Kanechika, H.N.K.N.K.F.H.N.R.F.K.S.T.N.: Robocuprescue 2019-rescue simulation league. team description ri-one (japan) (2019)