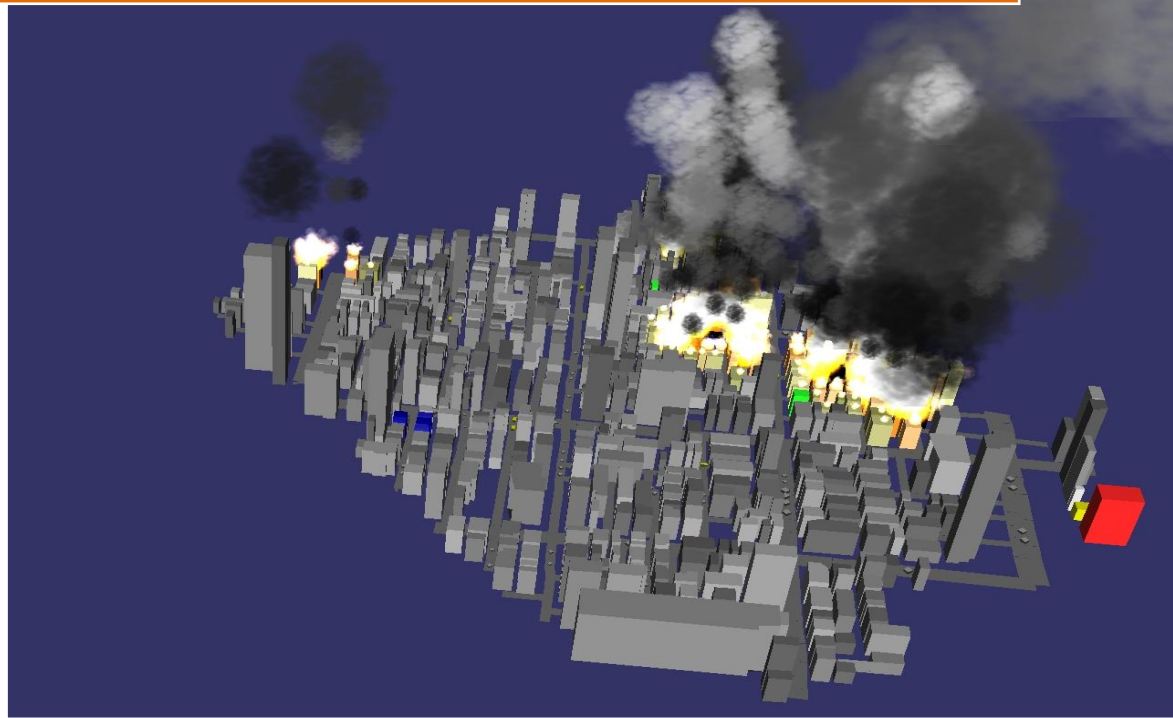


RoboCup 2014

# RoboCup Rescue Simulation League Agent Competition



## Rules and Setup

January 2014

### **Authors:**

Farshid Faraji  
Luis Gustavo Nardin  
Ali Modaresi  
Nobuhiro Ito  
Moritz Goebelbecker  
Salim Malakouti

## Table of Contents

Abstract	3
1. Introduction	3
2. Hardware	4
2.1. Cluster Configuration	4
2.2. Each PC Specification	4
2.3. Operating System	4
3. Simulation Package	4
4. Scenarios	4
5. Maps	4
6. Ranking	5
7. Rules	5
8. Parameters	8
9. Acknowledgments	12
10. References	12

## Abstract

This document describes the rules and the ranking formula for the RoboCup Rescue Agent Simulation 2014 competition as well as the used RoboCup Rescue simulator features and parameters.

## 1. Introduction

The purpose of the RoboCup Rescue Agents Simulation is twofold. First, it aims to develop simulators that form the infrastructure of the simulation system and emulate realistic phenomena predominant in disasters. Second, it aims to develop intelligent agents that are given the capabilities of the main actors in a disaster response scenario. The RoboCup Rescue simulation platform currently runs a Kernel which connects the following main simulators:

- **Traffic simulator** – enforces rules of traffic (e.g. one-way, multi-lane roads) and simulates traffic jams and the movements of agents along roads on a map (given speed and direction).
- **Fire Simulator** – simulates the processes involved in the propagation of fire according to the properties of buildings, wind direction, and water content.
- **Civilian Simulator** – simulates the behavior of civilians such as movement along roads.
- **Smoke Simulator** – simulates the behavior of smoke in the environment. Its intention is to allow the identification of possible fires not directly visible to the agents with the normal line of sight perception. Smoke is modeled as entities (called "Smoke Entities") in the environment and is created by the Smoke simulator.
- **Collapse Simulator** – simulates the generation of new blockades (aftershock blockages) during the simulation. Each scenario is limited to have a maximum of 10 aftershock blockages. And the mean of the severity of all aftershocks should not exceed 30% of the initial earthquake collapse.

In complement to these simulators, several features are available:

- **Clear Method** - The agents are able to clear blockades in preferred directions, sides and size. Figure 1 shows the new clearing method.

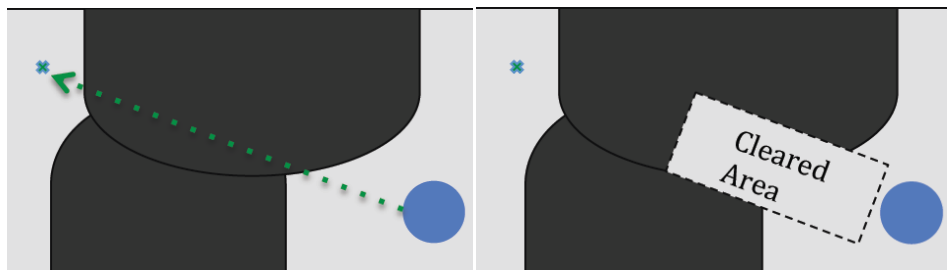


Figure 1: New clearing method

- **Gas Stations** - A Gas Station is an entity in the environment. When catches fire, it has two major consequences different from a normal building:
  - Increase the fire expansion in vicinity buildings
  - Destruct the nearby buildings

- **Hydrant** - Hydrant is an entity that allows the fire brigades to fill their water tanks instead of requiring them to go to the refuge. The tank fill rate of a Hydrant is lower than that of a refuge and only one fire brigade can use it in each cycle.



## 2. Hardware

### 2.1. Cluster Configuration

In the competition venue, there will be several clusters with 4 computers each. One PC per cluster will be reserved for the simulator components; the remaining three will be available to run the agent teams.

### 2.2. Each PC Specification

CPU: Core i7 processor 3.0GHz or higher  
RAM: 8.0 GB or higher

### 2.3. Operating System

Linux Ubuntu 12.04 LTS (64bit) or higher with Oracle Java 1.7 run-time

## 3. Simulation Package

The simulation package used in the RoboCup Rescue Agent Simulation 2014 competition will be available at <http://www.sourceforge.net/projects/roborescue>.

## 4. Scenarios

A scenario is composed of (i) a map, (ii) a set of initial rescue agent/civilian positions, and (iii) a set of configuration options for each of the simulator components. The Technical Committee will produce a set of scenarios for the competition. All teams will run on the same set of scenarios.

## 5. Maps

The maps will be provided by the Technical Committee.

The number of buildings and roads will be limited to 10.000 of each.

A validation tool will be used to check the full connectivity of roads and building entrances in the map. However, teams do not have the right to complain in the case in which roads or building entrances are not fully connected if evidenced that this results from a validation tool error.

**Map Scale:** Simulation parameters will be adjusted according to the maps scale, intended difficult level for the scenario, and to match the real world.

## 6. Ranking

The 2014 competition will be composed of several sessions ( $S$ ), and at each session the participating teams will receive an identification number from  $t_1$  to  $t_n$ , where  $n$  represents the number teams participating on that session. It session will have a set of maps ( $M$ ), and each map will also receive an identification  $m_1$  to  $m_p$ , where  $p$  represents the number of maps in that session. Then, the teams ( $T = \{t_1, \dots, t_n\}$ ) will be evaluated at each session for each map ( $M = \{m_1, \dots, m_p\}$ ) as described below.

In a session  $s \in S$ , for each map  $j$  ( $j \in M$ ), the points of each team  $i$  ( $i \in T$ ), will be calculated based on the following formula

$$Team_i Map_j Points = \frac{Team_i Map_j Score - Min(Teams' Map_j score)}{Max(Teams' Map_j score) - Min(Teams' Map_j score)}$$

In order to reduce the impact of incomparable maps (i.e., while scores are too close to compare), a map value is computed for each map according to the variance of the scores of all the teams and the initial score of the map.

$$Map_j Value = \frac{Teams' Map_j Scores Variance}{Map_j Initial Score}$$

The Map value and the Team Points are then aggregated as described in the formula below.

$$Team_i Final Value Session_s = \sum_{j=m_1}^{m_p} Map_j Value \times Map_j Team's Points$$

The Team Final value calculated for in a session is then used to generate a ranking for that session in which the team with the highest Team Final Value is ranked as first, the second highest as second, and so on.

**OBSERVATION:** In the case the maximum and minimum values are the same, the  $Team_i$  Final Value is 0. It does not impact the final ranking as the map, in which this exceptional condition occurs, does not create any difference among the teams.

## 7. Rules

- a. **Rounds:** The competition is structured into 3 rounds according to the following order: one preliminary round, one semifinal, and one final round. The preliminary round will be run in two consecutive days (first and second days of competition), while the semifinal and final rounds are executed in one day each (third and fourth day of competition).
- b. **Sessions:** Each round consists of several sessions. A session is the set of simulation runs for all teams in a specific scenario. A member of the Technical or Organizing Committee will be responsible for chairing each session. This session chair is responsible for starting the session, collecting scores and logs, and handling any issues that arise during the session. The teams do not have the right to object to the scenarios provided by the Technical Committee.

- c. **Code submission:** All teams should submit working source codes (no binaries will be accepted) and adequate compile scripts at the start of each round. The number of submissions as well as their time and requirements will be explained further in a brief presentation during the competition setup time. The Technical Committee has authority to do adjustments in the time of submissions during the competition. Furthermore, the Technical Committee has authority to review all submitted teams' source code.
- d. **Scenarios:** The scenarios will be provided by the Technical Committee. Teams shall NOT know the disaster situation (map, random seeds, simulator configuration, parameter values) of the session before it starts. All the conditions for a particular disaster situation shall be identical for all the teams.
- e. **Agents:** Teams shall implement all kinds of agents and must connect the correct number of agents as specified for the current session.
- f. **Shared memory:** Agents **must not** use any form of shared memory, including static memory accessible to all agents, direct function calls between agents, or writing files for use by other agents during the scenario simulation; however, agents are allowed to write files during the *Pre-Computation phase* as described in item h. The Technical Committee may require each agent to run in a different virtual/physical machine if teams are suspected of violating this rule.
- g. **Phases:** There will be two phases of execution for each scenario simulation requiring two executions of the team's code. The two phases are: (i) *Pre-Computation phase* (item h) and (ii) *Simulation phase* (item i).
- h. **Pre-Computation phase:** The Pre-Computation allows an agent of each type to load and use map-specific data in order to pre-process them and store it in a file. In this phase, the connected agents are permitted to write into files. However, only one agent of each type can connect to the server and executes such Pre-Computation algorithms. This phase is limited in 2 minutes and after the time is elapsed, the server will be terminated. Pre-Computation is allowed under the following conditions:
  - I. The data must be generated by a computer program with no human interaction.
  - II. Information for all known maps must be generated by a single computer program.
  - III. The computer program used for computing data for known maps should work properly if it is given a new map.
  - IV. An agent should choose the data file to be used itself.
  - V. Agents should be able to work if no Pre-Computation data is present for a map.
  - VI. The source of the Pre-Computation program shall be open after the competition.
- i. **Simulation phase:** The Simulation phase corresponds to the actual team's simulation in the competition scenario (the one valid for collecting the score). The team must connect all its agents to the kernel in order to perform the actual scenario simulation in 3 minutes. The scenario simulation will begin no later than 3 minutes after the first agent begins its handshake with the kernel. All file permissions, except read permission for previously written files, will be removed.

j. **Valid games:** In some cases a simulation may fail. There are two general cases:

- (a) A simulator or the kernel crashes.
- (b) Some agents stop or crash.

In the first case, the score for that run shall be zero. In the second, case the final score is calculated as usual.

Teams will not be entitled to a rerun in most circumstances. It is expected that teams write their agents so that they work correctly with the given simulators. In extreme circumstances teams may request a rerun. Examples of circumstances that may result in a rerun are:

- I. A power failure.
- II. Accidental or deliberate termination of a kernel, simulator or agent process.

Examples of events that will not result in a rerun are:

- I. A simulator crash.
- II. Agents failing to fully connect before the simulation starts.
- III. Agents crashing or failing to act during the run.
- IV. Observing apparently incorrect behavior by a simulator or the viewer.

Teams that wish to request a rerun must do so in writing. The request must include the team name, the scenario name, a description of the problem and the reasons why the team feels a rerun is appropriate. The request must also state whether the request is for a rerun of just that team or for a full rerun of the session.

k. **Simulation bugs:** It is the responsibility of teams to ensure that their code works correctly with the provided simulators. Although the Technical Committee makes every attempt to provide a reliable simulation environment, the Technical Committee accepts no responsibility for any kind of software failure at competition time. Simulator bugs are not sufficient grounds to request a rerun.

l. **Committee decisions:** If a problem arises during a session then teams may ask for the session chair to resolve the problem. The session chair may make a decision on the spot, or may refer it to the committee. Decisions are final, but if a team strongly disagrees then they may submit a written appeal to the committee. In order to allow the competition to continue, appeals will not be heard during a round but will be discussed by the committee at the end of each day. The Technical Committee will make final decision at any condition.

m. **Comments from teams:** Occasionally, members of participating teams may wish to voice concerns or make suggestions to the Technical Committee about the running of the competition. If these comments or suggestions are deemed derogatory or abusive then the matter will be referred to the RoboCup Trustees and may result in penalties for the team concerned. Penalties may include points reduction or, in the worst case, disqualification.

n. **Exploits:** A team that knowingly uses bugs in the simulation package to gain an advantage will be disqualified from the competition. Disqualifications will be made only after consultation with the RoboCup Trustees.

o. **Open source policy:**

- Source code files must be open immediately after the end of the competition to guarantee fair play and to encourage community activity after competition.
- Log files and related parameter files will be open.

## 8. Parameters

The following tables describe the set of parameters that can change during the competition. Note that agents will not be given all the available parameters – see Table 15 for the set of parameters that agents are guaranteed to be able to query.

Entity	Min	Max
Fire brigade	0	50
Police force	0	50
Ambulance	0	50
Fire station	0	5
Police office	0	5
Ambulance centre	0	5
Civilians	0	1000
Refuges	0	Unlimited
Initial ignition points	0	50

Table 1: Allowed number of agents, refuges and ignition points in a scenario.

The simulation parameters can be varied between scenarios. The following tables list all variable parameters and their possible ranges.

Parameter	Description	Range
random.seed	Seed for random number generator	Anything

Table 2: Allowed ranges for simulation parameters common to all components.

Parameter	Description	Range
kernel.timesteps	The number of timesteps in the simulation	100 – 1000
kernel.agents.think-time	The number of milliseconds each agent has to send commands	500 – 3000

Table 3: Allowed ranges for kernel parameters.

Parameter: comms.channels.*	Description	Range
count	Number of communication channels	1 – 20
max.platoon	The number of channels a platoon agent can subscribe to	0 – 10
max.centre	The number of channels a centre agent can subscribe to	0 – 20

Table 4: Allowed range for general communication channels configuration parameters.

Parameter comms.channels.<x>.* <x> is the channel number	Description	Range
type	The type of channel	Voice
range	The maximum range of a message in Mm	0 – 300000
messages.size	The maximum size of a voice message in bytes	64 – 2048
messages.max	The maximum number of a voice message an agent can send in one timestep	1 – 100



Table 5: Allowed range for voice channel parameters.

Parameter comms.channels.<x>.* <x> is the channel number	Description	Range
Type	The type of channel	radio
Bandwidth	The maximum capacity of the channel in bytes per timestep	0 – 8192

Table 6: Allowed range for radio channel parameters.

Parameter comms.channels.<x>.* <x> is the channel number	Description	Range
noise.[input   output].failure.use	Whether to use failure noise on channel x in input or output mode.	yes or no
noise.[input   output].failure.p	The probability of message failure.	0 – 1
noise.[input   output].dropout.use	Whether to use dropout noise on channel x in input or output mode.	yes or no
noise.[input   output].dropout.p	The probability of message dropout.	0 – 1

Table 7: Allowed range for voice and radio communication parameters.<sup>1</sup>

Parameter	Description	Range
perception.los.max-distance	Maximum distance the agent can perceive world changes.	30000 – 150000

Table 8: Allowed ranges for perception parameters.

Parameter	Description	Range
collapse.wood.*	Proportion of wooden buildings with each degree of damage	0 – 1 Sum = 1
collapse.steel.*	Proportion of steel buildings with each degree of damage	0 – 1 Sum = 1
collapse.concrete.*	Proportion of concrete buildings with each degree of damage	0 – 1 Sum = 1

Table 9: Allowed ranges for collapse simulator parameters.

Parameter	Description	Range
fire.tank.maximum	Water tank capacity	5000 – 50000
fire.tank.refill-rate	Water tank refill rate	500 – 15000
fire.extinguish.max-power	Maximum amount of water that can be used per timestep	500 – 150000
fire.extinguish.max-distance	Maximal extinguish distance	10000 – 150000

Table 10: Allowed ranges for fire simulator parameters.

<sup>1</sup> Noise can be specified as input or output (or both). Input noise is applied as the agent sends a message to the server; output noise is applied as an agent receives a message. Thus, input noise is identical for all receivers but output noise is unique to each receiver. There are two types of noise: failure noise and dropout noise. Failure noise means a message disappears completely with no notification to either the sender or the receiver. Dropout noise removes the content of a message but the receiver still receives a zero-length communication from the sender, i.e. the sender knows a message was sent but the content is lost.

Parameter: misc.* <type> can be wood, steel or concrete	Description	Range
<b>buriedness.&lt;type&gt;.severity.rate</b>	The probability that an agent in a collapsed building of type <type> with a degree of collapse severity will be buried	0 – 1
<b>buriedness.&lt;type&gt;.severity.value</b>	The initial buriedness value for a buried agent in a collapsed building of type <type> with a degree of collapse severity	0 – 200
<b>injury.collapse.&lt;type&gt;.severity.slight</b>	The probability that an agent inside a collapsing building of type <type> with a degree of collapse severity will receive a slight injury	0 – 1
<b>injury.collapse.&lt;type&gt;.severity.serious</b>	The probability that an agent inside a collapsing building of type <type> with a degree of collapse severity will receive a serious injury	0 – 1
<b>injury.collapse.&lt;type&gt;.severity.critical</b>	The probability that an agent inside a collapsing building of type <type> with a degree of collapse severity will receive a critical injury	0 – 1
<b>injury.collapse.slight</b>	The amount of damage that a slight injury due to collapse causes	0 – 10000
<b>injury.collapse.serious</b>	The amount of damage that a serious injury due to collapse cause	0 – 10000
<b>injury.collapse.critical</b>	The amount of damage that a critical injury due to collapse causes	0 – 10000
<b>injury.collapse.multiplier.&lt;type&gt;</b>	The damage multiplier for an agent of type <type> due to collapse	0 – 1
<b>injury.bury.&lt;type&gt;.severity.slight</b>	The probability that an agent buried inside a building of <type> with a degree of collapse severity will receive a slight injury	0 – 1
<b>injury.bury.&lt;type&gt;.severity.serious</b>	The probability that an agent buried inside a building of <type> with a degree of collapse severity will receive a serious injury	0 – 1
<b>injury.bury.&lt;type&gt;.severity.critical</b>	The probability that an agent buried inside a building of <type> with a degree of collapse severity will receive a critical injury	0 – 1
<b>injury.bury.slight</b>	The amount of damage that a slight injury due to buriedness causes	0 – 10000
<b>injury.bury.serious</b>	The amount of damage that a serious injury due to buriedness causes	0 – 10000

<b>injury.bury.critical</b>	The amount of damage that a critical injury due to buriedness causes	0 – 10000
<b>injury.bury.multiplier.&lt;type&gt;</b>	The damage multiplier for an agent of <type> due to buriedness	0 – 1
<b>injury.fire.&lt;type&gt;.severity.slight</b>	The probability that an agent inside a burning building of <type> with a degree of collapse severity will receive a slight injury	0 – 1
<b>injury.fire.&lt;type&gt;.severity.serious</b>	The probability that an agent inside a burning building of <type> with a degree of collapse severity will receive a serious injury	0 – 1
<b>injury.fire.&lt;type&gt;.severity.critical</b>	The probability that an agent inside a burning building of <type> with a degree of collapse severity will receive a critical injury	0 – 1
<b>injury.fire.slight</b>	The amount of damage that a slight injury due to fire causes	0 – 10000
<b>injury.fire.serious</b>	The amount of damage that a serious injury due to fire causes	0 – 10000
<b>injury.fire.critical</b>	The amount of damage that a critical injury due to fire causes	0 – 10000
<b>injury.fire.multiplier.&lt;type&gt;</b>	The damage multiplier for an agent of type <type> due to fire	0 – 1
<b>injury.&lt;type&gt;.k</b>	The k parameter for the damage progression function for injury type <type> (collapse, bury or fire)	0 – 1
<b>injury.&lt;type&gt;.l</b>	The l parameter for the damage progression function for injury type <type> (collapse, bury or fire)	0 – 1
<b>injury.&lt;type&gt;.noise.mean</b>	The mean noise added to the damage progression function for injury type <type> (collapse, bury or fire)	0 – 1
<b>injury.&lt;type&gt;.noise.sd</b>	The standard deviation of noise added to the damage progression function for injury type <type> (collapse, bury or fire)	0 – 1

Table 11: Allowed ranges for misc simulator parameters.

Parameter	Description	Range
<b>clear.repair.rate</b>	The rate of road clearing per police force agent in square m per timestep	0 – 50000

Table 12: Allowed ranges for clear simulator parameters.

Parameter	Description	Range
<b>ignition.random.lambda</b>	The parameter of the Poission distribution used to determine frequency of building ignition	0 – 1

Table 13: Allowed ranges for ignition simulator parameters.

Parameter	Description	Range
-----------	-------------	-------

Table 14: Allowed ranges for traffic simulator parameters.

Parameter	Description
<b>kernel.agents.think-time</b>	See Table 3
<b>kernel.startup.connect-time</b>	See Section 8 – Item f.
<b>comms.channels.count</b>	See Table 4
<b>comms.channels.&lt;x&gt;.type</b>	See Table 5 and Table 6
<b>comms.channels.&lt;x&gt;.range</b>	See Table 5
<b>comms.channels.&lt;x&gt;.messages.size</b>	
<b>comms.channels.&lt;x&gt;.messages.max</b>	
<b>comms.channels.&lt;x&gt;.bandwidth</b>	See Table 6
<b>fire.tank.maximum</b>	See Table 10
<b>fire.tank.refill-rate</b>	
<b>fire.extinguish.max-power</b>	
<b>fire.extinguish.max-distance</b>	
<b>clear.repair.rate</b>	See Table 12
<b>scenario.agents.fb</b>	The number of Fire Brigades in the scenario
<b>scenario.agents.fs</b>	The number of Fire Stations in the scenario
<b>scenario.agents.pf</b>	The number of Police Forces in the scenario
<b>scenario.agents.po</b>	The number of Police Offices in the scenario
<b>Scenario.agents.at</b>	The number of Ambulance Teams in the scenario
<b>scenario.agents.ac</b>	The number of ambulance centres in the scenario
<b>kernel.communication-model</b>	The communication model class name in use
<b>kernel.perception</b>	The class name of the perception model in use

Table 15: Parameters agents are guaranteed to be able to query.

## 9. Acknowledgments

Special thanks to Cameron Skinner, who were author of the rules of the previous years.

## 10. References

[1] Cameron Skinner, RoboCup Rescue Simulation League Agent Competition 2010 Rules and Setup