

# RoboCup Rescue Simulation League Agent 2016 Competition Rules and Setup

Farshid Faraji, Luis G. Nardin, Ali Modaresi, Dina Helal and Nobuhiro Ito

## Abstract

This document describes the rules and the ranking formula for the 2016 RoboCup Rescue Agent Simulation League Competitions.

## 1 Introduction

Natural disasters are major adverse events that cause large-scale economic, human, and environmental losses. They are usually difficult to predict and even more challenging to prevent from happening. These characteristics demand disaster management strategies to be in place to reduce the negative impact of the disaster. Disaster management is usually divided in four phases:

- *Mitigation* includes the identification of hazards, the assessment of threats, and the implementation of measures to reduce potential losses and damages. Examples are evacuation and environment assessment.
- *Preparedness* includes the elaboration of plans to deal with disasters and their consequences. This phase aims to have equipment and procedures ready for use when a disaster occurs.
- *Response* includes the search and rescue actions executed during and just after a disaster.
- *Recovery* takes place after the occurrence of a disaster and involves the assistance of affected people and the restoration of the basic services.

Efficacious response plans are essential for reducing the negative impacts of natural disasters. These plans are usually elaborated by policy-makers based on previous experiences and with limited evaluation of efficacy. Assess response plans, however, is a challenging task as one cannot reproduce the real conditions of a natural disaster and its effects in a real environment just

for assessment purposes. In this case, simulation plays a key role in assisting to assess response plans as it enables the reproduction of the real conditions of a disaster in an artificial environment.

The RoboCup Rescue Agent Simulation League mission is to promote research and development in this socially significant domain at various levels. The effective implementation of this mission is translated into three main league's goals. First, the league aims to provide a simulation software infrastructure able to realistically represent natural disaster scenarios where response plans may be assessed. Second, it aims to define evaluation benchmarks for response plans that may be used by policy-makers. Finally, it aims to promote the research and development by conducting competitions to stimulate the exchange of ideas and experience among practitioners, which may help in the development of more sophisticated and formalized plans to efficaciously respond to natural disasters.

The RoboCup Rescue Agent Simulation competition held during the RoboCup is focused mostly on the third objective. The competition is subdivided in the

- RoboCup Rescue Simulation Agent Competition (Section 2)
- Technical Challenge Competition (Section 3)
- Infrastructure Competition (Section 4)

## **2 RoboCup Rescue Simulation Agent Competition**

The RoboCup Rescue Simulation Agent Competition involves evaluating competing agent coordination algorithms on different scenarios of the RoboCup Rescue Agent simulation platform. Specifically, it involves evaluating coordination algorithms enabling teams of Ambulances, Police Forces, and Fire Brigades to rescue civilians and extinguish fires in cities where an earthquake has just occurred. Each team is evaluated on a set of scenarios that represent different disaster situations (e.g., civilians and fires, major fire in one corner of a city, blocked roads to refuges, damaged platoon agents, fire maps, civilian maps where only fires need to be extinguished or civilians need be saved respectively).

This competition is divided in three rounds (i.e., preliminary, semi-final, and final rounds). At each round, the teams provide their code that will be executed on a set of scenarios and a score calculated to each scenario.

A scenario is composed of a map, a set of initial rescue agent/civilian positions, and a set of configuration options for each of the simulator components. Each map is constrained to a maximum of 10,000 roads and 10,000 buildings. The buildings and roads entrances are supposed to be fully connected. A validation tool will be used to check the full connectivity of roads and building entrances in each map. However, teams do not have the right to complain in case roads or building entrances are not fully connected if evidenced that this was not detected by the validation tool.

In addition to the agent's teams performance on the scenarios, the teams are also evaluated on a presentation of the strategy implemented and the score obtained in this presentation counts as a scenario in all the rounds. At the end of each round, the teams with the highest sum of scores in all the scenarios (including the presentation score) are selected to move to the next round of the competition.

The RoboCup Rescue Agent Simulation Technical Committee is responsible to elaborate a set of scenarios for the competition and all teams code are executed on the same set of scenarios.

## 2.1 Presentation

The presentation aims to share the knowledge of the teams and improve the academic research aspects of the league. Each team will have 20 minutes to present their implementation and another 10 minutes for questions and answers. The presentation will be evaluated by a panel of experts and these evaluations will be incorporated into the score of all the rounds. The presentation ranking will be considered as an additional scenario in the scoring system of each round. The teams' presentation will be evaluated according to a set of criteria:

- *Relevance* (5 points) Evaluates how much relevant is the team's approach to the goals of the RoboCup Rescue. 0 means it is not relevant and 5 means it is very relevant.
- *Originality* (5 points) Evaluates how much original is the proposed team's approach to RoboCup Rescue. 0 means it is not original and 5 means it is very original.
- *Significance* (5 points) Evaluates how much significant to the league is the proposed team's approach. 0 means it is not significant and 5 means it is very significant.
- *Slides Content* (5 points) Evaluates the quality and completeness of the presentation material with respect to the team's strategy and the

readability. 0 means that the presentation slides are of poor quality with respect to the content and 5 means that it is very informative and complete.

- *Slides Structure* (5 points) Evaluates the structure of the presentation material. 0 means that the presentation slides are poorly structured and 5 means that it is very well structured.
- *Talk* (5 points) Evaluates how much clear and easy to follow was the presentation and explanation, and whether the presenter had a positive attitude or not, with respect to the presentation and the raised questions. 0 means it is a not clear presentation and bad attitude of the presenter and 5 means it is a clear presentation and good attitude of the presenter.

Each team's presentation score will be calculated taking into account the sum of points given by the other teams excluding the 3 (three) best and the 3 (three) worst scores, plus the points given by 2 Technical Committee members who are not member of any team participating on the RoboCup Rescue Simulation Agent Competition.

The score of each team will then be included in the ranking calculation in Section 2.3 as a scenario in all rounds it participates.

All teams will have the same number of evaluations and the same evaluators. In case some evaluator is not able to participate in the evaluation of all teams' presentation, his/her evaluation will be disregarded.

## 2.2 Rules

- Remote participation:** Remote participation is not allowed in any circumstance. At least one team member must register and be locally present at the venue.
- Rounds:** The competition is structured into three rounds: one preliminary round, one semifinal, and one final round. The preliminary round will be executed 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 days of competition).
- Sessions:** Each round consists of several sessions. A session is comprised of a set of simulations in different scenarios. A member of the Technical Committee will chair each session. The session chair is responsible for executing the simulations, collecting scores and logs, and handling any issues that arise during the session.

- (d) **Code submission:** All teams must submit the team's source code (binary code will not be accepted) and adequate compile scripts before the start of each round. The number and time of submissions and specific requirements will be explained in the brief presentation at the competition setup time at the venue. The Technical Committee has the authority to change the time of submissions and to review all submitted teams' source code.
- (e) **Scenarios:** The scenarios will be provided by the Technical Committee. Teams shall NOT know the disaster situation (map, random seeds, simulator configuration, parameter values, and phases of execution) of the session before it starts. All the conditions for a particular disaster situation shall be identical for all the teams.
- (f) **Agents:** Teams shall implement all types of agents. It is the responsibility of the team to ensure that its code connects the correct number of agents to the server.
- (g) **Shared memory:** Agents cannot 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. The exception is the Pre-Computation phase when agents are allowed to write files (see item (h) for details). The Technical Committee may execute each agent of the team in a different virtual/physical machine if the team is suspected of violating this rule.
- (h) **Phases:** The scenario simulation may be performed in two phases of execution of the team's code: the Pre-Computation phase (item (i)) and the Simulation phase (item (j)). The Pre-Computation phase, however, is not mandatory for all scenarios and is assumed a configuration parameter of the scenario. Thus, the execution of the Pre-Computation phase will be defined as a configuration parameter of the scenario (see item (e)).
- (i) **Pre-Computation phase:** The Pre-Computation phase allows an agent of each type to pre-process map-specific and scenario-specific data and store them into a file for using during the Simulation phase. Only one agent of each type can connect to the server and execute the Pre-Computation algorithm. This phase is limited to 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 or intervention.
  - II. Data for all maps must be generated by a single computer program.
  - III. The computer program should work for any new map.
  - IV. Agent must choose the file to store the pre-computing data.
  - V. Agents must be able to work if no Pre-Computation data is present for the map.
  - VI. The source of the Pre-Computation program must be released after the competition.
- (j) **Simulation phase:** The Simulation phase corresponds to the actual team's simulation in the competition scenario. The team must connect in at most 3 minutes all agents to the kernel in order to perform the scenario simulation. 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.
- (k) **Valid map:** The Technical Committee is entitled to define whether a map results is valid or invalid in a session. The decision is based on the results of the map and it may be decided that a map is invalid when all the teams score very close in the map.
- (l) **Valid games:** Teams will NOT be entitled to rerun their agent team in most circumstances. It is expected that teams write their agents so that they work correctly. In extreme circumstances teams may have the right of a single rerun. Circumstances that may result in a rerun are:
- I. A power failure.
  - II. Accidental or deliberate termination of a kernel, simulator or agent process.
  - III. Java Virtual machine crash.

In the case of rerunning, the latest score is used as the official score of the team on that map.

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 session rerun. Only one Java Virtual Machine crash rerun request is accepted in each session.

- (m) **Bugs:** It is the responsibility of each team to ensure that its code works correctly with the provided simulators. Although the Technical Committee makes every effort to provide a reliable simulation environment, the Technical Committee has no responsibility for any kind of software failure at competition time. Simulator bugs are not sufficient grounds to request a rerun.
- (n) **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. Chair 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.
- (o) **Comments from teams: Only the team leader** of participating teams can comment and 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. Expect team leaders, other team members are not entitled to comment or make suggestions to the Technical Committee.
- (p) **Bugs exploitation:** A team that knowingly uses bugs in the simulation package to gain advantage will be disqualified from the competition. Disqualifications will be made only after consultation with the RoboCup Trustees.

- (q) **Dispute resolution:** If there is an ambiguity in the rule and any unexpected situation happens, a temporary committee composed of the Technical, Organization and Executive committees members and the local chair have the power to take a decision regarding the issue. The temporary committee decision has the same effect as a rule.
- (r) **Open source policy:** Source code files must be released open-source 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 access.

## 2.3 Ranking

Each round is composed of several sessions ( $S$ ), and at each session the participating teams receive an identification ranging from  $t_1$  to  $t_n$ , where  $n$  represents the number of teams participating on that session. Each session is comprised of a set of scenarios ( $M$ ), and each scenario also receives an identification ranging from  $m_1$  to  $m_p$ , where  $p$  represents the number of scenarios in that session. A score  $SC_{ji}^k$  is assigned to each team  $i \in T$  ( $T = \{t_1, \dots, t_n\}$ ) at each session  $k \in S$  ( $S = \{s_1, \dots, s_n\}$ ) for each scenario  $j \in M$  ( $M = \{m_1, \dots, m_p\}$ ).

For each session  $k$  and scenario  $j$ , the Selective Minimum ( $SM_j^k$ ) is calculated as

$$SM_j^k = \max(SC_{ji}^k) - ((\max(SC_{ji}^k) - \text{mean}(SC_{ji}^k)) \times 2), \quad (1)$$

and the Maximum Score ( $MS_j^k$ ) is calculated as

$$MS_j^k = n \times SDC, \quad (2)$$

where  $n$  is the number of teams on session  $k$ , and  $SDC$  is the coefficient indicating the step between points among teams (we will use  $SDC = 2$  in RoboCup Rescue 2016 competition).

The maximum value of each step is calculated as

$$MSS_{j/step \in \{1, \dots, MS_j^k\}}^k = \frac{((\max(SC_j^k) - SM_j^k))}{(MS_j^k \times (MS_j^k - step))} \quad (3)$$

To each team is assigned the step value, whose  $MSS_{j/step}^k$  value is lower than the team' score, but the  $MSS_{j/step+1}^k$  value is greater than the team' score.

$$TP_{ji}^k = step \therefore MSS_{j/step}^k < SC_{ji}^k < MSS_{j/step+1}^k \quad (4)$$



The final team points for each scenario and team are calculated as

$$FTP_i^k = \sum_{j=m_1}^{m_p} TP_{ji^k} \quad (5)$$

The final team points is then used to generate a ranking of all the teams for that session. The team with the highest mean points is ranked as first, the second highest as second, and so on.

## 2.4 Hardware & Software 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. The hardware and software configurations are

*Hardware:* Intel i7 CPU processor 3.0GHz or higher, and 8.0 GB or higher RAM

*Operating Systems:* Linux Ubuntu 12.04 LTS (64bit) or higher with Oracle Java 1.7 run-time or higher.

*Simulation Packages:* The simulation packages used in the 2016 RoboCup Rescue Simulation League competition will be available at <https://sourceforge.net/projects/roborescue/>.

*Parameters:* The parameters and their possible values are the same as those in the 2014 Rules, which are listed in Section 8 of the document RoboCup 2014—RoboCup Rescue Simulation League Agent Competition (<https://sourceforge.net/projects/roborescue/files/competitions/2014/rs114-rules.pdf>).

## 3 Technical Challenge Competition

The Technical Challenge competition involves evaluating competing agent teams on different scenarios of the RoboCup Rescue Agent simulation platform (<http://sourceforge.net/projects/roborescue>) developed using exclusively the ADF (<https://github.com/RCRS-ADF/RCRS-ADF>). Specifically, it involves evaluating coordination algorithms enabling teams of Ambulances, Police Forces, and Fire Brigades to rescue civilians and extinguish fires in cities where an earthquake has just occurred. Each team is evaluated

on a set of scenarios that represent different disaster situations (e.g., civilians and fires, major fire in one corner of a city, blocked roads to refuges, damaged platoon agents, fire maps, civilian maps where only fires need to be extinguished or civilians need be saved respectively).

This competition is performed by executing the teams code on a set of scenarios and a score calculated to each scenario. The teams with the highest sum of scores in all the scenarios is the winner of the competition.

A scenario is composed of a map, a set of initial rescue agent/civilian positions, and a set of configuration options for each of the simulator components. Each map is constrained to a maximum of 10,000 roads and 10,000 buildings. The buildings and roads entrances are supposed to be fully connected. A validation tool will be used to check the full connectivity of roads and building entrances in each map. However, teams do not have the right to complain in case roads or building entrances are not fully connected if evidenced that this was not detected by the validation tool.

The RoboCup Rescue Agent Simulation Technical Committee is responsible to elaborate a set of scenarios for the competition and all teams code are executed on the same set of scenarios.

### 3.1 Rules

- (a) **Remote participation:** Remote participation is not allowed in any circumstance. At least one team member must register and be locally present at the venue.
- (b) **Rounds:** The competition is structured into one single round.
- (c) **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 scenario before it starts. All the conditions for a particular disaster situation shall be identical for all the teams.
- (d) **Code submission:** All teams must submit the team' source code (binary code will not be accepted) and adequate compile scripts before the start of the competition. The number and time of submissions and specific requirements will be explained in the brief presentation at the competition setup time at the venue. The Technical Committee has the authority to change the time of submissions and to review all submitted teams' source code.

- (e) **Agents:** Teams shall implement all three types of agents: Ambulance, Fire Brigade, and Police Office. It is the responsibility of the team to ensure that its code connects the correct number of agents to the server.
- (f) **Shared memory:** Agents cannot 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. The Technical Committee may execute each agent of the team in a different virtual/physical machine if the team is suspected of violating this rule.
- (g) **Simulation:** The team must connect in at most 3 minutes all agents to the kernel in order to perform the scenario simulation. 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.
- (h) **Valid map:** The Technical Committee is entitled to define whether a map results is valid or invalid in a scenario. The decision is based on the results of the map and it may be decided that a map is invalid when all the teams score very close in the map.
- (i) **Valid games:** Teams will NOT be entitled to rerun their agent team in most circumstances. It is expected that teams write their agents so that they work correctly. In extreme circumstances teams may have the right of a single rerun. Circumstances that may result in a rerun are:
  - I. A power failure.
  - II. Accidental or deliberate termination of a kernel, simulator or agent process.
  - III. Java Virtual machine crash.

In the case of rerunning, the latest score is used as the official score of the team on that map.

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.

V. Simulator or ADF bug.

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 set of scenarios rerun. Only one Java Virtual Machine crash rerun request is accepted for each scenario.

- (j) **Bugs:** It is the responsibility of each team to ensure that its code works correctly with the provided simulators and ADF. Although the Technical Committee makes every effort to provide a reliable simulation environment, the Technical Committee has no responsibility for any kind of software failure at competition time. Simulator and ADF bugs are not sufficient grounds to request a rerun.
- (k) **Committee decisions:** If a problem arises during a scenario then teams may ask for the chair to resolve the problem. The chair may make a decision on the spot or may refer it to the committee. Chair 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.
- (l) **Comments from teams: Only the team leader** of participating teams can comment and 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. Expect team leaders, other team members are not entitled to comment or make suggestions to the Technical Committee.
- (m) **Bugs exploitation:** 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.
- (n) **Dispute resolution:** If there is an ambiguity in the rule and any unexpected situation happens, a temporary committee composed of

the Technical, Organization and Executive committees members and the local chair have the power to take a decision regarding the issue. The temporary committee decision has the same effect as a rule.

- (o) **Open source policy:** Source code files must be released open-source 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 access.

### 3.2 Ranking

The competition is comprised of a set of scenarios ( $M$ ), in which the participating teams receive an identification ranging from  $t_1$  to  $t_n$ , where  $n$  represents the number of teams participating on the competition, and each scenario also receives an identification ranging from  $m_1$  to  $m_p$ , where  $p$  represents the number of scenarios. A score  $SC_{ji}$  is assigned to each team  $i \in T$  ( $T = \{t_1, \dots, t_n\}$ ) for each scenario  $j \in M$  ( $M = \{m_1, \dots, m_p\}$ ).

For each scenario  $j$ , the Selective Minimum ( $SM_j$ ) is calculated as

$$SM_j = \max(SC_{ji}) - ((\max(SC_{ji}) - \text{mean}(SC_{ji})) \times 2), \quad (6)$$

and the Maximum Score ( $MS_j$ ) is calculated as

$$MS_j = n \times SDC, \quad (7)$$

where  $n$  is the number of teams, and  $SDC$  is the coefficient indicating the step between points among teams (we will use  $SDC = 2$  in RoboCup Rescue 2016 competition).

The maximum value of each step is calculated as

$$MSS_{j/step \in \{1, \dots, MS_j\}} = \frac{(\max(SC_j) - SM_j)}{(MS_j \times (MS_j - step))} \quad (8)$$

To each team is assigned the step value, whose  $MSS_{j/step}$  value is lower than the team' score, but the  $MSS_{j/step+1}$  value is greater than the team' score.

$$TP_{ji} = step \therefore MSS_{j/step} < SC_{ji} < MSS_{j/step+1} \quad (9)$$

The final team points for each scenario and team are calculated as

$$FTP_i = \sum_{j=m_1}^{m_p} TP_{ji} \quad (10)$$

The final team points is then used to generate a ranking of all the teams for the set of scenarios. The team with the highest mean points is ranked as first, the second highest as second, and so on.

### 3.3 Hardware & Software Configuration

One PC will be used to run the simulator components and the agent teams. The hardware and software configurations are

*Hardware:* Intel i7 CPU processor 3.0GHz or higher, and 8.0 GB or higher RAM

*Operating Systems:* Linux Ubuntu 12.04 LTS (64bit) or higher with Oracle Java 1.7 run-time or higher.

*Simulation Packages:* The simulation packages used in the 2016 RoboCup Rescue Simulation League competition will be available at <https://sourceforge.net/projects/roborescue/>.

*Parameters:* The parameters and their possible values are the same as those in the 2014 Rules, which are listed in Section 8 of the document RoboCup 2014—RoboCup Rescue Simulation League Agent Competition (<https://sourceforge.net/projects/roborescue/files/competitions/2014/rs114-rules.pdf>).

*ADF:* The framework is available at <https://github.com/RCRS-ADF/RCRS-ADF>.

## 4 Infrastructure Competition

The Infrastructure competition involves the presentation of already existent tools and simulators of disaster management problems in general. The intent is the evaluation of possible enhancements and expansions of the basic RoboCup Rescue Agent simulator based on the new ideas and concepts proposed in these tools and simulators. The evaluation will be done in a panel and a winner chosen accordingly to a set of factors related to the technical aspects of the tool or simulator and the presentation. The best tool will be selected for further integration with the simulation platform.

### 4.1 Ranking

The score of each team participating in the Infrastructure Competition is given by the sum of the punctuation given by each other team to a set of evaluative factors related to the innovative aspect of the proposal and the presentation's quality and clarity. The team with the highest sum of scores is the winner of the competition. Each member of Technical Committee who

is not a member of a team may participate in the scoring. Each team should participate and evaluate the infrastructure presentation.

## **4.2 Duty of Release**

Teams participating in the Infrastructure competition must release their source code as an open-source project before the competition. Even though a team is accepted to participate in the competition, if it does not release the source code, a complete manual, prepare an easy installation and running script files (for example, `install.sh` and `run.sh`) before the beginning of the RoboCup 2016 (before June 30th 2016), will be disqualified. Thus, it will not be evaluated and considered for winning the prize.

## **Acknowledgments**

We thanks all the former RoboCup Rescue Simulation League members of the Technical, Organization, and Executive Committees for their effort in promote and organize this competition.