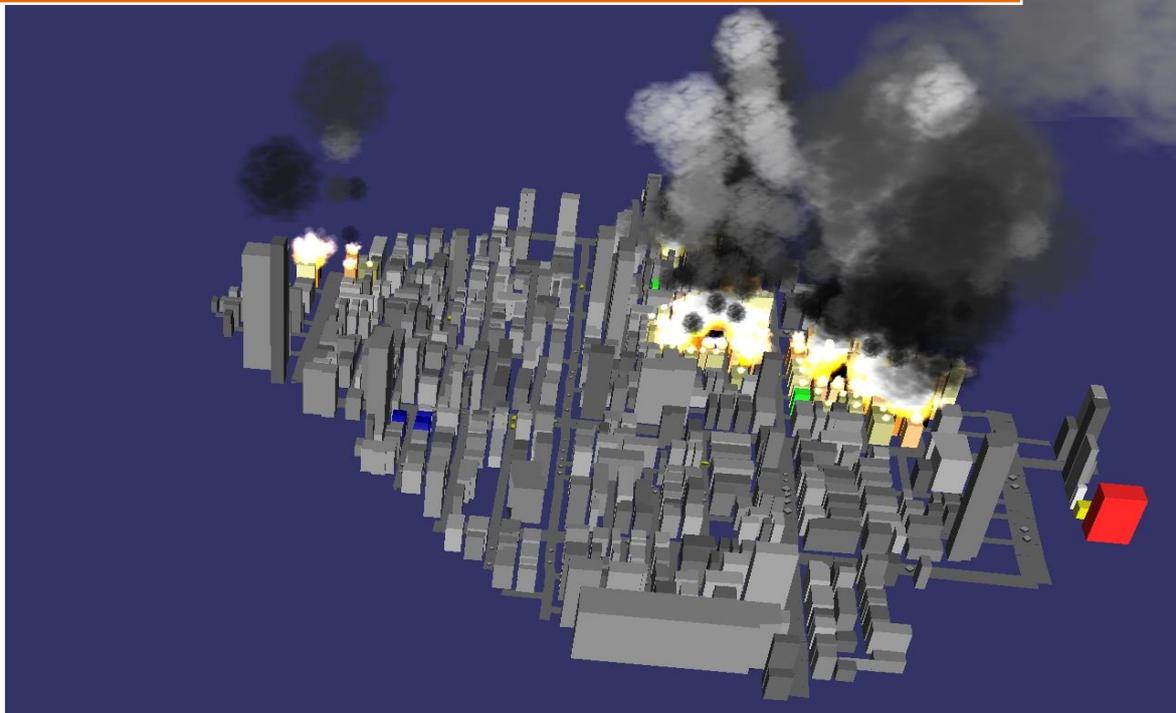


# RoboCup Rescue Simulation Agent Competition



## Rules and Setup (preliminary)

February 2013

### **Authors:**

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

## Table of Contents

Abstract	3
1. Simulation Environment	3
1.1. Smoke simulator	3
1.2. Clear Method	3
1.3. Aftershocks	4
1.4. Gas Stations	4
1.5. Hydrant	4
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	5
5. Maps	5
6. Ranking and Scoring	5
6.1. Scoring	5
6.2. Ranking	5
6.2.1 First Part	5
6.2.2 Second Part	6
7. Rules	6
8. Parameters	8
9. Acknowledgments	13
10. References	13

## Abstract:

A new simulation package v1.1 will be used in the 2013 competition, which includes some simulator server changes and the addition of several new features. A new smoke simulator as well as the new Gas Station and Hydrant entities are introduced, the clear method for Police Force agents is changed, and the aftershocks behavior is modified in order to create blockades during the simulation. Moreover, the ranking and scoring as well as the allowed ranges of simulation parameters are modified.

## 1. Simulation Environment

The new simulation package will be available by March 30th, 2013. Further information about these changes and new features will be provided together with the simulator release.

### 1.1. Smoke simulator

The new smoke simulator will be used in the competition to simulate 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. Smoke Entities will be observable from further away than the line of sight radius.

For each burning building, a specific amount of smoke is produced depending on its fieriness, temperature, wind direction and speed, and a random factor. Existing smoke can spread to neighboring areas depending on the direction and the speed of wind.

The exact position and size of fires would not be deterministically predictable from distance in smoky areas.

An agent may sense smoke in an area if the smokiness of the area exceeds specified threshold (modified by a random factor). Smoke also affects the accuracy of agents' visual sensors and agents in a smoky area will receive noisy perceptions. The amount of noise depends on the smokiness of the area as well as the distance to the agent.

### 1.2. Clear Method

In the current server, Police Force agents are only able to clear blockades congruently from all sides. This method of clearing will be changed and agents will be able to clear blockades in preferred directions, sides and size. Fig. 1 shows the new clearing method.

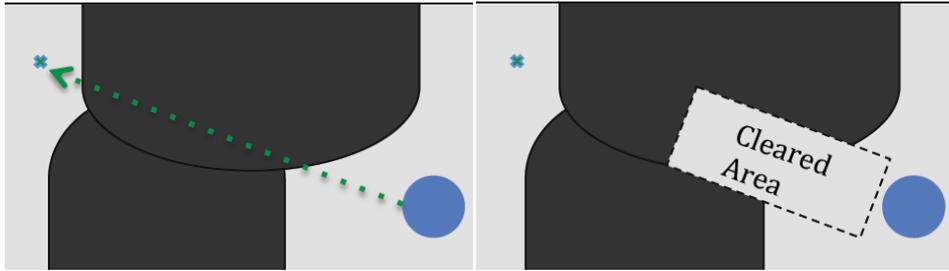


Figure 1: New clearing method

### 1.3. Aftershocks

In the current Collapse simulator, blockades are only generated in the first cycle of simulation at the moment the first earthquake happens. This method is modified and new blockades may be generated during the simulation.

Each scenario is limited to have maximum 10 aftershocks. And the mean of the severity of all aftershocks should not exceed 30% of the initial earthquake.

### 1.4. Gas Stations

Gas Station is a new entity in the environment. When catching fire, it has two major consequences:

- Rapid growth of fires in gas station vicinity
- Destruction of the nearby buildings

### 1.5. Hydrant

Hydrant is a new entity that allows the fire brigades to fill their water tanks. 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) with Oracle Java 1.7 run-time

## 3. Simulation Package

The new simulation package v1.1 will be used in the competition. Software 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. Teams do not have the right to object to the map in a situation in which there are not roads or building entrances not connected if it is evidenced that this results from an error of the validation tool.

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

## 6. Ranking and Scoring

New scoring and ranking formula will be used in 2013 competition as described below.

### 6.1. Scoring

Scores will be calculated using a weighted function of the parameters described in below table.

Parameter	Description
CS	Number of civilians saved
BD	Minimisation of building damage
ES	Speed of extinguishing fires
EE	Efficiency of extinguishing fires (time)
DT	Minimisation of distance travelled
IN	Minimisation of injuries
VD	Speed of discovering all victims
CE	Efficiency of clearing roads

Table 1: Parameters influencing the final score.

### 6.2. Ranking

#### 6.2.1 First Part

New Ranking method which is called Step Ranking will be used in 2013 competition. Teams will be ranked between 0 and  $2 * n$  ( $n$  is the number of teams) in each sessions. Step Ranking is as below:

$$\text{StepDifferenceLinear}(i) = \frac{\text{Best score}}{2 * n}$$

$$\text{Base Score}(i) = \text{Base Score}(i - 1) - \text{StepDifferenceLinear}(i)$$

where Base score(1) = Best Score

Teams with a score between two base scores will be given the upper bound unless another team has already gotten it. If two teams with the same base score exist, the one with a higher score receives the upper bound and the next receives the upper bound – 1.

If two teams have the same exactly score, they both will receive the same upper bound score.

### 6.2.2 Second Part

In order to reduce the impact of incomparable maps (i.e. While scores are too close to compare) the following formula will be used. In this formula a map value is computed for each map according to the variance of the team scores and the initial score of the map.

$$\text{Map Value} = \frac{\text{Teams' Scores Variance}}{\text{Map's Initial Score}}$$

In equation above, the Map value shows how teams' scores were different and it also values the maximum score of the map.

$$\text{Team Final Rank} = \sum \text{Map}_i \text{ Value} \times \text{Team's Rank}$$

**IMPORTANT:** Spite of presenting the formulas at this moment, there may be necessary some adjustments to it after the final simulator release. Therefore, the final Ranking and scoring formulas will be announced by April 15<sup>th</sup>, 2013.

## 7. Rules

- a. **Rounds:** The competition is structured into 4 rounds, each round taking usually one day. There will be two preliminary rounds, one semifinal and one final round.
- 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 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 time, requirements for submission and number of submissions will be explained further in a brief presentation during the competition setup time. Technical Committee has authority to do adjustments in the time of submissions during the competition.  
It is apparent that the Technical Committee has authority to review all 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. **Connection:** There will be two connection times for each scenario simulation: (I) Pre-Computation time and (II) Connection time. The Pre-Computation time is limited in 2

minutes and the team can connect only one agent of each type to the kernel in order to realize any desired pre-computation regarding the maps characteristics. The Connection time is limited in 3 minutes and the team must connect all its agents to the kernel in order to perform the actual scenario simulation. The scenario simulation will begin no later than 3 minutes after the first agent begins its handshake with the kernel in Connection time.

- g. 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 time 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.
- h. Pre-Computation:** Pre-Computation allows an agent of each type to load and use map-specific data in order to perform some preprocessing of these data and store it in a file. In the Pre-Computation phase, only one agent of each type can connect to the server and executes Pre-Computation algorithms. At this phase, connected agents are permitted to write into files. When the Pre-Computation time is finished, the server will be terminated and a new server process will be launched and at this moment all the agents can be connected. At this time, all file permissions, except read permission for previously written files, will be removed. 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. 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.

- j. **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.
- k. **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.
- l. **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.
- m. **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.
- n. **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 16 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 2: 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
<b>random.seed</b>	Seed for random number generator	Anything

Table 3: Allowed ranges for simulation parameters common to all components.

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

Table 4: Allowed ranges for kernel parameters.

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

Table 5: Allowed range for general communication channels configuration parameters.

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

Table 6: Allowed range for voice channel parameters.

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

Table 7: Allowed range for radio channel parameters.

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

	dropout.	
--	----------	--

Table 8: Allowed range for voice and radio communication parameters.<sup>1</sup>

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

Table 9: Allowed ranges for perception parameters.

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

Table 10: Allowed ranges for collapse simulator parameters.

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

Table 11: Allowed ranges for fire simulator parameters.

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	0 – 1

<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.

	will receive a serious injury	
<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	0 – 1

	receive a critical injury	
<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 12: 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 13: 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 14: Allowed ranges for ignition simulator parameters.

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

Table 15: Allowed ranges for traffic simulator parameters.

Parameter	Description
<b>kernel.agents.think-time</b>	See Table 4
<b>kernel.startup.connect-time</b>	See Section 8 – Item f.
<b>comms.channels.count</b>	See Table 5
<b>comms.channels.&lt;x&gt;.type</b>	See Table 6 and Table 7
<b>comms.channels.&lt;x&gt;.range</b>	See Table 6
<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 7
<b>fire.tank.maximum</b>	See Table 11

<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 13
<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 16: 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