RoboCup 2011 – Rescue Simulation League Infrastructure Team Description Paper SBCe_Saviour

Eslam Nazemi, Reza Emami, Mohammad Goodarzi,
Behrouz Shafiee, Behzad Nikbin

Department of Electrical and Computer Engineering
Shahid Beheshti University
Tehran, Iran

nazemi@sbu.ac.ir
{reza.emamii, iman.goodarzi, shafiee01, behzad.nikbin}@gmail.com
http://www.sbcee.net

Abstract. Online Rescue Simulation Launcher (ORSL) is an integrated web-based system which its aim is to make technical connections between teams in RoboCup community. The system is able to run simulation server packages and teams are able to test their team online before the formal competitions. ORSL is something more than a wiki or discussion forum and will be a comprehensive online system which records the teams activities, keep the latest used scenarios and maps and provide teams the ability to follow the community's work and be up to date. This system can also be used by Technical Committee to become aware of teams' improvements and works during a year and become a parameter for ranking the teams for qualification of the league.

1. Introduction

RoboCup Simulation leagues [2] are being held annually in the form of RoboCup World-Wide and Open competitions and several teams are participating in these tournaments. There are teams which are experienced in the leagues and improve their code each year. Also, there are some new teams who join the community and start their work from scratch. According to our experience and feedbacks got from our participation in such events, we perceived that most of the teams work hard just before the competition and they have technical discussions just during the event and then there are no more improvements on the teams in times between competitions. This is mainly because of no active technical communication among teams. There are wikis and discussion forums for each league, but they are just limited to open discussions or some news about the league and future competitions. There is not any place to technically communicate and get involved with other teams. Because of this situation, it usually takes significant times for new teams to reach the standard level of the league and become aware of the suitable team structures and algorithms.

Rescue Simulation (RS) is one of the leagues in RoboCup competitions in which teams try to devise new methods and algorithms for managing the disaster and minimum the damage opposed to a city and its civilians caused by a simulated earthquake

[1]. There is a Rescue Simulation Package which is being developed by the RoboCup rescue community and able to simulate different earthquake scenarios. The most challenging part in rescue competitions is performing in unknown and unpredictable scenarios.

There are lots of papers and also several team description papers which introduce experimental results and improvements on teams' code. These improvements are usually based on the team's old situation. There is not any possibility for teams to compare their performance with other teams working on the same thing before any competition and during the development phase.

Online Rescue Simulation Launcher (ORSL) is an integrated web-based system which its aim is to make technical connections between teams more than a wiki or discussion forum. ORSL is a comprehensive and dynamic system which will facilitate rescue agents development and make the community more active. Moreover, there will be an active repository for teams with several maps, log files and scenario configurations. Teams are able to not only test and improve their strategies but also become aware of their history of work, their logs, runs and their place in the community.

ORSL is consist of four main sub-systems including Core, User Manager, Reporting Server and Map Server. Section 2 is dedicated to ORSL User Manager. ORSL User Core is introduced in section 3 and ORSL Reporting and Map Servers are introduced in sections 4 and 5 respectively.

2. ORSL User Manager

The main responsibility of this sub-system is to keep the record of different users' information (including registration, updating team information and biography and etc.). The other task of user manager subsystem is to manage teams' requests. The last responsibility of this subsystem is to handle different users' access level.

In our x system to main groups of users exist. The first group is ordinary users including different rescue simulation teams. The second group is technical members including the TCs and OCs of rescue simulation league beside system administrators. The point of separating users is to handle access levels.

In the following we name user manager's features for the first group of users:

- The ability to register teams' biography: In our system each team has a separate blog for itself to share their information like team members' photos, logo, sponsors and etc.
- Version controller: each team has the ability to have different version of their binaries. Our system has special part which keeps record of teams' binary history.
- Uploading maps: each team can upload their maps along with different scenarios for that specific map. They have the ability to edit or remove their maps and scenarios also.
- Sending a request to participate in a specific map: each can evaluate their performance on different maps with their different version of binaries.
- The ability to specify access level for binary code (private or public access)

- Comment: each user can mention some tips for other users on different pages.

In this section user manager's features for the second groups (admin level) are provided:

- The ability to approve or disapprove users' registration.
- Creating a tournament: the admin level user can create a tournament with a specific time table.

3. ORSL Core

This subsystem has interaction with rescue simulation server. The core has a queue of different tasks and requests, which are sorted according to their priority. This subsystem executes each task on the server and passes the results of the execution from rescue simulation server to the reporting subsystem. Figure 1 depicts ORSL's architecture and core's interaction with other sub-systems.
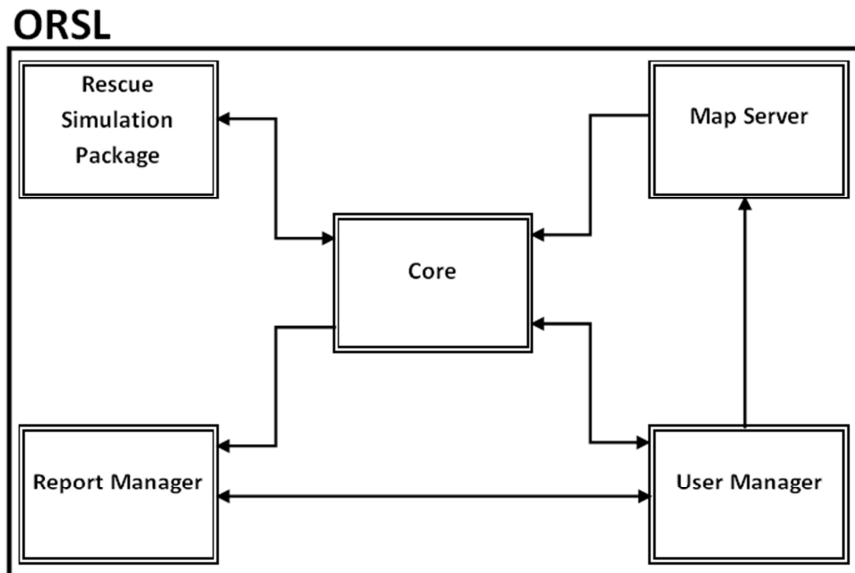

Figure 1. ORSL Architecture

4. ORSL Reporting System

The main responsibility of this subsystem is to receive reports from core and preparing them for presentation in different views. The main features of the reporting subsystem are described in the following:

- Statistical results of different teams on a specific map

- Accumulative or individual reports for each team during a time limit
- Presenting each tournament's final and daily results
- Announcing the top teams of each tournament and best teams in the whole history of tournaments

Special type of report for administrator to evaluate participation of different teams

5. ORSL Map Server

There are infinite scenarios in RoboCup Rescue Simulation league. Several maps is introduced to the community by TC while holding World-Wide and Open competitions. Moreover, individual teams may generate some maps by themselves. ORSL Map Server is a sub-system which allow the administrators to upload the maps used in World-Wide and Open competitions and individual teams to generate and upload their specific maps. Administrators should verify and accept the maps which are uploaded by users in order to be accessed by other users in the system.

6. Conclusion and Future Works

In this paper we presented definitions and system structure for Online Rescue Simulation Launcher (ORSL). The system will be used to keep record of teams' works and histories of runs and logs and provide teams with some useful information about the community. The system can also be used by TC to measure the teams' performance and qualify the high ranked teams who really contribute to the league during a year. For our future work, we hope to extend the system in order to be used in RoboCup Soccer2D and RoboCup Soccer3D leagues.

7. References

1. SBCe_Saviour Team Description Paper for RoboCup Rescue Simulation League, Shahid Beheshti University, Iran (2008)
2. RoboCup Federation Official Website (www.robocup.org)