A Review of Robot Rescue Simulation Platforms for Robotics Education

Josie Hughes[1][ID], Masaru Shimizu[2][ID], and Arnoud Visser[3][ID]

[1] Downing College, Cambridge University, UK
[2] School of Engineering, Chukyo University, Japan
[3] Intelligent Robotics Lab, Universiteit van Amsterdam, NL

Abstract. This review explores a natural learning curve which gives an appropriate RoboCup Rescue challenge at the right age. Children who got involved in the age group 14+ should continue their learning experience until they reach graduate level. To reduce the cost of such a learning experience, simulation is an attractive option in large part of the world. The realism of the simulations and challenges should increase step-by-step, which are supported by more powerful but also more complex interfaces at each level / age-group. The result is a natural learning curve which allows for life-long learning. In this paper, we detail the requirements for such a platform and review a number of different simulation platforms and accompanying interfaces focusing on suitability for use for education rescue robotics. Resulting from this review of simulation platforms a case-study of an example 'gamefied' rescue simulation platform suitable for students at different points along the learning curve.

Keywords: Educational kit · Search and Rescue · Simulation

1 Introduction

Robot rescue is seen as a grand challenge for intelligent systems [13]. The mission of the urban search and rescue (USAR) robot competitions is to increase awareness of the challenges involved in search and rescue applications, and to provide an objective evaluation of robotic implementations [23]. Rescue competitions have been a key part of the RoboCup Competition, and Rescue Simulation Leagues have been introduced from 2000 [2]. Simulation of rescue scenarios enables a more concerted focus on multi-agent collaboration, sensing and mapping.

Although primarily viewed as a research activity, rescue robot simulation has also got the potential to make a significant impact to robotics education, and rolling such an activity out could significantly benefit both the research community and education [22,7]. Rescue simulation provides a means of teaching key robotics and computer science principles in an engaging and meaningful way, whilst requiring minimal hardware development [11,8]. There has been the development of previous rescues robot platforms, most notably CoSpace Rescue

Simulation, which has formed part of the RoboCup Junior Competition. In this robot rescue platform, robots autonomously navigate a small indoor environment with infrared sensors [11]. This replicates the RoboCup Rescue Virtual Robot competition, users can simulate multiple agents, whose capabilities closely mirror those of real robots [2]. The simulated environment models both indoor (building, factory) and outdoor environment (street) that have partially collapsed due to an earthquake [28]. The indoor map includes a maze of walls, doors, different floors, overturned furniture, and problematic rubble which provide various tests for robot navigation, communication, and mapping capabilities [25]. The victims are distributed throughout the environment and the mission for the robots and its operators is to find victims, determine their location in its global map while each robot stays near a victim for further assistance [29].
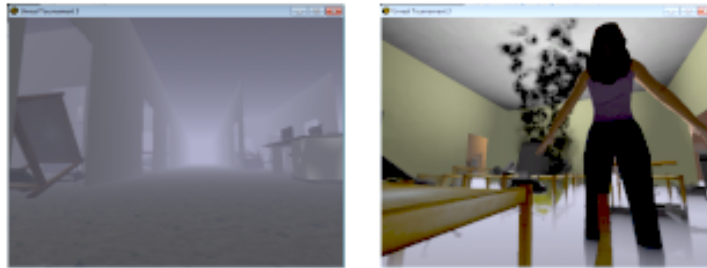
Fig. 1. Example of indoor environments with different types of smoke (Courtesy RoboCup Rescue Simulation Technical Committee [28]).

There is a need for a rescue simulation platform which bridges education to research, specifically targeted at older secondary students (14+) to undergraduate university students [4]. This requires a platform which has a low barrier to entry to allow wide-scale uptake from schools whilst also providing the scope to explore the more interesting and challenging concepts which Rescue Simulation such as navigation and sensing.

In this paper we summarize the requirements for a rescue based education simulation platform, review a number of different simulation platforms and associated interfaces and provide some initial proof-of-concept platforms and approaches.

2   Current challenge

Robotics has been demonstrated to be an excellent tool for teaching computer science and engineering and also exciting and engaging students [27]. However, obtaining suitably robust robots or robot kits can require significant financial outlay by a school, and can also demand a substantial time commitment from already busy teachers who need to become confident using the robot systems.

Fig. 2. A simple CoSpace world in the RoboCup Junior Rescue Simulation competition (Courtesy RoboCup Junior Technical Committee).

Yet, the current challenge is based on simple robots with a limited sensor suite (see Fig. 2), which makes it difficult to continue the learning experience with more advanced topics as computer vision, path-planning and simultaneous localization and mapping problem (SLAM). These are are topics one would expect at the undergraduate level [19].

3 Requirements & Methods

To evaluate the suitability and potential of different simulation environments and interface it is first necessary to define the requirements from both an education standpoint and a research perspective. From these requirements, metrics by which the different simulation platforms can be evaluated and can be determined.

3.1 Requirements

- Free access to the software such that it is fully accessible to students to use, and, not a high requirement for high performance computation facilities.
- Ease of install such that installation does not become a barrier to entry.
- Interface that is intuitive to use.
- Scope in the platform such that it can be used to underpin robotics teaching from fundamentals to higher complexity to provide scope for students to explore and test.
- A pathway that allows easy transition to research based platforms after use of the educational focused platform.

Following the defined requirements, we compared a number of free open source 3D robot simulators. In particular, the simulators Gazebo and Webots are considered, which both have different advantages and disadvantages.

4   Results & Analysis

In this results section we present analysis of two different simulation environments and three associated interfaces. We provide quantitative and qualitative assessment of the simulation environments which tie into the metrics and requirements provided in the previous section.

4.1   Gazebo & Associated Interfaces

Gazebo is the simulation environment integrated in Robot Operating System (ROS), although it was originally developed for the Player-Stage environment [14]. Gazebo development boosted when it was selected as the simulation environment for the DARPA Robotics Virtual Challenge (VRC) [1]. Although the aim of the VRC is also to rescue people, the fundamental difference between RoboCup Rescue and the DRC/VRC is the breadth of capabilities required of the robots [2].

In 2016 Gazebo was selected as simulation environment for the RoboCup Rescue Virtual Robot competition [24]. Previously, the RoboCup Rescue Virtual Robot competition was based on a simulation environment which was based on Gamebots on top of the Unreal Engine [12]. This simulation environment was called USARSim [3], which at the end also provided a ROS-interface [16]. Because also a USARSim interface in Gazebo was created [24], both backward and forward compatibility was guaranteed.

The main advantage of Gazebo is that it actively maintained by the Open Source Robotics Foundation[4], which allows a variety of robots and environments to be simulated. To give an example of the diversity of possible environment; Gazebo was recently used as simulation platform for the DARPA Subterrain Challenge[5], the RoboNation maritime Virtual RobotX competition[6] and NASA's Space Robotics Challenge [10]. The tight coupling with Robot Operating System (ROS) allows the usage of state-of-the-art ROS modules directly to the robots, simulated or real [15]. Yet, the of those state-of-the-art modules requires a deep understanding of robotics; experience that has to build up at the undergraduate and graduate level [6]. An interface has been developed to allow Scratch, a GUI based programming language typically used in schools, to be used to control and interface with ROS topics[7]. This has the potential to provide a far easier and more intuitative interface to ROS and Gazebo.

4.2   Webots & Associated Interfaces

Webots is a simulation environment with a long history [18]. From the beginning, the environment was meant for both education and research. Webots was

---

[4] https://www.openrobotics.org/

[5] https://www.subtchallenge.world/

[6] https://robotx.org/index.php/about/about-virtual-robotx

[7] http://wiki.ros.org/scratch

designed as a sensor-based simulator for mobile robotics [17], which allows to transfer the behavior developed in simulation to be transferred to a real robot, due to the realistic responses of the IR proximity-sensors and a particular vision sensor, the EDI artificial retina.

Recently, Webots has become free open source software, released under the terms of the Apache 2.0 license, which makes it fulfill one of the requirements to be used at high-schools world-wide. In addition, the professional support of the Cyberbotics Ltd. has made installation easy and reliable on Windows, Linux and MacOS platforms. In addition Olivier Michel, the founder of Cyberbotics, has initiated a WikiBook, which was further developed by Fabien Rohrer and Nicolas Heiniger. The WikiBook *Cyberbotics' Robot Curriculum*[8] contains beginner, novice, intermediate and advanced levels.

Table 1. A comparison of the different interfaces to Webots.

	Blockly	Matlab Interface	Python Interface
Ease of Install	Easy	Easy but many toolboxes	Several 'easy' ways
Free/Cost/Open source	Free	Complimentary for RoboCup (Junior) teams	Open source
GUI/Interface			
Suitability for Age <14	Yes	-	-
Suitability for Age 14-18	-	Yes	-
Suitability for Age 18+	-	-	Yes
CV/ML Possible	No	Integrated CV and ML Toolboxes	Interface with OpenCV and ML Libraries
SLAM/Localisation	None	Integrated Robotics System Toolbox	Possible [21]
Sensory Data	None	Integrated Sensor Fusion and Tracking Toolbox	Possible but not standardized
Existing Support/ Online Resources	Forum	Extensive	Community
Level of Support Provided	Minimal	Dedicated	Fragmented

The interface of Webots is intuitive to use [9]. At the beginner level BotStudio is used as interface, which learns the students the concept of an automaton. At the novice level this is used to create simple behaviors as line following. On the intermediate level several behavioral modules are combined, to the level of a full subsumption architecture [5]. At this level also the concept of image processing is introduced. On advanced level one can think of pattern recognition and simultaneous localization and mapping problem (SLAM) [26]. At the advanced level one no longer works in the BotStudio, but directly programs the robot in a programming language as C (or C++, Python, Java, Matlab).

---

[8] https://en.wikibooks.org/wiki/Cyberbotics'_Robot_Curriculum

A number of third parties have created graphical interfaces to the simulator. For the younger age groups, Blockly[9] appears to be the most suitable as it provides a Scratch[10] like environment whilst also allowing the potential for reasonable levels of complexity.

4.3   Comparison between Webots & Gazebo

A comparison can be made between the simulation environments provided by Webots and Gazebo. This is provided in Table 2. This demonstrates how the scope, and support of Gazebo is extensive in comparison to Webots. However, the windows compatability and previously use in educational platforms makes Webots more suited for younger students.

Table 2. Comparison between different simulation environments: Webots and Gazebo

| Simulation Platform	Gazebo	Webots
Physics Engine	ODE, Bullet, Simbody, and DART	ODE: Open Dynamics Engine
Realism	Advanced 3D Graphics	3D Graphics
Sensory Modelling	Sensors and Noise	Some Sensors, lower controllability
Windows/Linux	Linux	Windows/Linux
Gameification	Can be implemented	Can be implemented
Open-Source	Open Source	Free, Open Source
Existing Education Applications	Limited	Extensive, many platforms
Long Term Support	Well-established Online Support	Some Support & many examples

5   Case Studies: Prototype Platforms

5.1   Gamefication in Gazebo

One of the smart tricks to enhance the involvement of students in the current design of the CoSpace challenge [8] is that the scoreboard is integrated in the environment, which increases the suspense and directly gives feedback on the progress. In principle this sort of integration is also possible in Gazebo, yet it had to be demonstrated, which is done in this case study. A Gazebo environment has been developed, which recreates many of the elements which can be found in a CoSpace arena (see Fig. 3).

Gazebo supports a variety of research robots, such as the PR2 and the Atlas robot [1], but has no model for the small direct-drive robots typically used in education. Because the size of the robot directly has influence on the design of the challenges (both objects and obstacles), a new model of a small direct-drive robot was created for this challenge. This robot not only has the size, but also the sensor suite typically used in education (see Fig. 4).

---

[9] https://developers.google.com/blockly/
[10] https://scratch.mit.edu/
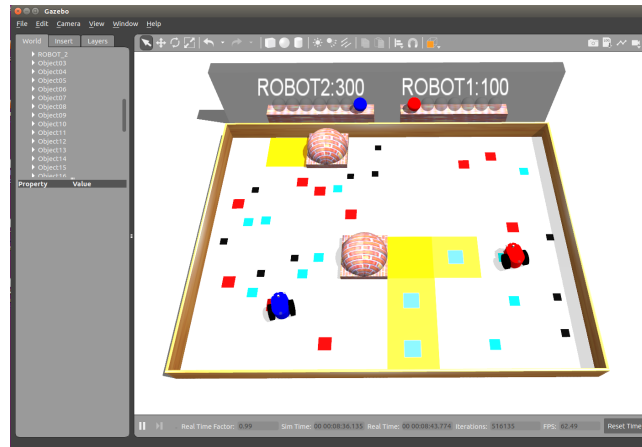
Fig. 3. The creation of a simple CoSpace world in the Gazebo environment.
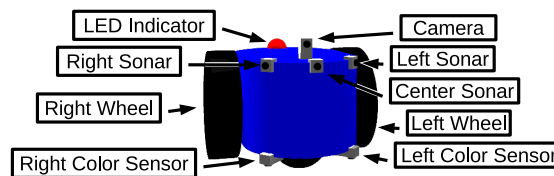
Fig. 4. The Robot functionality.

So, to recreate the important elements which defines the interaction with the world in a CoSpace-like challenge the following functionality had to be added to Gazebo:

1. Two small direct-drive robots, each with an own color.
2. The sensor suite which can be expected on such small robot: a webcam(Fig. 5, three distance sensors and two distance senors.
3. Controllable LEDs to indicate the status of the robot.
4. 2D objects which indicate the location of victims (and disappear when driven over).
5. A score board which indicates how many victims are picked up by team Red and Blue.
6. A game manager who implements the rule of the game, such as creating victims on new (random) locations once there picked up.

A sensor which is not part of the current CoSpace-like challenge, but which could facilitate an interesting challenge for the age 14-18, is the webcam mounted on the simulated robot (see Fig. 5).

The camera feed allows to introduce Computer Vision and Pattern Recognition to those students. An example of such an assignment would be the recog-

Fig. 5. A view image from the webcam on the robot.

nition of a few landmarks on the walls, by combining a simple artificial neural network with a backpropagation algorithm.

The result is a world which resembles one of the CoSpace worlds, which can now be controlled with standard ROS-commands as `cmd_vel`(See Fig. 6).
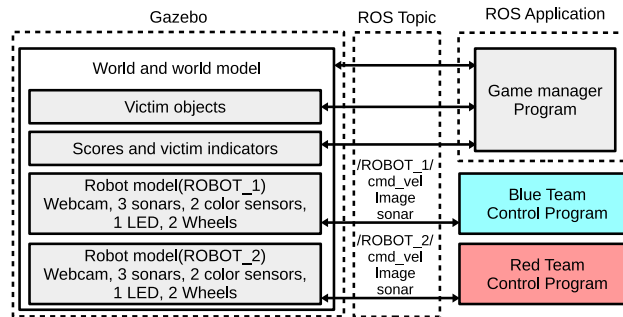
Fig. 6. A connection diagram between the robot models and the team control programs.

This challenge can be easily extended to more advanced challenges in the future. One of the challenges which seems to be on an appropriate level for undergraduates (and in the spirit of the RoboCup Rescue Virtual Robot competition) are the virtual equivalent of the perception tasks as defined in the RoboCup Rapid Manufacturing Challenge, such as visual/thermal activity, motion, colour and Hazmat tests (see Fig. 7).

5.2   An advanced challenge in Webots

A prototype environment has been created based upon the Webots simulation platform. An environment has been created using Python to create a rescue scenario which includes gamefication. Within this environment two E-Puck robots are active (one fore each team), as illustrated in Fig. 8. The environment is more complex than the current CoSpace challenge, because there are now also crate based obstacles (which could be pushed out of the way, but also intentionally or

Fig. 7. The detection of thermal activity in a standard test crate (Courtesy RoboCup Rescue Technical Committee).

accidentally pushed to a location where it blocks the route (of the other team) to the ball deposit area). The victims are now also no longer 2D locations, but 3D objects (balls) which could be 'saved' by moving them to a deposit area (representing a medical post).
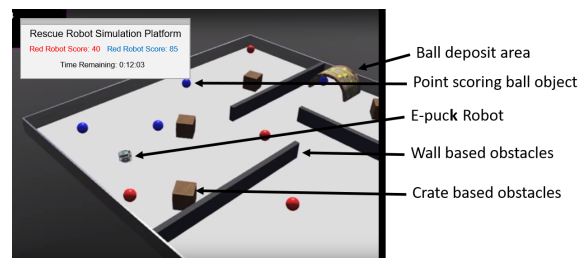
Fig. 8. Webots simulation environment created.

This has been implemented by using the Webots by using the Webots concept of a supervisor node. A set of functions are available for each robot object whose supervisor field is set to true. This has been set for all the ball and deposit areas to allow the status and location of the balls to accessed to determine the scoring. The Supervisor API is then used to access the position, and other state variables of the ball objects. A GUI for the scoring has then been created using Tkinter. This is summarized in Fig. 9.

The robots that are driving around in this new challenge are E-Puck robots. This robots are especially designed for educational purposes [20]. The robots can be controlled with code inside the Webots environment, or controlled by an external program. The same E-Puck Monitor program which can control a real E-Puck via its wifi-interface, can also be used to control the simulated robot (see Fig. 10). The task for the students is to make autonomous decisions based on the sensor information which is available.
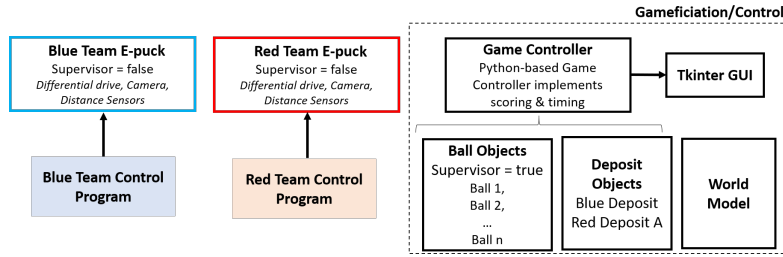
Fig. 9. Summary of the different components to implement the simulation.
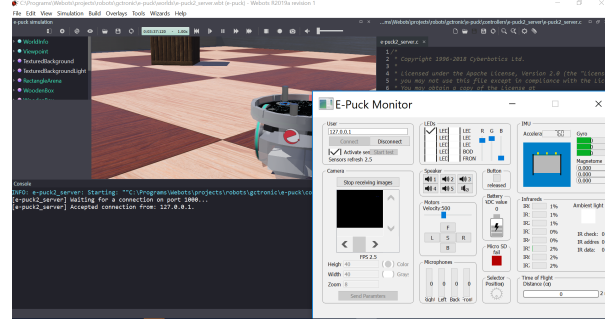

Fig. 10. Connection of the official E-Puck Monitor with Webots.

Going forwards, this platform will be tested with student groups to explore and test how best this meets the requirements. In particular this process will investigate the level of complexity, to explore if it has a sufficiently low entry barrier, yet lets students explore high level concepts.

The ability to customize the rescue scenario and implement gameficiation provides the flexibility and ability to tailor the challenge and over time create increasingly more interesting and complex scenarios.

6   Discussion & Conclusion

In this review, we discussed that suitable robot simulation platforms for different age groups with respect to the requirements required for a natural learning curve. In particular we focused on the appropriate platforms for a RoboCup Rescue Challenge for various age groups. The rescue robot simulation has also got the potential to make an impact on robotics education, and rolling such activity out could benefit both the research community and school. To realize the long learning, new simulation platforms are necessary to bridge education to research, specifically targeted at older secondary students (14+) to undergraduate university students. This could assist in engaging students into Rescue Robotics Research and encouraging participation in leagues such as the RoboCup Virtual Robotic Simulation Competition. And we summarized the requirements for a rescue based education simulation platform, review two suitable different simulation

platforms and associated interfaces and provided some initial proof-of-concept platforms and approaches.

Acknowledgement

References

1. Agüero, C.E., Koenig, N., Chen, I., Boyer, H., Peters, S., Hsu, J., Gerkey, B., Paepcke, S., Rivero, J.L., Manzo, J., Krotkov, E., Pratt, G.: Inside the virtual robotics challenge: Simulating real-time robotic disaster response. IEEE Transactions on Automation Science and Engineering **12**(2), 494–506 (April 2015)
2. Akin, H.L., Ito, N., Jacoff, A., Kleiner, A., Pellenz, J., Visser, A.: Robocup rescue robot and simulation leagues. AI magazine **34**(1), 78–78 (2013)
3. Balaguer, B., Balakirsky, S., Carpin, S., Lewis, M., Scrapper, C.: Usarsim: a validated simulator for research in robotics and automation. In: Workshop on "Robot Simulators: Available Software, Scientific Applications, and Future Trends" at IEEE/RSJ (2008)
4. Blank, D., Meeden, L., Kumar, D.: Python robotics: An environment for exploring robotics beyond legos. In: Proceedings of the 34th SIGCSE technical symposium on Computer Science education (2003)
5. Brooks, R.: A robust layered control system for a mobile robot. IEEE journal on robotics and automation **2**(1), 14–23 (1986)
6. Crick, C., Jay, G., Osentoski, S., Pitzer, B., Jenkins, O.C.: Rosbridge: Ros for non-ros users. In: Robotics Research, pp. 493–504. Springer (2017)
7. Eguchi, A.: RoboCupJunior for promoting STEM education, 21st century skills, and technological advancement through robotics competition. Robotics and Autonomous Systems **75**, 692–699 (2016)
8. Eguchi, A., Shen, J.: Student learning experience through cospace educational robotics: 3d simulation educational robotics tool. In: Cases on 3D Technology Application and Integration in Education, pp. 93–127. IGI Global (2013)
9. Guyot, L., Heiniger, N., Michel, O., Rohrer, F.: Teaching robotics with an open curriculum based on the e-puck robot, simulations and competitions. In: Proceedings of the 2nd International Conference on Robotics in Education. Vienna, Austria (2011)
10. Hambuchen, K.A., Roman, M.C., Sivak, A., Herblet, A., Koenig, N., Newmyer, D., Ambrose, R.: Nasa's space robotics challenge: Advancing robotics for future exploration missions. In: AIAA SPACE and Astronautics Forum and Exposition (2017)
11. Hughes, J.: Robotic rescue simulation for computing teaching in the UK: A case study. In: 2016 IEEE Global Engineering Education Conference (EDUCON). pp. 1051–1055. IEEE (2016)
12. Kaminka, G.A., Veloso, M.M., Schaffer, S., Sollitto, C., Adobbati, R., Marshall, A.N., Scholer, A., Tejada, S.: Gamebots: a flexible test bed for multiagent team research. Communications of the ACM **45**(1), 43–45 (2002)

13. Kitano, H., Tadokoro, S.: Robocup rescue: A grand challenge for multiagent and intelligent systems. AI magazine **22**(1), 39–39 (2001)
14. Koenig, N., Howard, A.: Design and use paradigms for gazebo, an open-source multi-robot simulator. In: Intelligent Robots and Systems, 2004.(IROS 2004). Proceedings. 2004 IEEE/RSJ International Conference on. vol. 3, pp. 2149–2154. IEEE (2004)
15. Kohlbrecher, S., Meyer, J., Graber, T., Petersen, K., Klingauf, U., von Stryk, O.: Hector open source modules for autonomous mapping and navigation with rescue robots. In: Behnke, S., Veloso, M., Visser, A., Xiong, R. (eds.) RoboCup 2013: Robot World Cup XVII. pp. 624–631. Springer Berlin Heidelberg, Berlin, Heidelberg (2014)
16. Kootbally, Z., Balakirsky, S., Visser, A.: Enabling codesharing in rescue simulation with usarsim/ros. In: Behnke, S., Veloso, M., Visser, A., Xiong, R. (eds.) RoboCup 2013: Robot World Cup XVII, Lecture Notes in Computer Science, vol. 8371, pp. 592–599. Springer Berlin Heidelberg (2014)
17. de Meneses, Y.L., Michel, O.: Vision sensors on the webots simulator. In: Heudin, J.C. (ed.) Virtual Worlds. pp. 264–273. Springer Berlin Heidelberg, Berlin, Heidelberg (1998)
18. Michel, O.: Webots: Symbiosis between virtual and real mobile robots. In: International Conference on Virtual Worlds. pp. 254–263. Springer (1998)
19. Michieletto, S., Ghidoni, S., Pagello, E., Moro, M., Menegatti, E.: Why teach robotics using ros? Journal of Automation Mobile Robotics and Intelligent Systems **8** (2014)
20. Mondada, F., Bonani, M., Raemy, X., Pugh, J., Cianci, C., Klaptocz, A., Magnenat, S., Zufferey, J.C., Floreano, D., Martinoli, A.: The e-puck, a robot designed for education in engineering. In: Proceedings of the 9th conference on autonomous robot systems and competitions. vol. 1, pp. 59–65. IPCB: Instituto Politécnico de Castelo Branco (2009)
21. Sakai, A., Ingram, D., Dinius, J., Chawla, K., Raffin, A., Paques, A.: Pythonrobotics: a python code collection of robotics algorithms (2018)
22. Sammut, C., Sheh, R., Haber, A., Wicaksono, H.: The robot engineer. In: ILP (Late Breaking Papers). pp. 101–106 (2015)
23. Sheh, R., Schwertfeger, S., Visser, A.: 16 years of robocup rescue. KI - Künstliche Intelligenz **30**(3), 267–277 (Oct 2016)
24. Shimizu, M., Koenig, N., Visser, A., Takahashi, T.: A realistic robocup rescue simulation based on gazebo. In: Almeida, L., Ji, J., Steinbauer, G., Luke, S. (eds.) RoboCup 2015: Robot World Cup XIX. pp. 331–338. Springer International Publishing, Cham (2016)
25. Takaya, K., Asai, T., Kroumov, V., Smarandache, F.: Simulation environment for mobile robots testing using ros and gazebo. In: 2016 20th International Conference on System Theory, Control and Computing (ICSTCC). pp. 96–101 (Oct 2016)
26. Thrun, S., Leonard, J.J.: Simultaneous localization and mapping. Springer handbook of robotics pp. 871–889 (2008)
27. Toh, L.P.E., Causo, A., Tzuo, P.W., Chen, I., Yeo, S.H., et al.: A review on the use of robots in education and young children. Educational Technology & Society **19**(2), 148–163 (2016)
28. Visser, A.: A guide to the RoboCup Virtual Rescue worlds. Tech. rep., IRL-UVA-16-01 - University of Amsterdam (2016)
29. Williams, A., Sebastian, B., Ben-Tzvi, P.: Review and analysis of search, extraction, evacuation, and medical field treatment robots. Journal of Intelligent & Robotic Systems (Feb 2019)