

Towards an Open Simulation Environment for the Pepper Robot

Florian Lier¹ and Sven Wachsmuth¹

¹CITEC, Bielefeld University, Germany
{fler,swachsmu}@techfak.uni-bielefeld.de

ABSTRACT

Softbank’s Pepper robot recently gained massive traction in diverse domains. On the one hand, the robot interacts with potential customers in shopping malls, stores, at trade fairs and various social events serving as a concierge or “Pepper-as-Promoter”, grabbing attention and fostering customer engagement. On the other hand, the RoboCup federation opened up a completely new league in 2017: the *Social Standard Platform League (SSPL)*. In this new league, the Pepper was chosen as *the standard* social platform that teams will rely on in competitions in the years to come. Lastly, Pepper is an attractive platform for academic institutions since it is, in contrast to other platforms, relatively low priced and does not require a high degree of maintenance or prior knowledge with respect to, e.g., mechanical engineering.

However, designing, developing and implementing social skills for a humanoid robot is not a trivial task that is additionally subject to constant change in the robots code base and configuration parameters for instance. Thus, one of the major drawbacks of the Pepper platform is the lack of a proper simulation environment in order to test new algorithms, high-level task execution strategies, regression testing or simply to provide an additional robot “instance” to compensate peaks in utilization. In this contribution we present our work towards such a simulation environment. We focus on two major topics a) seamless integration with the robot’s ecosystem, e.g., NAOqi and ROS b) basic human-robot-interaction capabilities that can foster behavior modeling and functional regression testing.

KEYWORDS

Simulation for HRI, Robotics, Pepper Robot, Software Engineering

ACM Reference Format:

Florian Lier¹ and Sven Wachsmuth¹. 2018. Towards an Open Simulation Environment for the Pepper Robot. In *HRI ’18 Companion: 2018 ACM/IEEE International Conference on Human-Robot Interaction Companion*, March 5–8, 2018, Chicago, IL, USA. ACM, New York, NY, USA, 2 pages. <https://doi.org/10.1145/3173386.3177088>

1 INTRODUCTION AND PROBLEM STATEMENT

The Pepper robot has recently become a highly requested platform in the commercial sector, as well as in academia. In the commercial context it serves as a promoter, concierge or simply as an attraction that is intended to gain customer engagement — thus, the robot has

already paved its way into “real life”. In academia it is mostly utilized to investigate Human-Robot-Interaction (HRI) design, strategies, patterns and of course, correspondingly, to develop new algorithms and eventually full software stacks. The fundamental research made in academia will, hopefully, help enhancing the user experience and improve our understanding of how people (want to) interact with robots. Designing HRI and implementing the corresponding software systems however is not a trivial task, in both “worlds”. From a high level perspective, designing software for HRI-related tasks requires access to almost all robot capabilities, e.g. locomotion, sensing (cameras, LIDAR, accelerometer, microphones, etc.), speech recognition and production.

At the other end of the spectrum developing and testing HRI scenarios obviously requires a **human-in-the-loop**. In general, simulators can help accelerating the development of scenarios and software systems mainly due to following factors. Using a simulator in order to reconstruct and run experiments in a simulated environment, prior to real-world evaluation, saves time and manual labor [4]. Moreover, simulation helps to compensate peaks in robot utilization, e.g. when a physical robot is used in a user-study for several weeks where changes to its configuration or software system would interfere with the robustness or even the outcome of the ongoing study. Lastly, simulation environments provide a safe way of testing new algorithms, e.g. for autonomous navigation in narrow or crowded spaces [2].

Regarding simulation with a human-in-the-loop in general [4], but especially with respect to the Pepper robot, we identified a major shortcoming. For the Pepper robot, there is no simulator available that includes the following features **a)** virtual sensor data for: cameras (rgb & depth), LIDAR and odometry **b)** basic capabilities with respect to a virtual human avatar that can be actuated and interacted with in the simulated world, and **c)** an interface to change the configuration of the simulated world during runtime, e.g. open and close doors. Lastly, a simulator should provide the *same interfaces* for robot control and sensing as the physical robot in order to provide code consistency when testing and verifying implemented robot behaviors in simulation and subsequently in the real world.

At the time of submission there are only two candidates available for using a simulated Pepper robot. The first option is Gazebo [3]. Gazebo offers the ability to simulate populations of robots in complex indoor and outdoor environments, focusing on realistic physics. Gazebo also includes a model for the Pepper robot¹ but unfortunately support for a virtual human avatar has been removed in prior versions of Gazebo. The other option is Softbank’s Choregraphe [5]. Choregraphe allows to conveniently design basic robot behaviors using visual programming and directly inspect the outcome in a simulator view. However, support for a simulated human avatar is missing, as well as virtual cameras and laser scans which renders

¹http://wiki.ros.org/pepper_gazebo_plugin

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).

HRI ’18 Companion, March 5–8, 2018, Chicago, IL, USA

© 2018 Copyright held by the owner/author(s).

ACM ISBN 978-1-4503-5615-2/18/03.

<https://doi.org/10.1145/3173386.3177088>

Choregraphe unusable for simulation purposes. In the following we will present our ongoing work on a simulator that incorporates the previously mentioned features and requirements.

2 THE SIMULATOR

Our simulator is based on MORSE. Implementation details of MORSE can be found in [1]. The system architecture of our simulator follows the design decisions made in MORSE. The simulation engine is provided by Blender², while interfaces to external software components are realized using diverse middleware implementations. We chose to support the two most popular programming interfaces of the Pepper robot. Thus, we provide full ROS³ and NAOqi⁴ support.

The virtual robot can be controlled by using either native NAOqi API calls or corresponding ROS wrappers. Vice versa, sensory data is exposed via ROS topics, e.g. `/pepper/laser_scan`, `./odometry`. Furthermore, objects as well as a human avatar can be controlled via ROS, e.g., `rostopic pub /open_door . . .`, via an additional GUI application, or lastly, via direct interaction using the mouse and keyboard.

The scenario: the Pepper robot is in front of the apartment and the door is closed. The GUI can be used to open the door.



Figure 1: The robot enters a simulated apartment. On the left: the simulator. On the right: ROS standard navigation stack with sensory input from simulator.

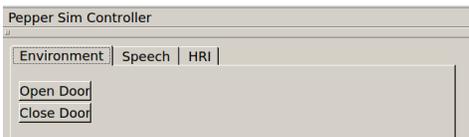


Figure 2: Simulation control GUI. The features, e.g., speech and environment are also available via ROS topics.

As depicted in Figure 1 (left) laser scans are simulated in order to provide input for the ROS navigation stack (right). Our simulator was used to record a map using state-of-the-art mapping approaches in the first place. Once the simulated robot has autonomously reached the navigation goal, virtual camera sensors can be used by external components to, e.g. detect people since images are published via ROS topics (Figure 3).

²<https://www.blender.org/>

³<http://wiki.ros.org/pepper>

⁴<http://doc.aldebaran.com/2-5/naoqi/index.html>

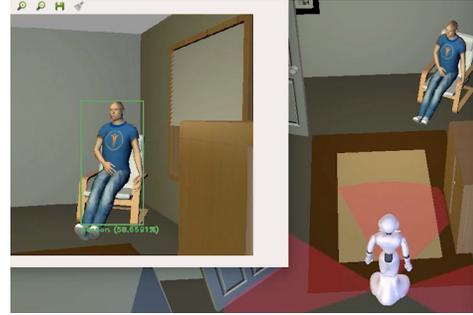


Figure 3: External people detection component (top left) uses the simulated camera images.

As depicted in Figure 2 (speech tab), the simulation also features so-called “fake say” and “fake-recognize” actuators and sensors which can be used via the GUI or command line. These are useful to, e.g., test/verify if a scenario has been modeled correctly — in a behavior state-machine for example.

In our exemplary scenario, the robot detected that the door is open, set a navigation goal and eventually reached the target location. Now, developers can make use of the “fake say” actuator in order to greet the virtual human avatar. Once the fake say command has been issued, a text-to-speech component in our simulator will produce the corresponding output.

Moreover, the “fake-recognize” sensor can be used to indicate that the virtual human answered the greeting. This is realized by sending content, i.e., an utterance, to a configurable ROS topic. Lastly, the human avatar as depicted in Figure 3, can be controlled using the keyboard arrow keys. This function is vital in order to conduct (regression) testing of different navigation strategies, computer vision algorithms and their corner cases for instance.

In summation: this ongoing work towards an HRI-enabled simulator for the Pepper robot a) seamlessly integrates with the robots’ software ecosystem by offering API/middleware compatibility and b) provides advanced functionality in order to model and test HRI scenarios in a safe, efficient and convenient way. The source code is available on GitHub⁵.

REFERENCES

- [1] Gilberto Echeverria, Nicolas Lassabe, Arnaud Degroote, and Séverin Lemaignan. 2011. Modular open robots simulation engine: Morse. In *Robotics and Automation (ICRA), 2011 IEEE International Conference on*. IEEE, 46–51.
- [2] P. Henry, C. Vollmer, B. Ferris, and D. Fox. 2010. Learning to navigate through crowded environments. In *2010 IEEE International Conference on Robotics and Automation*. 981–986. DOI : <http://dx.doi.org/10.1109/ROBOT.2010.5509772>
- [3] N. Koenig and A. Howard. 2004. Design and use paradigms for Gazebo, an open-source multi-robot simulator. In *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (IEEE Cat. No.04CH37566)*, Vol. 3. 2149–2154 vol.3. DOI : <http://dx.doi.org/10.1109/IROS.2004.1389727>
- [4] Severin Lemaignan, Marc Hanheide, Michael Karg, Harmish Khambhaita, Lars Kunze, Florian Lier, Ingo Luetkebohle, and Gregoire Milliez. 2014. Simulation and HRI Recent Perspectives with the MORSE Simulator. In *Simulation, Modeling, and Programming for Autonomous Robots. 4th International Conference, proceedings*, Davide Brugali, Jan F. Broenink, Torsten Kroeger, and Bruce MacDonald (Eds.), Vol. 8810. Springer International Publishing.
- [5] Emmanuel Pot, Jérôme Monceaux, Rodolphe Gelin, and Bruno Maisonnier. 2009. Choregraphe: a graphical tool for humanoid robot programming. In *Robot and Human Interactive Communication, 2009. RO-MAN 2009. The 18th IEEE International Symposium on*. IEEE, 46–51.

⁵<https://github.com/CentralLabFacilities/hri-sim>