WoZ4U: An Open-Source Wizard-of-Oz Interface for Easy, Efficient and Robust HRI Experiments - *DRAFT*

Finn Rietz [1], Alexander Sutherland [1], Suna Bensch [2], Stefan Wermter [1] and Thomas Hellström [2,*]

[1] *Department of Informatics, University of Hamburg, Hamburg, Germany*
[2] *Department of Computing Science, Umeå University, Umeå,Sweden*

Correspondence*:
Corresponding Author
thomas.hellstrom@umu.se

ABSTRACT

Wizard-of-Oz experiments play a vital role in Human-Robot Interaction (HRI), as they allow for quick and simple hypothesis testing. Still, a publicly available general tool to conduct such experiments is currently not available in the research community, and researchers often develop and implement their own tools, customized for each individual experiment. Besides being inefficient in terms of programming efforts, this also makes it harder for non-technical researchers to conduct Wizard-of-Oz experiments.

In this paper, we present a general easy-to-use tool for the Pepper robot, one of the most commonly used robots in this context. The presented software may also be used to develop similar tools for other types of robots. A configuration file, which saves experiment specific parameters, enables quick setup for reproducible and repeatable Wizard-of-Oz experiments. Keyboard shortcuts may be assigned to phrases, gestures, and composite behaviors to simplify and speed up control of the robot. The interface is lightweight and independent of the operating system. Our initial tests confirm that the system is functional, flexible, and easy to use. The interface, including source code, is made commonly available, and it is our hope that it will be useful for researchers with any background who want to conduct HRI experiments.

Keywords: Human-Robot Interaction, HRI experiments, Wizard-of-Oz, Pepper, Interface, open source, Tele-operation

1 INTRODUCTION

The multidisciplinary field of Human-Robot Interaction (HRI) becomes more and more relevant, with robots being increasingly present in our everyday life (Yan et al., 2014; Sheridan, 2016). Especially proximate interaction, where humans interact with embodied agents in close proximity (Goodrich and Schultz, 2008), gains increased attention, with social robots taking on different roles as interaction partners. Naturally, with social robots becoming more and more integrated and involved in our societies, there is also an increased need to investigate the arising interaction patterns and scenarios. Many such investigations are conducted as “Wizard-of-Oz” (WoZ) experiments, where a human operator (the “wizard”) remotely operates the robot, and controls its movements, what it says, gestures, etc.[1] The test participants interacting

[1] Sometimes several wizards control the same robot, as a way to provide real-time responses by the robot, see for example, (Marge et al., 2016).

with the robot, are not aware that the robot is controlled by the wizard. Thus, WoZ experiments allow to investigate human-robot interaction patterns and scenarios simulating a robot with advanced functionalities such as using natural language, gaze and gestures in interaction with a human.

A general tool to design and conduct WoZ experiments is unfortunately not commonly available. Hence, researchers often develop and implement their own tools, customized for each individual experiment. Besides being inefficient in terms of programming efforts, this also makes it harder for non-technical researchers to conduct WoZ experiments.

The WoZ paradigm is an effective instrument for the investigation of hypotheses relating to many typical HRI problems. While originally coined and used in human-computer interaction (HCI) research (Kelley, 1984), the WoZ method was introduced as a technique to simulate interaction with a computer system that was more intelligent than currently possible, or practical, to implement. As such, it may be used to study user responses with hypothetical systems, like fluent, real-world interaction with an embodied agent. For example, the WoZ technique was, and still is, commonly used to evaluate and develop dialogue systems (Dahlbäck et al., 1993; Bernsen et al., 2012; Schlögl et al., 2010; Pellegrini et al., 2014). Research in HRI, in particular social robotics, has picked up the idea, and WoZ has, for at least the last two decades, been extensively used to investigate the interaction between humans and robots. The possibility to study hypothetical systems is valuable also in HRI, in particular when the interaction is unpredictable, and the robot has to adapt to the interacting human to be convincing and engaging.

As Riek shows in a review (Riek, 2012) of 54 research papers in the HRI area, the motivation for, and nature of, the usage of WoZ varies widely. However, in 94.4% of the papers, WoZ was used in a clearly specified HRI scenario, whereas in only 24.1% of the papers, WoZ was mentioned as being part of an iterative design process (Dow et al., 2005; Lee et al., 2009). These numbers illustrate how WoZ is most often used to study and evaluate the interaction between humans and a given robot, and not as a design tool, at least in the HRI community.

In the HRI community, most WoZ implementations are hand-crafted by research groups, specifically for their own, planned HRI experiments, such as the WoZ system developed and used during the *emote* project[2]. For natural reasons, such systems become problem or experiment specific and require reprogramming to be used in different experiments. In (Thunberg et al., 2021), a WoZ based tool for human therapists is presented. The functionality is designed for psychotherapy with older adults suffering from depression in combination with dementia.

A few attempts have been made to develop general WoZ tools for HRI research. The *Polonius* interface from 2011 (Lu and Smart, 2011) is a ROS based system based on both a GUI and scripts for configuration and control of robots. While it describes powerful functionality, there seems to be limited continued development of the system. Hoffman (Hoffman, 2016) describes *Open-WoZ*, a WoZ framework with largely the same ambition as our system presented in this paper. However, it is unclear how much of the envisioned functionality, e.g. for sequencing of behaviors, is implemented, and whether the software is publicly available. Other work address common identified problems with WoZ. In (Thellman et al., 2017), techniques to overcome control time delays are suggested, using a motion tracking device to allow the wizards to act as if they are the robot.

This paper presents an easy-to-use, configurable, robust, and open-source WoZ interface to the HRI community. The resulting system, denoted *WoZ4U*, is implemented and demonstrated on the Pepper

[2] http://gaips.inesc-id.pt/emote/woz-wizard-of-oz-interface/

robot from SoftBank (Pandey and Gelin, 2018), arguably one of the most used social robots. A short video-demonstration of WoZ4U can be found at `https://www.youtube.com/watch?v=BaVpz9ccJQE`.

2 METHOD

To identify the most relevant functionalities for WoZ based HRI research, we reviewed several HRI publications. Riek's meta study (Riek, 2012) of the usage of WoZ in HRI provides valuable insight, and concludes that the three most common functionalities are natural language processing (e.g. dialogue), nonverbal behaviour (e.g. gestures), and navigation.

Many WoZ based HRI experiments require an interactive multi-modal robot (i.e. a robot that processes multiple modalities such as visual and auditory input). In (Hüttenrauch et al., 2006), for example, a WoZ study is performed in which the wizard simulates navigation, speech input and output of robot. Multi-modality in HRI is investigated in WoZ experiments in (Markovich et al., 2019; Sarne-Fleischmann et al., 2017), with a focus on navigation, gesturing and natural language usage. HRI using voice and visual cues are investigated in a WoZ study in (Olatunji et al., 2018). The study in (van Maris et al., 2020) uses pre-programmed natural language utterances to facilitate dialogues between a robot and a test participant, but the wizard controls the reactivity of the robot. HRI experiments often require multi-modality, and the built-in functionalities are often not sufficiently robust. For example, HRI experiments that use pre-programmed natural language functionalities, require the test leaders to control robot's responses (van Maris et al., 2020; Bliek et al., 2020) to ensure a natural flow for the human-robot interaction.

Most HRI experiments require post-analysis of the experiments, where multi-modal data from the test participants has to be analyzed (Aaltonen et al., 2017; Bliek et al., 2020; Singh et al., 2020; Siqueira et al., 2018), and therefore many experiments are monitored in some way, during the experiment by the wizard or afterwards using video and audio recordings. The WoZ4U interface supports post-analysis of experiments with a possibility to record visual and auditory data from the perspective of the robot.

Some robots used for HRI experiment allow interaction via an integrated touch screen. For example, the Android tablet PC on the Pepper robot's chest (see Figure 1) may be used for touch-based input, and to display information such as videos during the experiments (Aaltonen et al., 2017; Tanaka et al., 2015).

To the extent possible, the functionalities mentioned above are included in the developed WoZ4U interface. See Table 1 for a summary of desired and implemented functionality.

2.1 WoZ4U system architecture

The overall architecture of the WoZ4U system is divided into a *frontend* and a *backend*, as illustrated in Figure 2 as an extended deployment UML diagram. The wizard interacts with the frontend through a graphical user interface (GUI) displayed in a regular browser. The GUI communicates with a server in the backend. The server, which is implemented with Python's *Flask* library, reads the configuration file (see Section 2.3), and continuously maps the wizard's actions to NAOqi API calls, which subsequently execute the commands on the physical robot. In the other direction, the server continuously reads the robotic state (e.g. velocities, ongoing actions, interaction mode etc.) from the robot and displays the information in the GUI.

2.2 The Pepper Robot

SoftBank's Pepper robot was designed with the intent of engaging in social interactions with humans, unlike robots that instead focus on physical work tasks (Pandey and Gelin, 2018). With this goal in mind, the Pepper robot has a humanoid appearance, and a size suitable for interacting indoors with humans

Table 1. A summary of desired robot functionalities found in the literature, how this is provided in WoZ4U, and the corresponding part number introduced in Fig. 3.

Desired functionality	**Literature**	**Implementation in WoZ4U**	**Part Nr.**
Documentation of experiments	(Aaltonen et al., 2017; van Maris et al., 2020; Bliek et al., 2020; Singh et al., 2020; Siqueira et al., 2018; Markovich et al., 2019; Olatunji et al., 2018)	Recording of data from cameras and microphones	(1)
Supervision of experiments	(van Maris et al., 2020; Bliek et al., 2020; Shiomi et al., 2007; Law et al., 2017; Breazeal et al., 2013; Bainbridge et al., 2008)	Real-time monitoring of cameras and microphones	(1)
Accepting non-verbal input	(Aaltonen et al., 2017; Tanaka et al., 2015)	Touch interface for the tablet	(1)
Ability to move in the environment	(Chapa Sirithunge et al., 2018; Hüttenrauch et al., 2006; Riek, 2012; Markovich et al., 2019; Sarne-Fleischmann et al., 2017; Shiomi et al., 2007; Walters et al., 2005)	Control of translation and rotation	(2)
Social cues through facial expressions	(Thunberg et al., 2021; Singh et al., 2020; Siqueira et al., 2018)	Control of eyes and LEDs	(6) (7)
Social cues through movements	(Andriella et al., 2020; Thunberg et al., 2021; Bliek et al., 2020; Tozadore et al., 2017; Tanaka et al., 2015; Aaltonen et al., 2017; Sarne-Fleischmann et al., 2017)	Gesture control	(8)
Ability to speak in a natural way	(Funakoshi et al., 2008; Thunberg et al., 2021; Chapa Sirithunge et al., 2018; Thunberg et al., 2017; Bliek et al., 2020; Tozadore et al., 2017; Singh et al., 2020; Riek, 2012; Markovich et al., 2019; Olatunji et al., 2018; Kahn et al., 2008)	Generation of animated speech	(5)
Ability to speak using different voices	(Siqueira et al., 2018)	Setting of speech parameters	(5)
Presenting audio based information	(van Maris et al., 2020; Tanaka et al., 2015; Singh et al., 2020; Riek, 2012)	Play audio files in loudspeakers	(5)
Presenting image-based information	(Tanaka et al., 2015; Bliek et al., 2020; Law et al., 2017)	Display images and videos on tablet	(4)
Varying levels of automated behavior	(Aaltonen et al., 2017; Bliek et al., 2020)	Autonomy settings	(3)

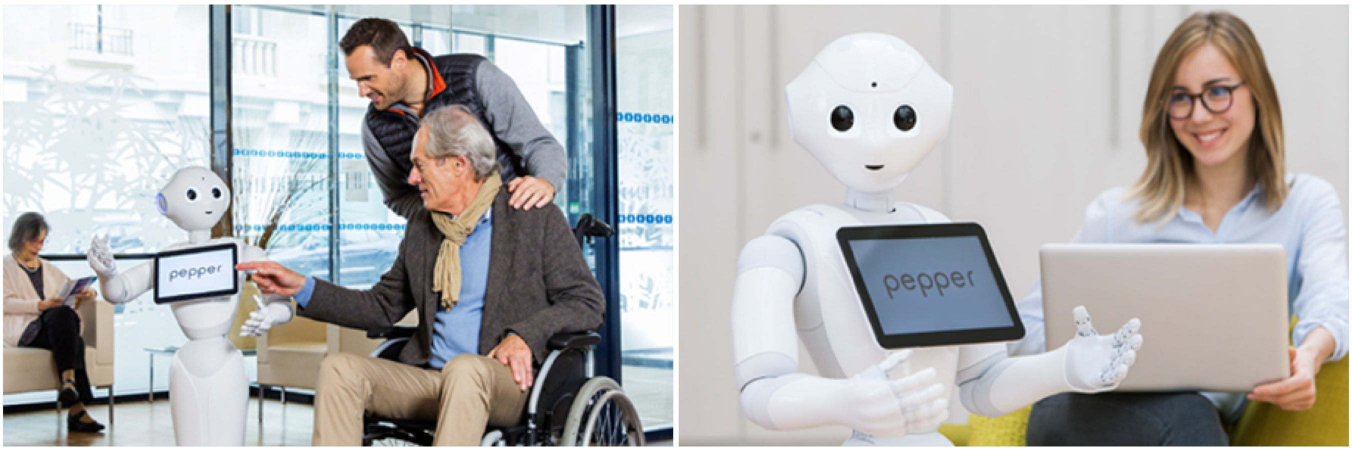

Figure 1. SoftBank's Pepper robot interacting with humans in different environments (images courtesy of SoftBank Robotics).

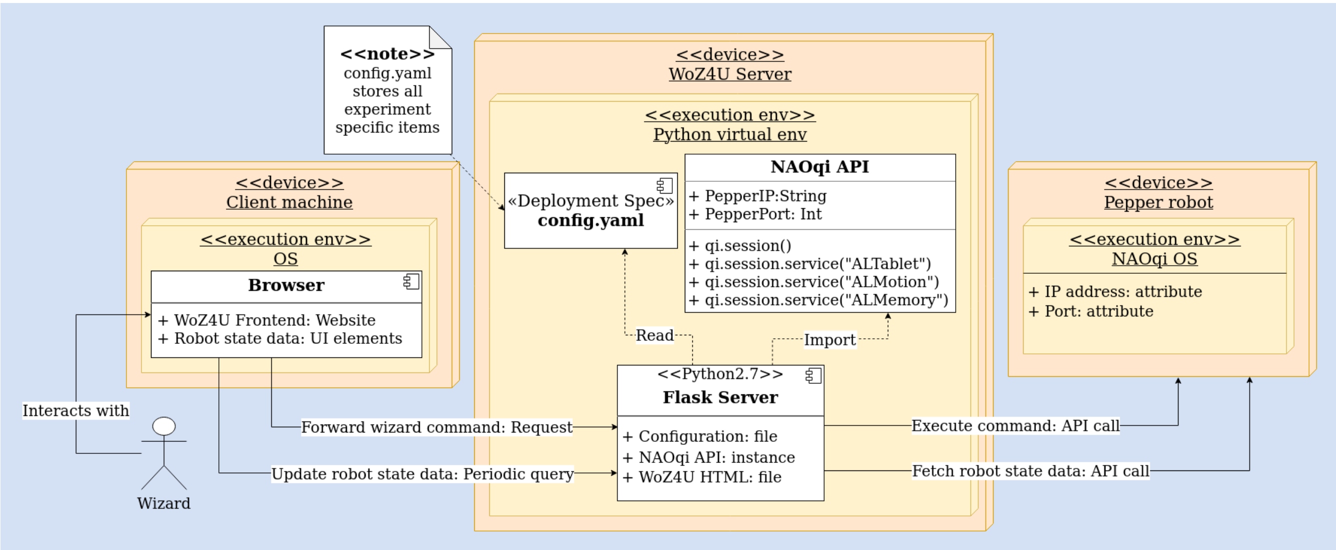

Figure 2. The complete WoZ4U system in the form of an extended deployment UML diagram, showing the connected and interacting components, starting with the wizard's input, and leading to the execution of a command in the robot. The GUI is kept up to date by continuously fetching data from the robot.

sitting down. Figure 1 shows the physical dimension and appearance of the Pepper robot. Pepper is one of the most commonly used research platforms in HRI research, and has also been used in real-world applications, for example as greeter in stores (Aaltonen et al., 2017; Niemelä et al., 2017), and as museum guide (Allegra et al., 2018). The Pepper robot has several functionalities required for social HRI. For example, the humanoid design, head-mounted speakers, multi-colour LED eyes, and gesturing capabilities allow the robot to express itself in different manners when interacting with humans. The mobile base of the robot allows it to traverse the environment, and the internal microphones and cameras allow Pepper to perceive the world, as it engages in social interactions.

The Pepper is usually accessed through the NAOqi Python API[3] or through Choregraphe (Pot et al., 2009). Using the NAOqi API and Python gives access to a multitude of possibilities, but requires good knowledge of the Pepper software. Choregraphe is a drag-and-drop based graphical interface for general

[3] `http://doc.aldebaran.com/2-5/naoqi/index.html`

programming of the Pepper. It is commonly used to design HRI experiments, by programming sequences of interaction patterns such as verbal utterances and gestures issued by Pepper, and corresponding anticipated responses by the test participant. However, Choregraphe is not well-suited when the experimental design requires the robot to respond quickly, adapted to the test participant's, sometimes unpredictable, behavior.

2.3 WoZ4U graphical user interface

Figure 3 shows the GUI of the frontend, as seen by the wizard. For reference, the GUI in the figure is divided into eight parts labelled (1)-(8) and each part is described in more detail in the following subsections. The wizard can control the robot's behavior in real-time, either by clicking on buttons using the mouse, or by using assigned keyboard shortcuts.

The part headings in the GUI are clickable, and open the documentation of the corresponding part of the API, providing an inexperienced user with more detailed information. To simplify configuration for each specific HRI experiment, experiment specific elements such as keyboard shortcuts, gestures, spoken messages, and image/audio/video file names, are defined in a *configuration file* in YAML format[4]. The configuration file is the only part of the system that has to be modified for a new experiment, and by maintaining multiple files, an experiment may be interrupted and continued later with identical settings. This ensures repeatability between multiple users as well as test participants. Several concrete examples of the configuration file's syntax and overall structure can be found in the Listings 1, 2, 3 & 4.

```yaml
gestures:  # each object in the list is made available in the WoZ4U frontend
  -  # list item one:
    title: "Yes"  # Label for UI element
    gesture: "animations/Stand/Gestures/Yes_1"  # NAOqi gesture
    tooltip: "Yes_1 gesture"  # tooltip for UI element
    key_comb: ["shift", "1"]  # keyboard shortcut
  -  # list item two:
    title: "No"
    gesture: "animations/Stand/Gestures/No_1"
    tooltip: "No_1 gesture"
    key_comb: ["shift", "2"]
```

Listing 1. YAML snippet from the configuration file showing how to assign specific NAOqi gestures to buttons in the interface.

```yaml
# Entry in dropdown menu will be create for every list item
# IP addresses must be reachable via network form host machine
pepper_ips:
  - 192.168.10.1
  - 192.168.10.2
  - 192.168.10.3
  - 192.168.10.4
```

Listing 2. YAML snippet from the configuration file showing how to define IP addresses for Pepper robots to which the interface can connect.

[4] https://yaml.org/

```yaml
tablet_root_location: static/tablet_folder/  # place image or video files here, on host machine
tablet_items:                                # notice that the path is relative
  -  # list item one:
    title: "Timeline"  # Label for UI button
    file_name: "climate_change.jpg"  # image to display on tablet
    key_comb: ["ctrl", "shift", 1]  # keyboard shortcut
  -  # list item two:
    title: "Landscape"
    file_name: "landscape.jpg"
    key_comb: ["ctrl", "shift", 2]
```

Listing 3. YAML snippet from the configuration file showing how to make images available for display on Pepper's tablet.

```yaml
# values must be valid for set methods in naoqi API
autonomous_life_config:
  autonomous_state: "solitary"
  tangential_collision: ""  # Leave empty strings to keep default
  orthogonal_collision: ""
  blinking: "False"  # can interfere with manual eye control
  basic_awareness: "False"  # gets enabled by default if pepper enters solitary or interactive state
  engagement_mode: ""
  head_breathing: "False"
  arms_breathing: "True"
  body_breathing: "True"
  legs_breathing: "True"
  listening_movement: "False"
  speaking_movement: "True"
```

Listing 4. YAML snippet from the configuration file defining the autonomous life configuration for the Pepper robot.

During startup, GUI elements are generated based on the given configuration file, and appropriate settings are applied to the robot. Conflicting (duplicated) shortcuts are detected and a warning message is issued.

As many interacting processes are running in the background on the Pepper, especially regarding autonomous life functionality, settings or states on the robot may change automatically, without any commands being issued by the wizard. This may, for example, happen when Pepper switches between its available interaction modes, depending on detected stimuli in its environment. In order to make sure that the state of the Pepper displayed in the browser (as seen by the wizard) and the actual state of the physical Pepper are the same, the NAOqi API is continuously queried and in case of a discrepancy between the interface and the actual physical state, the GUI is updated accordingly.

2.3.1 Connection to robot and monitoring (Part (1))

After startup, the Pepper robot must be connected by selecting the robot's IP address from the drop-down menu in Part (1) (see Figure 3). The IP address(es) must have been previously defined in the configuration file. Note, that the computer must be in the same WiFi network as the Pepper robot, such that the IP address of the robot is reachable via TCP/IP. Once the connection with the robot has been established, the GUI is ready to use. Part (1) also contains controls to provide access to Pepper's camera and microphone data. If activated, a new browser tab with additional on/off controls is opened. To accomplish transmission of microphone data, we extend the NAOqi API, which does not include the necessary functionality. By pressing the button "TABLET ACTIVITY", an additional browser tab is opened, with information on touch

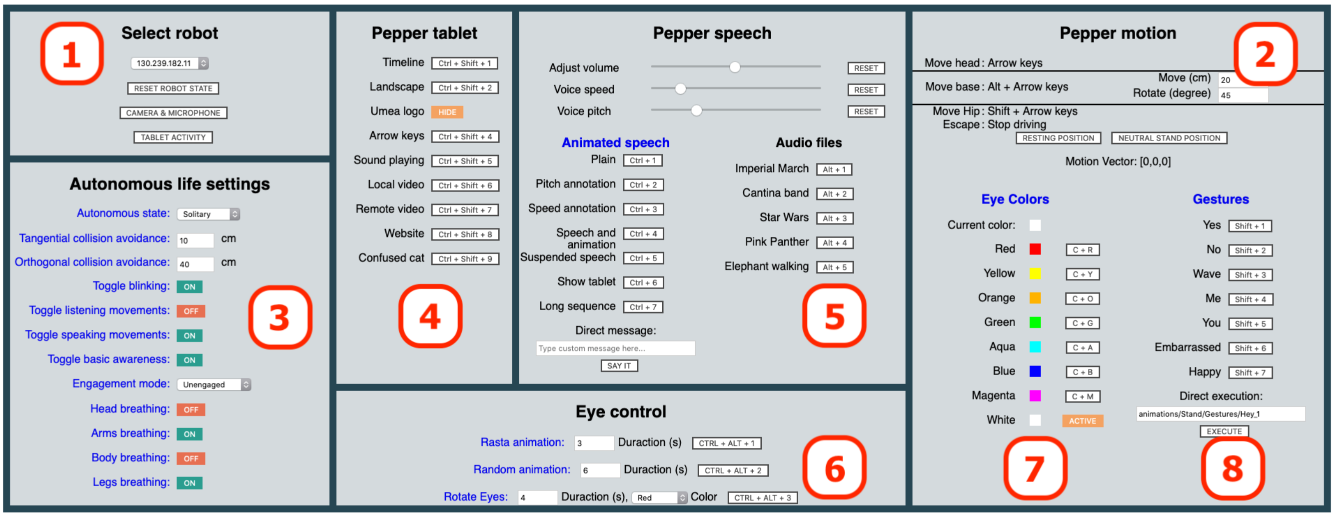

Figure 3. The WoZ4U GUI, comprising parts (1)-(8) with the following functionalities: (1): Connection to robot and monitoring, (2): Motion and rotation of head and base, (3): Autonomy configuration, (4): Tablet control, (5): Speech and audio control, (6) & (7): Eye control, and (8): Gesture control.

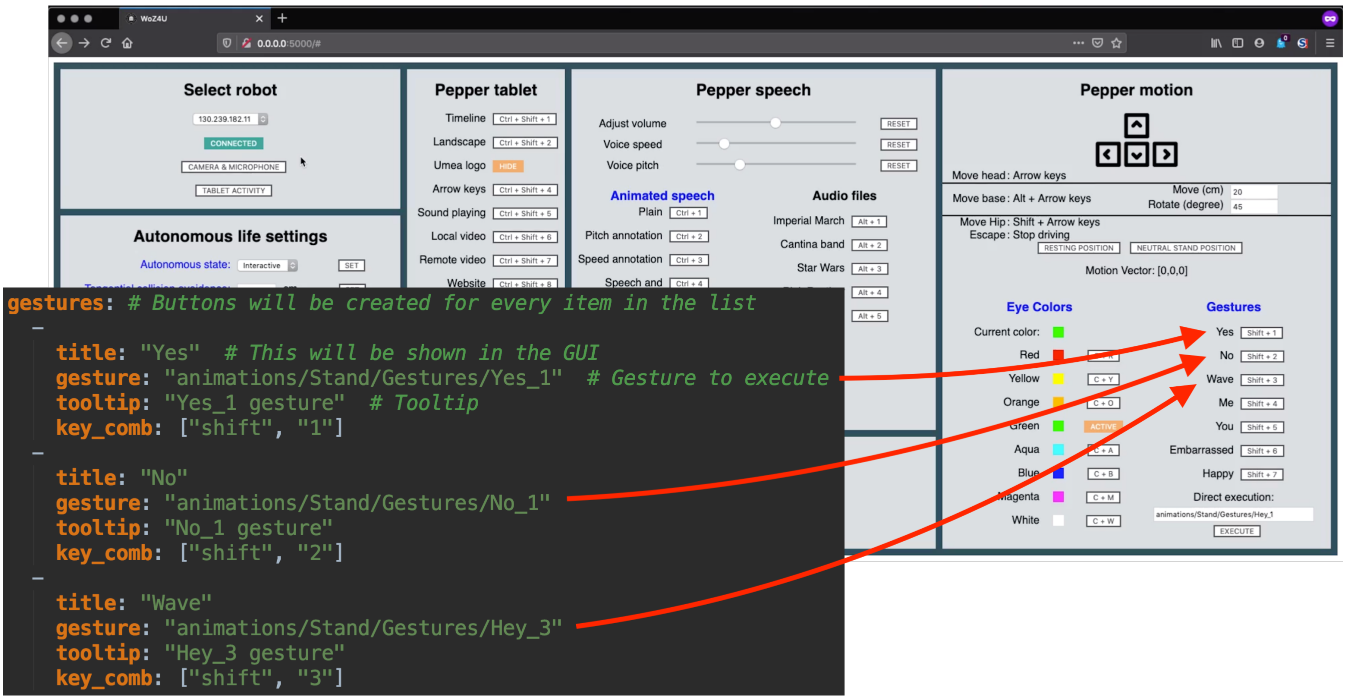

Figure 4. Illustration of how items in the configuration file control the appearance of the GUI.

events from the Pepper's tablet. This enables the wizard to easily monitor touch-interaction on the tablet. The externalization of functionality to dedicated browser tabs prevents overload of the main GUI tab, and also makes it possible to rearrange the screen contents, for example by dragging tabs to separate windows or even to a a secondary screen.

2.3.2 Motion and rotation of head and base (Part (2))

Part (2) of the GUI is used to move the robot using predefined keyboard shortcuts:

- Arrow keys (alone): Rotate Pepper's head around the pitch axis (up/down) and yaw axes (left/right).

- Alt + Arrow keys: With a given increment, drive forward or backward, or rotate around the robot's vertical axis.
- Shift + Arrow keys: With a given increment, rotate the robot's hip around the pitch (forwards/backwards) and roll (sideways) axes.
- Esc: Emergency break, immediately stops all drive-related movements.

The increments for rotation and motion are pre-defined, and may be altered in the GUI.

2.3.3 Autonomy configuration (Part (3))

Part (3) of the GUI addresses the autonomous life settings of the Pepper robot. These settings are important for most HRI experiments, since they govern how the robot reacts to stimuli in the environment and how it generally behaves. The settings often influence each other, sometimes in a non-obvious manner. For example, even if the blinking setting is toggled on, Pepper only blinks when it's autonomous configuration is set to "interactive". The setting is enabled and displayed correctly, but is ignored at a lower level in the NAOqi API. To tackle such non-intuitive behavior, the wizard is informed (by a raised Javascript alert) whenever conflicting settings are made. Pepper's *autonomous state*, which is displayed in Part (3) of the GUI, is affected by independent Pepper functionality that sometimes overrides the API. Hence, observed sudden changes in the state may arise. While it would be possible to enforce Pepper to never diverge from the state specified in the configuration file, this would cause unintended behavior and conflicts with how the NAOqi API is meant to be used.

2.3.4 Tablet control (Part (4))

Part (4) of the GUI provides access to Pepper's tablet. By pressing designated keys, images or videos are displayed on Pepper's tablet. Keyboard shortcuts, and file names (or URLs to websites) may be defined in the configuration file. Showing images on Pepper's tablet and monitoring touch events on the tablet (as described in 2.3.1) can be a powerful, additional mode for interaction with experiment participants. Displaying websites may, for example, allow participants to fill out forms or answer queries. It is possible to flag a specific image as the default image in the configuration file. This causes the image to be displayed on the tablet at startup, and in addition when no other tablet contents is being displayed. If no image is flagged as default, Pepper will display its default animation on the tablet when no other tablet contents is set as active.

2.3.5 Speech and audio control (Part (5))

Part (5) addresses the speech and audio-related functionality of the Pepper robot. Sliders for volume, voice pitch, and voice speed are provided. Text messages and associated keyboard shortcuts may be predefined in the configuration file. The messages may be plain text, or contain tags defined in the NAOqi animated speech module[5]. Table 2 lists tags that change parameters of the speech module. Tags may also be used to execute gestures alongside a spoken message, as shown in Table 3. This makes it possible to define convenient combinations of speech and gestures, that can be activated by one keyboard shortcut. For example, instead of first telling the participant to look at Pepper's tablet, and then executing a gesture where Pepper points at its tablet, a tagged string can accomplish the same thing.

Apart from the text messages, audio files can be played via Pepper's speakers. Due to strong limitations in the NAOqi API, this requires additional preparation beyond the configuration file. Concretely, the audio files (preferably .wav) must be stored on the Pepper robot directly. After storing the audio files on the Pepper, the absolutes paths to those files (on the Pepper), must be provided in the configuration file. This is because the only reliable way to play sound files on Pepper's speaker is to play files that are stored directly

[5] `http://doc.aldebaran.com/2-5/naoqi/audio/alanimatedspeech.html`

Table 2. Tags that can be included in sentences to be spoken by the robot. The range and default values are standardized scales provided by SoftBank.

Tag	Function	Range	Default
\\vct=value	Changes the pitch of the voice	50-200	100
\\rspd=value	Changes the speaking rate	50-400	100
\\pau=value	Pauses speech for value msec.		
\\vol=value	Changes the volume for speech	0-100[6]	80
\\rst	Resets control sequences to default		

Table 3. Control sequences for execution of animations that may be mixed with regular text in sentences.

Control sequence	Function
ˆrun(A)	Suspend speech, run animation A, resume speech.
ˆstart(A)	Start animation A (speech is uninterrupted).
ˆstop(A)	Stop animation A.
ˆwait(A)	Suspend speech, run animation A until it is finished, then resume speech.

on the Pepper. While other solutions are possible in theory (e.g. hosting them remotely), we found them to be unreliable and limiting other API calls, and thus opt for this restrictive solution.

2.3.6 Eye control (Part (6) and Part (7))

Part (6) and Part (7) of the GUI provide control over Pepper's RBG eye LEDs. Part (7) controls a configurable set of colors, which can be assigned to Pepper's eye LEDs, either via button clicks or keyboard shortcuts. Additionally, the available colors and keyboard shortcuts assigned to each color, are configurable. Similar to the tablet items, described in Section 2.3.4, a color can be defined as the default color, which will then be applied to Peppers LEDs at startup. The current color of the eyes is indicated by a colored rectangle in the GUI. This is sometimes helpful since some Pepper gestures change the color of the LEDs. Part (6) of the GUI provides access to the few animations for the Eye LEDs, that come with the NAOqi API. Since there are only three animations available in total, it is not possible to further configure which animations are accessible. Instead, default duration for the animations can be stored in the configuration file. The animation *Rotate Eyes*, which lights the Eye LEDs in a rotating manner, requires a color code as an argument. The color for the rotation can be selected from a dropdown menu, which contains the same colors that are predefined for access in the configuration file.

2.3.7 Gesture control (Part (8))

The Pepper is equipped with a large number of predefined gestures that, when executed, move the head, body and arms of the robot in a way that resembles human motions associated with greetings, surprise, fear, happiness, etc. Such gestures may be executed on demand by the wizard, using keyboard shortcuts. In the configuration file, gestures (defined through their path names) are assigned to keyboard shortcuts, and each gesture is also given a name that is displayed in Part (8) of the GUI. Figure 4 shows an example of how gestures may be specified in the configuration file. Keyboard shortcuts are defined with flags of the form $\mathrm{key\_comb} : [\mathrm{keycode\_0}, \mathrm{keycode\_1}]$.

3 RESULTS

In this paper, we present WoZ4U, a configurable, easy-to-use interface for WoZ experiments. The interface was implemented and demonstrated on SoftBank's Pepper robot, and provides access to most of Pepper's functionality required for WoZ based HRI experiments.

With WoZ4U, the overhead of setting up and executing HRI experiments is greatly reduced, from programming a complete control system to simply adjusting the experiment specific items in a configuration file. This lowers the threshold to run WoZ experiments, not least for researchers with a non-technical background. The WoZ4U interface enables utilization, monitoring, and analysis of multiple input and output modalities (e.g. gestures, speech, navigation) in HRI experiments through *one* GUI (see Section 2.3). Reproducability of experiments, and repeatablility between multiple users as well as test participants is highly important in HRI, and in research in general (Plesser, 2017). In WoZ4U, this is supported by the configuration file which defines and constrains an experiment. This file can be easily shared between researchers and research communities.

The chosen systems architecture, in which the GUI runs as a web application, combines the functionality offered by HTML and JavaScript, with a powerful Python backend communicating with the robot. This modular approach makes the released source code valuable in several ways. It may, for example, serve as a starting point for the development of similar WoZ tools for other robot platforms. The comprehensive Python based interface to the Pepper may also be useful when developing non-WoZ programs for control and communication with the Pepper robot. Since both Python and the NAOqi API are available on all common, modern operating systems (e.g. Debian based Linux, Mac OS, and Windows), WoZ4U is platform independent and can be installed on all these systems (so far it has been tested only with Mac OS).

4 DISCUSSION

The WoZ4U interface may be further developed in several respects. As argued in (Martelaro, 2016), a WoZ interface may, or even should, include not only control functionality but also display data from the robot's sensing and perception mechanisms. This is particularly valuable if the WoZ experiment is a design tool for autonomous robot functionality. For example, a robot's estimation of the test participant's emotion (based on camera data) may be valuable to monitor, in order to assess the quality of the estimation, and also as a way to adapt the interaction with the participant, in response to varying quality of the estimation during the experiment. However, such features would be experiment specific, and not appropriate for a general-purpose WoZ tool. Nevertheless, such functionality can be easily added in additional browser tabs, in the same way as already done for the camera and audio data (Part 1).

The usage of WoZ in HRI is sometimes criticized for turning an experiment into a study of human-human interaction rather than of human-robot interaction (Weiss et al., 2009). One aspect of this problem is related to the fact that the wizard often acts on perceptual information at a higher level than an autonomous robot would do. One approach to prevent this is to restrict the wizard's perception (Sequeira et al., 2016). Another aspect was investigated in (Schlögl et al., 2010), where reported experiments show how the wizard's behavior sometimes vary significantly, thereby potentially influencing the outcome of the experiment. Sometimes, this can be avoided with a non-WoZ approach, i.e. by programming the robot to interact autonomously with the test participants (Bliek et al., 2020). Another approach, which addresses both aspects mentioned above, is to automate and streamline the wizard's work as much as possible. WoZ4U provides keyboard shortcuts and other functionality towards these ends, but this could be further advanced by introducing more complex, autonomous robot behaviors. For example, by supporting the combination of multiple gestures into larger chains of interaction blocks, even larger chunks of the interaction could be automated. This would not only simplify the wizard's work, but also ensure a more consistent and human-robot like interaction.

During the evaluation and testing of the system, we noticed that the detailed documentation of the installation process for the WoZ4U system might still be hard to follow for non-technical users. As such, an automated installation solution would greatly benefit non-technical users and make the system as a whole even more accessible.

AUTHOR CONTRIBUTIONS

The original idea and design requirements were formulated by Suna Bensch and Thomas Hellström. Finn Rietz wrote the majority of the software, while Alexander Sutherland provided expert knowledge on the NAOqi API. Alexander Sutherland, Suna Bensch, Thomas Hellström, and Stefan Wermter provided hardware resources, design ideas, and guidance throughout the entire project. Suna Bensch, Thomas Hellström, and Finn Rietz tested the interface for functionality and usability. All authors contributed in writing this paper.

CONFLICT OF INTEREST STATEMENT

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

CODE AVAILABILITY STATEMENT

WoZ4U is available under the **MIT** license, and downloadable at `https://github.com/frietz58/WoZ4U`. This web page also provides additional information on how to install WoZ4U, and how to write and modify configuration files.

REFERENCES

Aaltonen, I., Arvola, A., Heikkilä, P., and Lammi, H. (2017). Hello pepper, may i tickle you? children's and adults' responses to an entertainment robot at a shopping mall. In *Proceedings of the Companion of the 2017 ACM/IEEE International Conference on Human-Robot Interaction*. 53–54

Allegra, D., Alessandro, F., Santoro, C., and Stanco, F. (2018). Experiences in using the pepper robotic platform for museum assistance applications. In *2018 25th IEEE International Conference on Image Processing (ICIP)*. 1033–1037. doi:10.1109/ICIP.2018.8451777

Andriella, A., Siqueira, H., Fu, D., Magg, S., Barros, P., Wermter, S., et al. (2020). Do i have a personality? endowing care robots with context-dependent personality traits. *International Journal of Social Robotics* doi:10.1007/s12369-020-00690-5

Bainbridge, W. A., Hart, J. W., Kim, E. S., and Scassellati, B. (2008). The effect of presence on human-robot interaction. In *The 17th IEEE International Symposium on Robot and Human Interactive Communication, RO-MAN 2008, Munich, Germany, August 1-3, 2008*, eds. M. Buss and K. Kühnlenz (IEEE), 701–706. doi:10.1109/ROMAN.2008.4600749

Bernsen, N. O., Dybkjær, H., and Dybkjær, L. (2012). *Designing interactive speech systems: From first ideas to user testing* (Springer Science & Business Media)

Bliek, A., Bensch, S., and Hellström, T. (2020). How can a robot trigger human backchanneling? In *29th IEEE International Conference on Robot and Human Interactive Communication, RO-MAN 2020, Naples, Italy, August 31 - September 4, 2020*. 96–103. doi:10.1109/RO-MAN47096.2020.9223559

Breazeal, C., DePalma, N., Orkin, J., Chernova, S., and Jung, M. (2013). Crowdsourcing human-robot interaction: New methods and system evaluation in a public environment. *Journal of Human-Robot Interaction* 2, 82–111

Chapa Sirithunge, H. P., Muthugala, M. A. V. J., Jayasekara, A. G. B. P., and Chandima, D. P. (2018). A wizard of oz study of human interest towards robot initiated human-robot interaction. In *2018 27th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)*. 515–521. doi:10.1109/ROMAN.2018.8525583

Dahlbäck, N., Jönsson, A., and Ahrenberg, L. (1993). Wizard of oz studies — why and how. *Knowledge-Based Systems* 6, 258 – 266. Special Issue: Intelligent User Interfaces

Dow, S., MacIntyre, B., Lee, J., Oezbek, C., Bolter, J. D., and Gandy, M. (2005). Wizard of oz support throughout an iterative design process. *IEEE Pervasive Computing* 4, 18–26

Funakoshi, K., Kobayashi, K., Nakano, M., Yamada, S., Kitamura, Y., and Tsujino, H. (2008). Smoothing human-robot speech interactions by using a blinking-light as subtle expression. In *Proceedings of the 10th International Conference on Multimodal Interfaces* (New York, NY, USA: Association for Computing Machinery), ICMI '08, 293–296. doi:10.1145/1452392.1452452

Goodrich, M. A. and Schultz, A. C. (2008). *Human-robot interaction: a survey* (Now Publishers Inc)

Hoffman, G. (2016). Openwoz: A runtime-configurable wizard-of-oz framework for human-robot interaction. In *2016 AAAI Spring Symposia, Stanford University, Palo Alto, California, USA, March 21-23, 2016* (AAAI Press)

Hüttenrauch, H., Eklundh, K. S., Green, A., and Topp, E. A. (2006). Investigating spatial relationships in human-robot interaction. In *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems* (IEEE), 5052–5059

Kahn, P. H. J., Freier, N. G., Kanda, T., Ishiguro, H., Ruckert, J. H., Severson, R. L., et al. (2008). Design patterns for sociality in human-robot interaction. In *Proceedings of the 3rd ACM/IEEE international conference on Human robot interaction, HRI 2008, Amsterdam, The Netherlands, March 12-15, 2008*, eds. T. Fong, K. Dautenhahn, M. Scheutz, and Y. Demiris (ACM), 97–104. doi:10.1145/1349822.1349836

Kelley, J. F. (1984). An iterative design methodology for user-friendly natural language office information applications. *ACM Transactions on Information Systems* 2, 26–41

Law, E., Cai, V., Liu, Q. F., Sasy, S., Goh, J., Blidaru, A., et al. (2017). A wizard-of-oz study of curiosity in human-robot interaction. In *26th IEEE International Symposium on Robot and Human Interactive Communication, RO-MAN 2017, Lisbon, Portugal, August 28 - Sept. 1, 2017* (IEEE), 607–614. doi:10.1109/ROMAN.2017.8172365

Lee, M. K., Forlizzi, J., Rybski, P. E., Crabbe, F. L., Chung, W., Finkle, J., et al. (2009). The snackbot: documenting the design of a robot for long-term human-robot interaction. In *Proceedings of the 4th ACM/IEEE International Conference on Human Robot Interaction, HRI 2009, La Jolla, California, USA, March 9-13, 2009*, eds. M. Scheutz, F. Michaud, P. J. Hinds, and B. Scassellati (ACM), 7–14. doi:10.1145/1514095.1514100

Lu, D. V. and Smart, W. D. (2011). Polonius: A Wizard of Oz interface for HRI experiments. In *2011 6th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*. 197–198

Marge, M., Bonial, C., Byrne, B., Cassidy, T., Evans, A., Hill, S., et al. (2016). Applying the Wizard-of-Oz technique to multimodal human-robot dialogue. In *RO-MAN 2016 - 25th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)*

Markovich, T., Honig, S., and Oron-Gilad, T. (2019). Closing the feedback loop: The relationship between input and output modalities in human-robot interactions. In *Human-Friendly Robotics 2019, 12th International Workshop, HFR 2019, Modena, Italy, October 24-25, 2019*, eds. F. Ferraguti, V. Villani,

L. Sabattini, and M. Bonfè (Springer), vol. 12 of *Springer Proceedings in Advanced Robotics*, 29–42

Martelaro, N. (2016). Wizard-of-oz interfaces as a step towards autonomous HRI. In *2016 AAAI Spring Symposia, Stanford University, Palo Alto, California, USA, March 21-23, 2016* (AAAI Press)

Niemelä, M., Heikkilä, P., and Lammi, H. (2017). A social service robot in a shopping mall: expectations of the management, retailers and consumers. In *Proceedings of the Companion of the 2017 ACM/IEEE International Conference on Human-Robot Interaction*. 227–228

Olatunji, S., Oron-Gilad, T., and Edan, Y. (2018). Increasing the understanding between a dining table robot assistant and the user. In *Proceedings of the International PhD Conference on Safe and Social Robotics (SSR-2018)*

Pandey, A. K. and Gelin, R. (2018). A mass-produced sociable humanoid robot: Pepper: The first machine of its kind. *IEEE Robotics & Automation Magazine* 25, 40–48

Pellegrini, T., Hedayati, V., and Costa, A. (2014). El-WOZ: a client-server wizard-of-oz open-source interface. In *Language Resources and Evaluation Conference - LREC 2014* (Reykyavik, IS: European Language Resources Association (ELRA)), 279–282

Plesser, H. E. (2017). Reproducibility vs. replicability: A brief history of a confused terminology. *Frontiers Neuroinformatics* 11, 76

Pot, E., Monceaux, J., Gelin, R., and Maisonnier, B. (2009). Choregraphe: a graphical tool for humanoid robot programming. In *RO-MAN 2009-The 18th IEEE International Symposium on Robot and Human Interactive Communication* (IEEE), 46–51

Riek, L. D. (2012). Wizard of oz studies in HRI: A systematic review and new reporting guidelines. *Journal of Human-Robot Interaction* 1, 119–136. doi:10.5898/JHRI.1.1.Riek

Sarne-Fleischmann, V., Honig, H. S., Oron-Gilad, T., and Edan, Y. (2017). Multimodal communication for guiding a person following robot. In *26th IEEE International Symposium on Robot and Human Interactive Communication, RO-MAN 2017, Lisbon, Portugal, August 28 - Sept. 1, 2017* (IEEE), 1018–1023

Schlögl, S., Doherty, G. J., Karamanis, N., and Luz, S. (2010). Webwoz: a wizard of oz prototyping framework. In *Proceedings of the 2nd ACM SIGCHI Symposium on Engineering Interactive Computing System, EICS 2010, Berlin, Germany, June 19-23, 2010*, eds. N. Sukaviriya, J. Vanderdonckt, and M. Harrison (ACM), 109–114. doi:10.1145/1822018.1822035

Sequeira, P., Alves-Oliveira, P., Ribeiro, T., Di Tullio, E., Petisca, S., Melo, F. S., et al. (2016). Discovering social interaction strategies for robots from restricted-perception wizard-of-oz studies. In *2016 11th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*. 197–204. doi:10.1109/HRI.2016.7451752

Sheridan, T. B. (2016). Human–robot interaction: status and challenges. *Human factors* 58, 525–532

Shiomi, M., Kanda, T., Koizumi, S., Ishiguro, H., and Hagita, N. (2007). Group attention control for communication robots with wizard of OZ approach. In *Proceedings of the Second ACM SIGCHI/SIGART Conference on Human-Robot Interaction, HRI 2007, Arlington, Virginia, USA, March 10-12, 2007*, eds. C. Breazeal, A. C. Schultz, T. Fong, and S. B. Kiesler (ACM), 121–128. doi:10.1145/1228716.1228733

Singh, A. K., Baranwal, N., Richter, K., Hellström, T., and Bensch, S. (2020). Verbal explanations by collaborating robot teams. *Paladyn J. Behav. Robotics* 12, 47–57

Siqueira, H., Sutherland, A., Barros, P., Kerzel, M., Magg, S., and Wermter, S. (2018). Disambiguating affective stimulus associations for robot perception and dialogue. In *2018 IEEE-RAS 18th International Conference on Humanoid Robots (Humanoids)* (IEEE), 1–9

Tanaka, F., Isshiki, K., Takahashi, F., Uekusa, M., Sei, R., and Hayashi, K. (2015). Pepper learns together with children: Development of an educational application. In *2015 IEEE-RAS 15th International*

Conference on Humanoid Robots (Humanoids) (IEEE), 270–275

Thellman, S., Lundberg, J., Arvola, M., and Ziemke, T. (2017). What is it like to be a bot? toward more immediate wizard-of-oz control in social human-robot interaction. In *Proceedings of the 5th International Conference on Human Agent Interaction*. 435–438

Thunberg, S., Angström, F., Carsting, T., Faber, P., Gummesson, J., Henne, A., et al. (2021). A Wizard of Oz Approach to Robotic Therapy for Older Adults With Depressive Symptoms. In *Companion of the 2021 ACM/IEEE International Conference on Human-Robot Interaction (HRI '21 Companion), March 8, 2021, Boulder, CO, USA*. 4

Thunberg, S., Thellman, S., and Ziemke, T. (2017). Don't judge a book by its cover: A study of the social acceptance of nao vs. pepper. *Proceedings of the 5th International Conference on Human Agent Interaction*

Tozadore, D., Pinto, A. M. H., Romero, R., and Trovato, G. (2017). Wizard of oz vs autonomous: Children's perception changes according to robot's operation condition. *2017 26th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)* , 664–669

van Maris, A., Sutherland, A., Mazel, A., Dogramadzi, S., Zook, N., Studley, M., et al. (2020). The impact of affective verbal expressions in social robots. In *Companion of the 2020 ACM/IEEE International Conference on Human-Robot Interaction*. 508–510

Walters, M. L., Dautenhahn, K., te Boekhorst, R., Koay, K. L., Kaouri, C., Woods, S. N., et al. (2005). The influence of subjects' personality traits on personal spatial zones in a human-robot interaction experiment. In *IEEE International Workshop on Robot and Human Interactive Communication, RO-MAN 2005, Nashville, TN, USA, 13-15 August, 2005* (IEEE), 347–352. doi:10.1109/ROMAN.2005.1513803

Weiss, A., Bernhaupt, R., Lankes, M., and Tscheligi, M. (2009). The USUS evaluation framework for human-robot interaction. In *Adaptive and Emergent Behaviour and Complex Systems - Proceedings of the 23rd Convention of the Society for the Study of Artificial Intelligence and Simulation of Behaviour, AISB 2009*. 158–165

Yan, H., Ang, M. H., and Poo, A. N. (2014). A survey on perception methods for human–robot interaction in social robots. *International Journal of Social Robotics* 6, 85–119