Note

Performing Accessibility Evaluations for Social Robot

Universal Design of Robots

Note no.	DART/22/22
Authors	Trenton Schulz; Diana Saplacan
Date	23. des. 2022

© Copyright:
Norsk Regnesentral

Authors

Trenton Schulz is a senior research scientist at the Norwegian Computing Center. Diana Saplacan is a research scientist at the University of Oslo, Department of Informatics in the Robotics and Intelligent Systems Group.

Norsk Regnesentral

Norsk Regnesentral (Norwegian Computing Center, NR) is a private, independent, non-profit foundation established in 1952. NR carries out contract research and development projects in statistical-mathematical modelling, geomodelling, earth observation, machine learning, image analysis, artificial intelligence, language technology, digital security, digital inclusion and digital transformation. The clients include a broad range of industrial, commercial and public service organisations in the national as well as the international market. Our scientific and technical capabilities are further developed in cooperation with The Research Council of Norway, the European Research Council, and key customers.

University of Oslo

The University of Oslo is a public research university located in Oslo, Norway. It is the highest ranked and oldest university in Norway. It is consistently ranked among the top universities in the world and as one of the leading universities of Northern Europe; the Academic Ranking of World Universities ranked it the 58th best university in the world and the third best in the Nordic countries. In 2016, the Times Higher Education World University Rankings listed the university at 63rd, making it the highest ranked Norwegian university. The Department of Informatics at the University of Oslo is the oldest and largest department for informatics in Norway. The department was in 2017 ranked number 1 in Norway, 3rd in Europe, and 12th in the world in Computer Science and Engineering by Academic Ranking of World Universities. Famous researchers that have been associated with the department includes Turing Award winners such as Kristen Nygaard, Ole-Johan Dahl and Donald Knuth.

Acknowledgment

This work was done as part of the Universal Design of Robots (UD-Robots) project. The UD-Robots project was funded by a grant from the Forum for Universal ICT (UnIKT). UnIKT strengthens work with digital participation and helps ensure that digital initiatives can benefit more people. UnIKT will also stimulate good, inclusive digital projects aimed at the public. The UnIKT program was established in 2013, with a forum and a grant scheme for universal design of ICT. D. Saplacan wishes also to thank her employer, University of Oslo, and to the Vulnerability in Robot Society (VIROS) research project funded by the Research Council of Norway (Grant Agreement No. 288285), for giving her the opportunity to work with the UD-Robots project.

Title	**Performing Accessibility Evaluations for Social Robot**
Authors	**Trenton Schulz; Diana Saplacan**
Quality assurance	Wolfgang Leister
Date	23. des. 2022
Year	2022
Publication number	DART/22/22

Abstract

This note is a compilation of several evaluations that were done as part of the UD-Robots project. The method used is based on WCAG-EM and initially introduced by Al-Qbilat. We have given it a name: Accessible Socially Assistive Robots Evaluation Method (ASAR-EM). We evaluate 5 robots: a NAO, a Pepper, AV1, Berntsen, and Tiago, and present the results from the evaluation.

Keywords	Social robotics; universal design; accessibility; evaluation
Target group	Researchers in universal design and social robotics
Availability	Public
Project number	320685
Research field	Digital Inclusion
Number of pages	36
© Copyright	Norsk Regnesentral

Table of Content

List of figures

1 Introduction

The focus of the document is going over a robot and its context to find potential barriers that could make the robot in-accessible for people with disabilities. Since universal design is a process for creating an object and the actual designed object itself (Schulz, 2014), this method below can be part of a universal design process for robots leading to a universally designed robot.

Addressing these barriers is necessary but is not sufficient to find out if the robot is universally designed. It likely will also require some sort of user testing with people with different disabilities to see if it works for them. Identifying and addressing the barriers found using this method should be done before the robot is sent out for a user evaluation with people with disabilities. This is to ensure that an evaluation with people with disabilities has the most benefit for people running the evaluation and the people with disabilities participating.

2 Finding the accessibility barriers for A robot

We will start with the method for finding barriers that is based on Chapter 4 from Malak Al-Qbilat's Ph.D. thesis (Al-Qbilat, 2022). This is, in turn, based on the Website Accessibility Conformance Evaluation Methodology (WCAG-EM) method defined by the World Wide Web Consortium(W3C Working Group, 2014). The methodology was a systematic way of going through a website to see how it meets WCAG 2.0 requirements. The basic steps of the process are:

1. Define the evaluation scope: this is for defining what will be evaluated, whet level of the WCAG to follow, and any additional requirements.

2. Explore the target website: this is a process of exploring to determine the functionality of the web page and the different kind of webpages that will be used.

3. Select a representative sample of the webpages: After exploration, select a sample of the webpages to evaluate, this can either be done randomly or in a structured way, this is dependent on the size of the website that will be evaluated, but it is important to document it and ensure that it is representative.

4. Audit the selected sample: Perform a WCAG evaluation on the sample based on the scope defined in Step 1.

5. Report the results: Results should be reported in a structured format so that others can understand what was done.

Al-Qbilat (2022) adjusted this method to work with robots. The method that she proposes is:

1. Define the evaluation scope. This would be a way of defining the context for use of the robot and other things. Define the input and output methods are to be used and the services that will be evaluated.

2. Explore the target robotic application (software and hardware): Get an idea of what the robot can do and what needs to be evaluated.

3. Audit the robot application: check all initial interaction components and check all complete processes. One could also use guidelines that exist in this situation as part of the audit process.

4. Report the results (the elicited accessibility barriers).

Al-Qbilat did not give this method a name. For this document, we will call it the *Accessible Socially Assistive Robot Evaluation Methodology* (ASAR-EM). Al-Qbilat defined this method before she had adapted the WCAG in her thesis, but she argues that this method can work in different ways of understanding the situation. Whether it is through observation and interviews, creating personas and scenarios, or through evaluation of an expert (Al-Qbilat, 2022). Further, Al-Qbilat only got to try her guidelines for one robot, and she also updated her guidelines to include information from WCAG 2.1. It would therefore be useful to try and use Al-Qbilat's updated guidelines for socially assistive robots in ASAR-EM processes (for example, as part of Step 3) for a variety of robots and contexts to see how well it works.

One reason to use the ASAR-EM is that it provides context and scope to the evaluation of a robot. Since robots are a combination of software and hardware, there are many design decisions that need to be made in their construction and this naturally leads to limitations in certain situations. So, a robot may not be accessible in some use cases or modalities due to its limitations, but this might not be relevant in another use case. Regardless, it is still useful to look and understand the robot's capabilities so that one can be aware of these issues when designing a solution. The first two steps in the ASAR-EM method gives us an opportunity to help define where the robot is going to be used; what inputs, outputs, and services it uses; and explore how well this context works with the robot. This initial work can basically include an inventory of the robot's abilities. If there are certain robots are popular, (e.g., a robot such as Pepper or Nao) the information can be a starting point for a new ASAR-EM in a different context.

Robots may also provide additional interfaces for interaction. Although social robots in theory should provide interaction that is close to human social interaction, such as speaking or gestures, there can also be different mobile applications, web pages, buttons, or other methods that might be used for interacting with the robot. If one uses a mobile phone app, web page, or some other software actively to interact with the robot, that software needs to be examined too using something like the WCAG.

3 WCAG-EM reporting tool & Adaption

In addition to the document, the W3C/WAI have produced a tool that follows the WCAG-EM methodology (Abou-Zahra et al., 2022). This tool is recommended by the W3C/WAI along with "light WCAG checks" to use a starting point for evaluating the accessibility of an existing website. The tool itself is open source and runs in an evaluator's web browser with no connection to the internet. It has one neat feature that you can collect a list of URLs from the web page and then it can use that to randomly select your sample of web pages. On the other hand, the tool simply offers a structure for following the methodology and that text is written in the correct place. One could create a template in your favorite word processor or text editor and have near-equivalent experience.

Like the basic idea of the WCAG-EM can be made to work as an ASAR-EM, we can use the reporting tool and adjust it to our needs. Below, we take each of the main points from the Reporting Tool and adjust the fields to work for a robot evaluation.

3.1 Define the scope

This consists of several different fields:

1. Robot name:
   The robot (or application's) name.

2. Scope of the robot (what we will include and what we will not include).
   Document how and what the robot will be used for. This helps establish the context for the robot's use.

3. Conformance targets
   This depends on the robot and the different interfaces that are used for the robot. For interfaces like the app and websites, we can use the WCAG 2.1, for the robot itself, we will use the guidelines from in appendix in the Ph.D. thesis from Al-Qbilat (2022).

4. Accessibility support baseline: (what the goal is to support)
   For apps and websites, pick an accessibility level such as A, AA, or AAA. For robots where we use Al-Qbilat's guidelines, we can also use the same letters for conformance. In 2022, it was expected that ICT-solutions in Norway could match at least A or AA in WCAG 2.0 with some exceptions for audio descriptions and captions. We can start by putting things at Level AA.

5. Additional evaluation requirements.
   Any additional evaluation requirements that aren't included under the conformance targets from above.

3.2 Explore the robot application

In general, this is an area for taking notes about the process as you explore the robot. One gathers material that will be used as a basis for testing later. You can add extra notes on the technology. This normally may require input from different people involved with the scope of the robot.

The original reporting tool has a split between “essential” pages and “variety” pages. That is, a website might have pages that are essential for using the site and others that are often not visited but still important for the site. For example, a banking website would likely classify the login page and balance page as essential, while marketing pages about new loans might be classified as variety. This distinction may not be as relevant in the robot context, at least for the robots that we have evaluated in this document. In the future, such distinctions might be more important. Especially since the robot may have “fun” features that it could use but are not part of its essential program.

For the robot, this should also be where one takes stock of the capabilities of the robot.

3.3 Select a representative sample

The tool provides a mechanism to select random pages (which is why you need to define them in the previous step). Otherwise, you can pick them yourself. Since there we do not have a collection of URLs, we just document what we evaluate. On the other hand, most of the robot applications are currently not as complex as most of the

3.4 Audit the sample

This is where we do the evaluation.

3.5 Report the findings

Include the report title, who commissioned the evaluation, who the evaluator was, an executive summary, and the results. The reporting tool has an option to include an archive of the web page content. For other media, alternative methods must be used if there is a desire to include the content.

3.6 Trying out the ASAR-EM method

This method seems promising. some of the items, like defining the scope, can be done in bulk, at least when one needs to consider multiple robots. Once the first part is done for each robot, it is simply a matter of going through each item. Here are the different robots that we tried the method out on.

1. NAO: The remote for the ROSA project plus the setup process

2. Pepper: The reception experiment

3. AV1: Setup and using the device in a classroom context.

4. Berntsen: Looking only at the teleoperation solution

5. Tiago :: The experiment or context at Care+

4 NAO

4.1 Scope

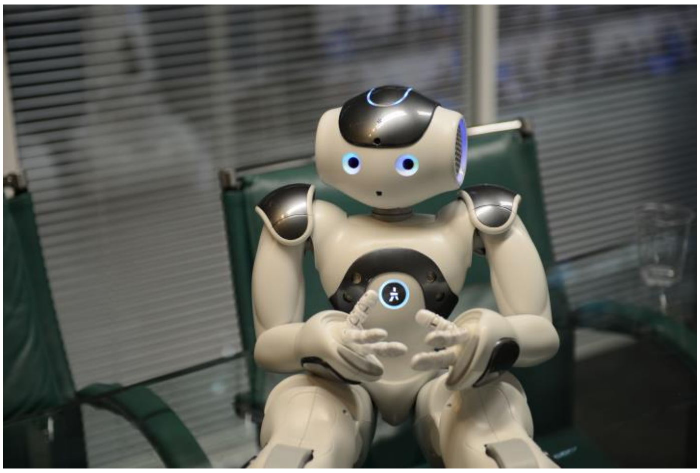


Figure 1: The NAO Robot (Photo Credit: Stephen Chin)

1. Robot name: NAO (Pictured in Figure 1).

2. Scope of the robot: The robot is to be used in the ROSA project, which aims to use a robot to help in teaching social and communication skills (Schulz & Skeide Fuglerud, 2022). There is more functionality that is planned, but for now, we limit ourselves to the remote-control functionality. This is a set of functionalities that can be run from a web browser. The current remote control has nothing to touch. It primarily is an output device meant to serve as a couch/personal trainer/classmate when working with the device.

3. WCAG version and conformance target. We will aim for the standard level of conformance, which would be WCAG 2.1 AA.

4. Accessibility support baseline: We will try to support the browsers and mobile devices that will most likely be used for the evaluation. That is, we will go with Microsoft Edge, Chrome, Firefox, and Voiceover on the device.

5. Additional evaluation requirements. There is a bootstrapping issue that the robot needs to get on the network. So, we will also include the built-in setup page and the robot settings application. Primary goal is to find accessibility issues.

4.2 Explore the robot

4.2.1 Information about NAO

We used the NAO[6] for this evaluation. The basics of NAO are:

- Degrees of freedom: 25

- Microphones: 4

- Speakers: 2

- Lights: 42 LEDS distributed between the top of head, eyes, ears, and feet

- Cameras: 2

- Tactile sensors: 3 capacitive sensors on the head, one push button on the chest, 3 capacitive sensors on each hand, and two feed bumpers

- Languages: 21 languages with varying levels of support.

4.2.2 The remote control

The website is straightforward. It consists of several buttons that go through and run actions on the robot (Figure 2). The actions are divided by categories, but the list is:

- Stand up,
- Sit down,
- Greet,
- Say good bye,
- Say some text
- Tell the student to go to the next task
- Ask the student to try again
- Tell the student the task was correct
- Perform one of four dances: a robot dance, a chicken dance, a disco dance, or the macarena
- Adjust the volume up or down
- Check the connection with the robot
- Shutdown the robot

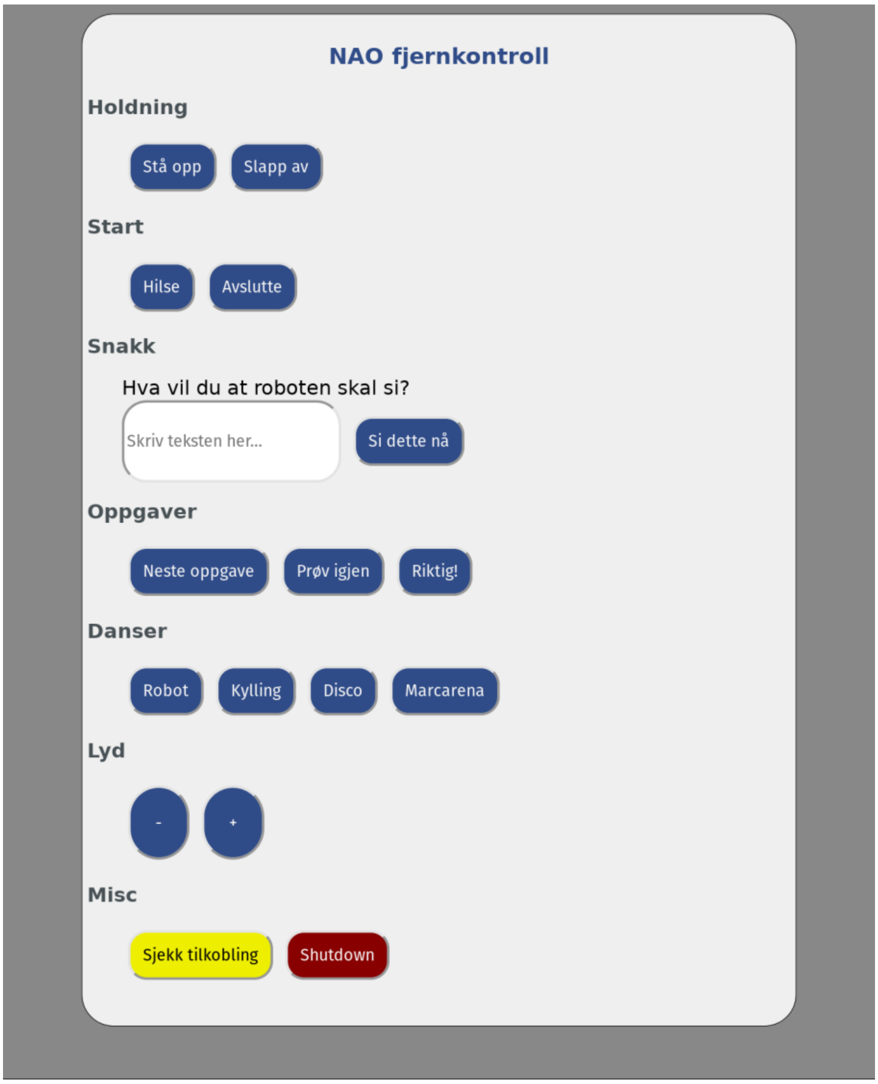


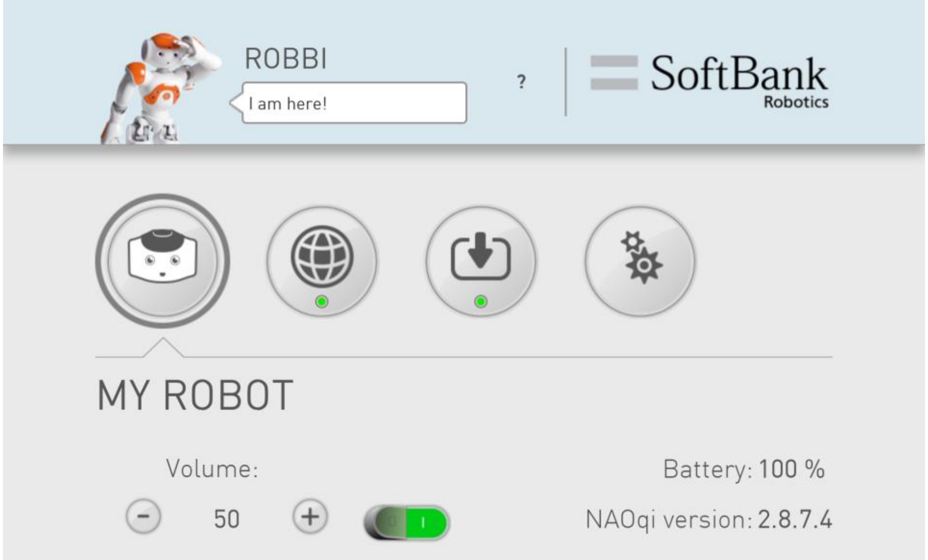
Figure 2 The remote control used for NAO in the ROSA project.

4.2.3 The configuration page

The configuration page (Figure 3) is available by default on the robot, but it is deprecated. They recommend using a separate program called *Robot Settings* that is available on Linux, Windows, and macOS. I've downloaded it for macOS, but it is difficult to run on newer versions of macOS since the application isn’t signed.

Regardless, the login page allows one to configure the network for the robot, select apps to be installed from the Aldebaran store, and adjust the robot’s language, and time zone. Each of these items is in their own tab. There is also a word balloon next to a picture of the robot in the top left corner you can click on and write text to say.

The Robot Settings app (Figure 4) provides similar functionality, but it has additional help and documentation to help people with connecting to the robot.

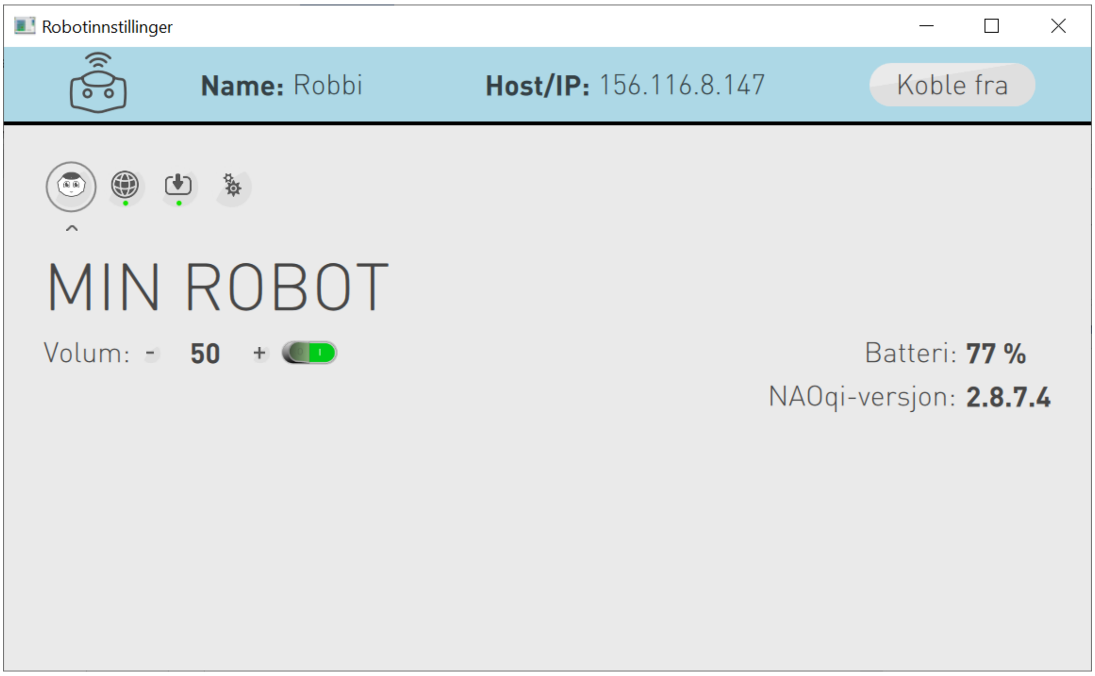


Volume: 50

Battery: 100 %

NAOqi version: 2.8.7.4

Figure 3 The configuration page for NAO with the first tab selected; all buttons are image only and have no text.



Name: Robbi

Host/IP: 156.116.8.147

Volum: 50

Batteri: 77 %

NAOqi-versjon: 2.8.7.4

Figure 4. The Robot Settings app running on Windows, notice that the text is automatically in Norwegian.

4.3 Selecting a sample

We'll use both the remote-control page, the login page, and the robot settings if we can get that running.

4.4 Results from the evaluation

The evaluation is split between each of the different parts.

4.4.1 The remote-control

The remote-control is a web page follows the WCAG guidelines well. Since it consists of buttons and an edit field, this is normally supported well by the different assistive technologies that exist. There are no images, audio, or video that need to be considered. The colors chosen for the remote have contrast that is above the threshold ratio for AAA. The tab ordering for the buttons matches what is on the screen. The edit field has a label. So, the web page itself has no WCAG issues. It is an open question if the placement of the buttons is optimal for working with the remote control, but that has still to be determined in general. It should happen early next year.

Turning to the robot, we will focus most on the perceivable set of guidelines. Starting with Guideline 1, *multiple modalities for interaction* Most of the actions are a combination of voice, audio, and motion from NAO. For some actions, such as the dances, there is an accompaniment of colored lights in the eyes, ears, and feet. Overall, the audio voice feedback from the robot appears to be the most important for many interactions (“hello”, “goodbye”, “correct”, “try again”). In its current iteration, there is no opportunity to have the robot’s speech captioned.

Consideration needs to be made as to how to incorporate Guideline 4, *alternatives for non-text elements*. There is a limitation NAO’s mouth is not movable. This means that it is difficult to know when it is talking. NAO’s motion and gestures also assist in what the robot is saying, but it might be difficult interpret what the robot said by some gestures alone. Some additional use of light in the eyes could be helpful for indicating if the answer was correct or not. Some lights could also be used (for example in the robot ears) when the robot is talking. This can also be useful to check if the volume is correct for the robot.

The “check connection” button uses lights and sound to indicate it’s connection. NAO’s ears will blink blue, and it makes a noise to help indicate if the robot is connected. This means it’s possible to hear if the robot is connected even if one can’t see the robot or the volume is low. If there is no connection, neither of these things happen. Of course, if one cannot see nor hear the robot, it will be difficult to determine if it is connected. In that case, some extra functionality could be added to the web page for the remote control.

On the other hand, all flashing from the interactions is done at safe levels to not cause seizures. This follows Guideline 6 (*Flashing visual content*).

On the other hand, pressing the volume buttons in the remote does not provide any feedback on the robot or in the remote. To be fair this does not happen either when using the physical buttons on the robot (the buttons on the top of the NAO’s head can also change the volume if they have not been overridden by another application running on the robot).

4.4.2 Configuration page

The configuration page is not conformant to the WCAG. You need to login to the robot, and that uses the standard HTTP AUTH prompt, which should be accessible on any browser, but the pages themselves are links with images. None of these images have alternative text. If you turn off image loading on the page, it is very difficult to know what to do. The edit field for entering text to say also lacks a label. In addition, there is very little structure to the page. The page uses a table to lay out items instead of using semantic information and putting the layout in CSS. This makes its usability for a screen reader very poor. The web page is also only available in a couple of languages. The Robot Settings application will follow the language from the operating system.

Although the robot is passive in this situation. Some notes could be made about configuration. When the robot is brought to a new location and it doesn't know about the network connection ahead of time, it is necessary to configure network connectivity. This is normally done by connecting an ethernet cable between the robot and a computer. Unless there is special set up ahead of time, the robot will eventually get a self-assigned IP address. One can find this self-assigned IP address by pressing the NAO button on the robot's chest. The robot will then say the IP address aloud and you can configure your computer to connect to the robot. In addition, if there is a multicast DNS service, you can find the NAO by name (it will be something like nao.local). This set up, at least as it is currently envisioned, requires someone who can see and interpret the buttons and can hear the robot to be able to configure it.

The robot settings application has attempted to make this less difficult as it will do the multicast DNS resolving, so that you can see the NAO robots that are on the same network. This is a Qt application, which should, in theory have some accessibility support built in, but I was unable to test this. On Windows, Robot Settings locked up when the screen reader was active. On macOS, there have been changes in running unsigned applications that cause the app to hang on start up. On Linux, it seems to require a certain version of Linux as well. When I ran it on a Linux system, I could not get Orca screen reader to run. For now, the Robot Settings app can likely be made accessible, but it does not appear to be accessible in its current iteration. It would likely need as much work as the configuration page. Set up of the robot must be done with someone who has knowledge of the network and can see and interpret the images from the configuration page or Robot Settings.

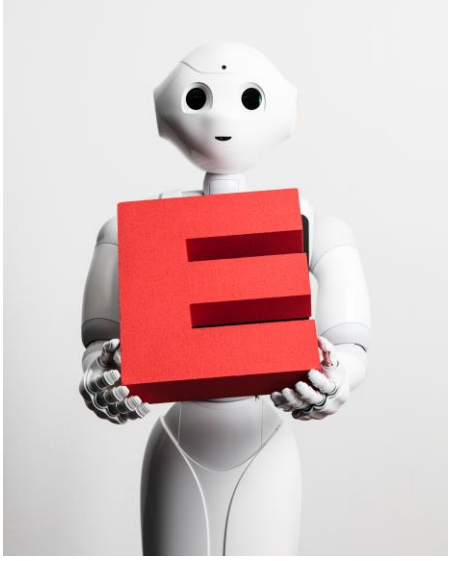


Figure 5: The Pepper robot (Photo Credit TEDx Matingly).

5 Pepper

5.1 Scope

1. Robot name: Pepper (pictured in Figure 5).

2. Scope of the robot: The robot is envisioned to be used as a sort of receptionist

at an office, hotel, or hospital where it is meant to receive guests. The robot can offer directions, call people who the person has an appointment for, or other typical receptionist tasks.

3. If there are screen elements, we'll try and make that work with WCAG 2.1 AA

4. Given that this is a reception robot, we expect that the person interacting with the robot will not necessarily have access to different assistive technology other than what they have with them.

5. We can also look at the setup page to see if it is any different than what is on Nao, we'll see. The primary goal is to find accessibility issues.

5.2 Explore the robot

5.2.1 Information about Pepper

This is a Pepper 1.8 that we were using:

- Tablet: Android Tablet with 1GB RAM and 32GB storage, Running some version of Android, Capacitive touch display. Talkback can be run on the tablet, but it is slow (the tablet in general is slow).

- Microphones: 4

- Speakers: 2

- Video & depth sensors: 2 Normal cameras and either a depth camera or a depth camera and a stereo camera.

- Lights: 40 LED split between the Eyes, ears, and shoulders

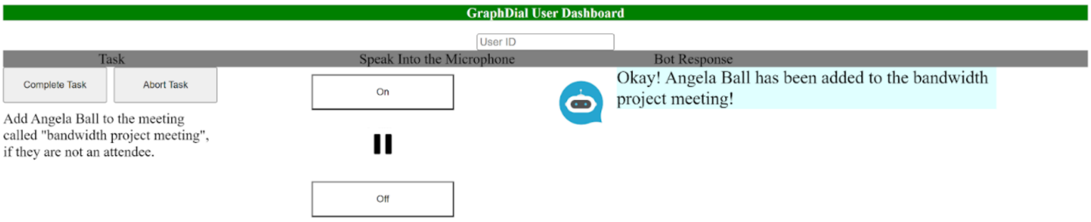
- Degrees of Freedom: 20

- Tactile sensors: 2 push buttons, 3 tactile sensors on the head, 2 sensors on each hand, and a bumper over each wheel.

- Languages: 21 languages with varying levels of support.

5.2.2 The Experiment

The experiment was supposed to be a robot receptionist and a person coming to ask about where a room was or that they were there for a meeting and that the robot would summon the correct person. The actual goal of the experiment was to collect data that could be used for training new models (Walker et al., 2022). The experiment itself used the Wizard of Oz technique, where a facilitator sat in another room and would respond to the queries from the participant in the study. There was a web application (Figure 6) that ran on the web browser on Pepper, where the participant would press the “Record” button to record the query and a “Stop” button to stop the conversation.



Add Angela Ball to the meeting called "bandwidth project meeting", if they are not an attendee.

Okay! Angela Ball has been added to the bandwidth project meeting!

Figure 6: The user interface used for the experiment.

For the context of talking to the robot, there was a web application that had a record button and a stop record button. This was because the experiment used a Wizard of Oz technique (Riek, 2012) by the researcher. In a real-world situation, the person would be asking Pepper these questions without any interaction with the tablet. That is, the person would just be talking to Pepper. So, in this context, we ignore the interface and think more about a pure conversation interface.

The robot is running software on the tablet. The tablet does have the possibility to use TalkBack (Android’s screen reader), and it does work on the tablet. Unfortunately, TalkBack runs slow and only in English. In theory if one can install apps on the tablet, one could get different voices too, but one would also need to investigate what is slowing down TalkBack.

5.3 Selecting a sample

We took the different use cases that were used in the experiment. Asking for directions to a room, or that you were here to meet someone. We assumed that the person would only be talking with the robot directly and not use the “record”, “stop record” buttons.

5.4 Results from evaluation

Most of the interaction is through spoken dialog. This breaks with the *multiple modalities for interaction* guideline. This could possibly be mitigated by adding a keyboard for entering information and echoing the text that Pepper says on the tablet. The keyboard could also be an input on Pepper’s tablet. This may be helpful also in the cases that Pepper cannot hear what the person is saying, for example if there is a lot of background noise or the microphone is not working. Another option would be to add sign language as an option to the interaction. That is, Pepper could understand the signs of the person and sign back.

6 AV1

6.1 Scope

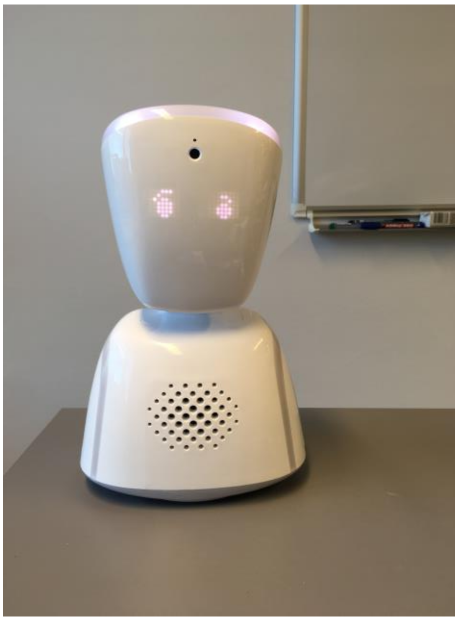


Figure 7: The AV1 with its “confused” look.

1. Robot name: AV1 (pictured in Figure 7).

2. Scope of the robot: The robot is used as a type of telepresence robot, mostly in a classroom environment. The robot is sometimes used in other contexts, but we will primarily focus on the classroom context. There is an administration web page and two mobile apps: one that offers the main telepresence functionality and one that allows most of the administrative tasks from the web page.

3. There are web pages and mobile apps that work with the robot. The apps and webpages will be checked with WCAG 2.1, level AA. For the robot, we will use the guidelines from Al-Qbilat (2022)

4. Baseline. iOS voiceover and Firefox, Edge, Chrome.

5. Primary goal is to find accessibility issues.

6.2 Explore

6.2.1 About AV1

- Degrees of freedom: 2

- Microphone: 1

- Speaker: 1

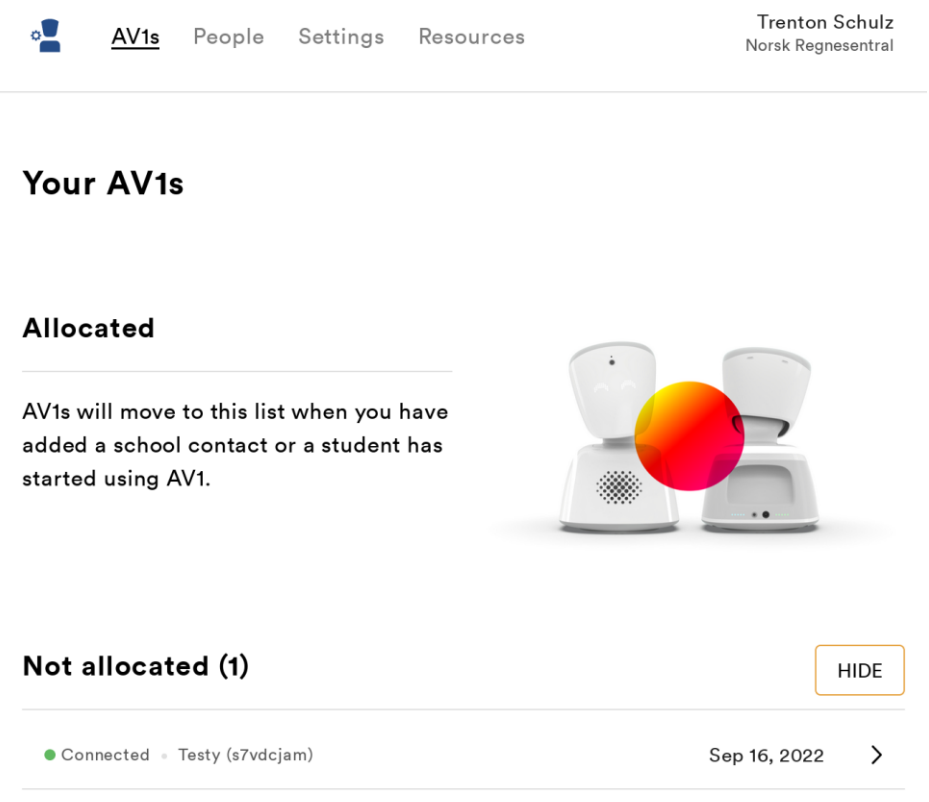
- Camera: 1

- Lights: One color LED on the top of its head, and two eye LED displays that can show simple eye patterns.

6.2.2 AV1 website

There is a website for administering AV1s: https://av1-admin.noisolation.com/. This website (Figure 8) provides a list of robots that are registered to you and their status (if they are on or off). You can add notes about the robot, add school contacts (if it isn’t allocated to a school), or create a new keyword for the robot to register it with a new student. You can also add additional WI-FI networks here as well as update your contact information so people can contact you if the robot is not work. There are also links to the help center if you need more information.

There are also resources for adding the AV1 to a classroom. These are mostly targeted at the child who would be using it and the infrastructure in the school. There is a user guide, a teacher's guide, a lesson plan for introducing the AV1 into the classroom, including slides, a coloring sheet, posters for troubleshooting and features of the robot. A certificate for inclusion in the classroom, a general onboarding guide that collects much of the above information. There are also draft letters that can be used for teachers and parents (for the child and the rest of the class).



Trenton Schulz
Norsk Regnesentral

Your AV1s

Allocated

AV1s will move to this list when you have added a school contact or a student has started using AV1.

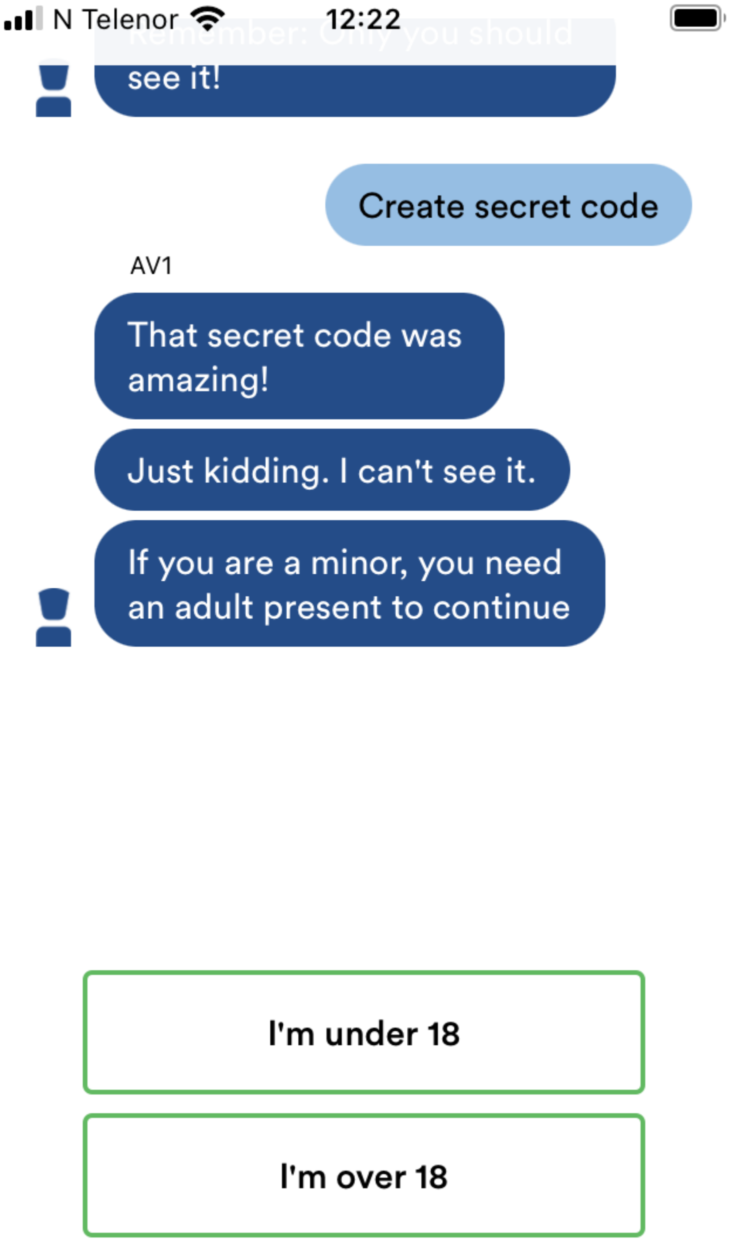
Not allocated (1)

● Connected • Testy (s7vdcjam) Sep 16, 2022 >

Figure 8: The Admin website for the AV1.

6.2.3 AV1 App

The app is straightforward to use. You need to add a robot and then you can connect to it. The app uses a chatbot metaphor with the robot (or app) “talking” to the person through dialog and most options for the use in the form of buttons (Figure 9).



Remember: Only you should see it!

Create secret code

AV1

That secret code was amazing!

Just kidding. I can't see it.

If you are a minor, you need an adult present to continue

Figure 9: Example of the chatbot interface in the AV1 app.

When starting the app the first time, you are asked to add a robot. This involves creating a code and agreeing to the terms and services.

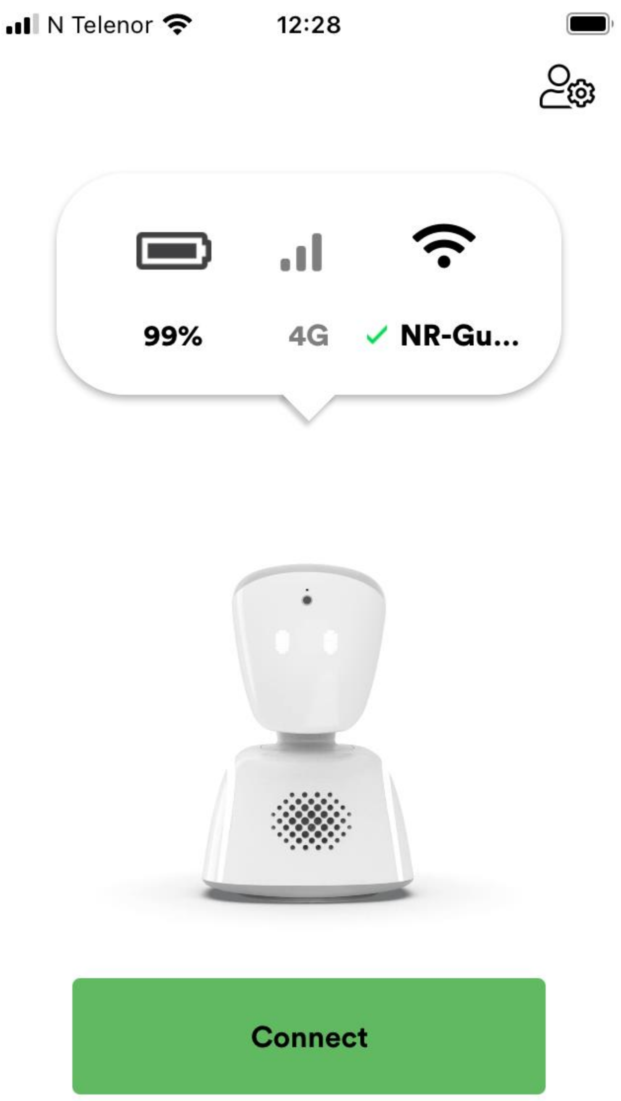


Figure 10: The start screen after an AV1 has been added; the settings button is in the top right corner.

Once the robot is added, you can connect to the robot (Figure 10). This starts the camera, speakers, and microphone on the robot. You can control where the robot looks methods via different control methods.

This leads to the main connection screen for the AV1 app (Figure 11). In the lower right corner, there is an icon of the AV1. Clicking on the icon reveals additional buttons. The buttons allow you to mute and unmute yourself. You can also choose icons to adjust the eyes on the robot to indicate your emotion:

- A laughing icon to indicate amusement.
- A confused icon to indicate wondering.
- A smiling icon as the “default”.

There are two other items: a raised hand icon that will blink the light on top of AV1's head and indicate that you want to be called on and a crescent moon icon that indicates that the child is not currently available.

The settings for the app (Figure 12, left) allows you to connect to different Wi-Fi networks, change your secret code, add assistants that can help with admin tasks (the AV1 Admin app, Section 6.2.4). There are also accessibility options for specifying different ways of controlling the way the robot moves its head (Figure 12, right). By default, you can swipe to move or tap to move the robot head. Swipe moves the AV1's head in a general direction but tapping a specific point on the screen will move the robot to center it in the view.

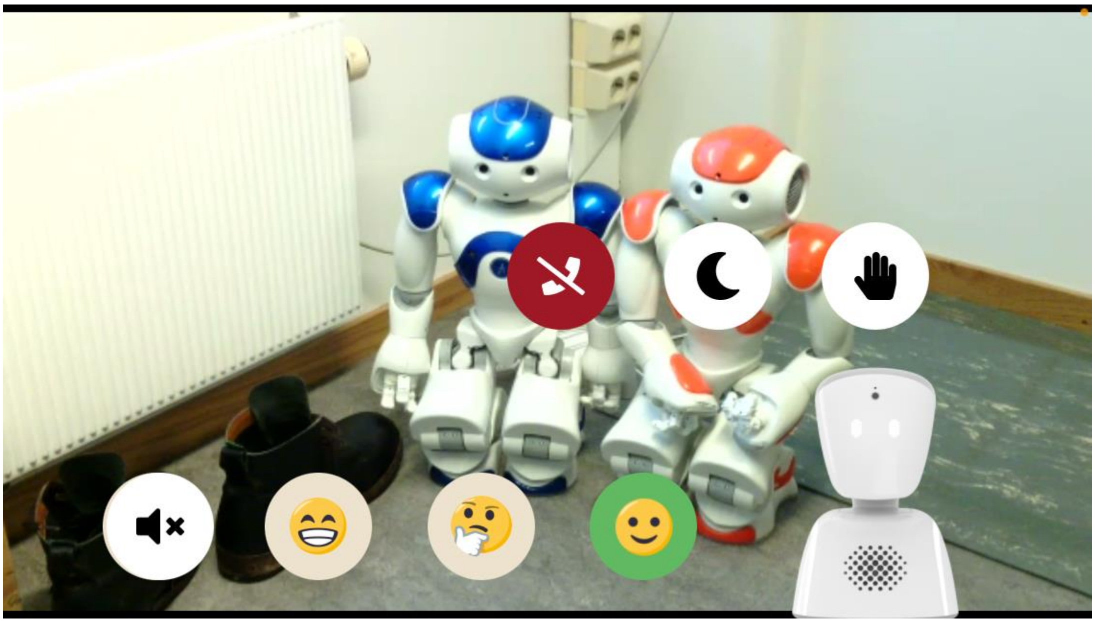
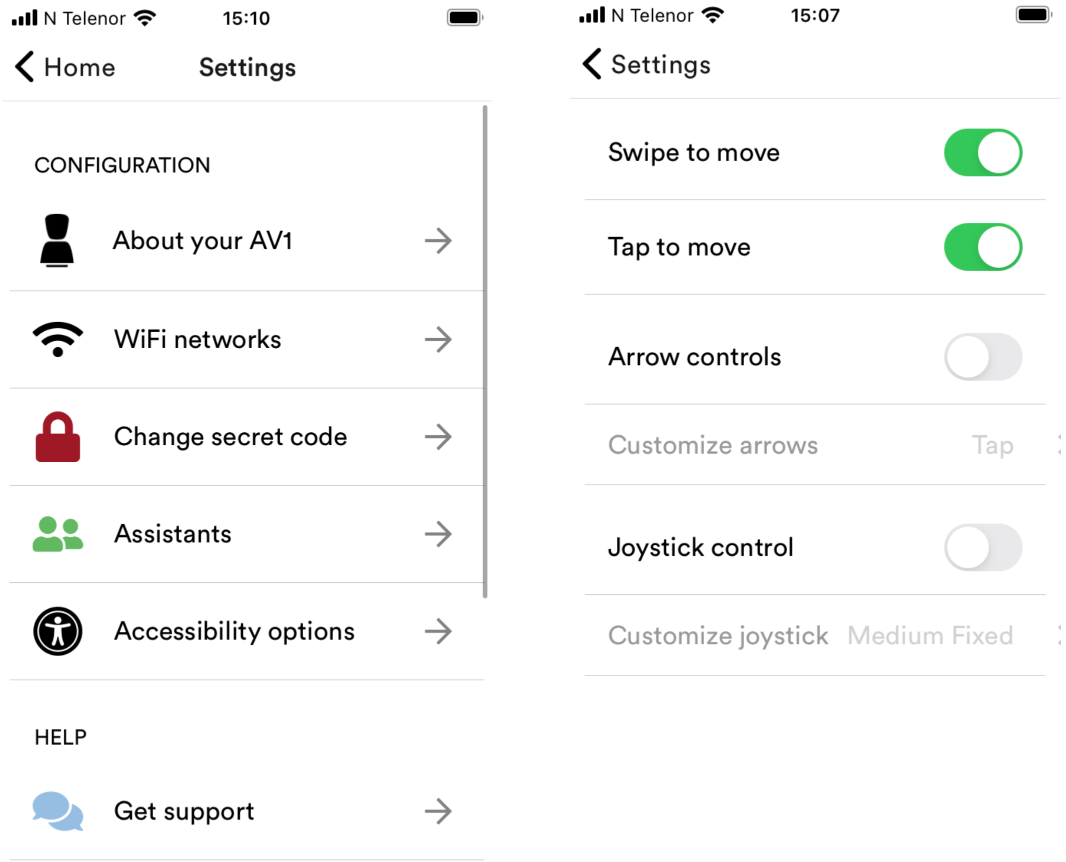


Figure 11: The main connection screen in the AV1 app. Note that pictures in the AV1 app are not allowed.

If these controls don't work well for the person. One can activate either arrow controls or a joystick control. Then, either a group of arrows or a joystick like control is added to the screen, and the controls work like other game controls on mobile phones. You can customize both the arrow keys and joystick to how they work (e.g., do you push and hold the arrow keys or only tap them, does the joystick recenter itself after the finger is loaded).



N Telenor 15:10

‹ Home Settings

CONFIGURATION

- About your AV1 →
- WiFi networks →
- Change secret code →
- Assistants →
- Accessibility options →

HELP

- Get support →

N Telenor 15:07

‹ Settings

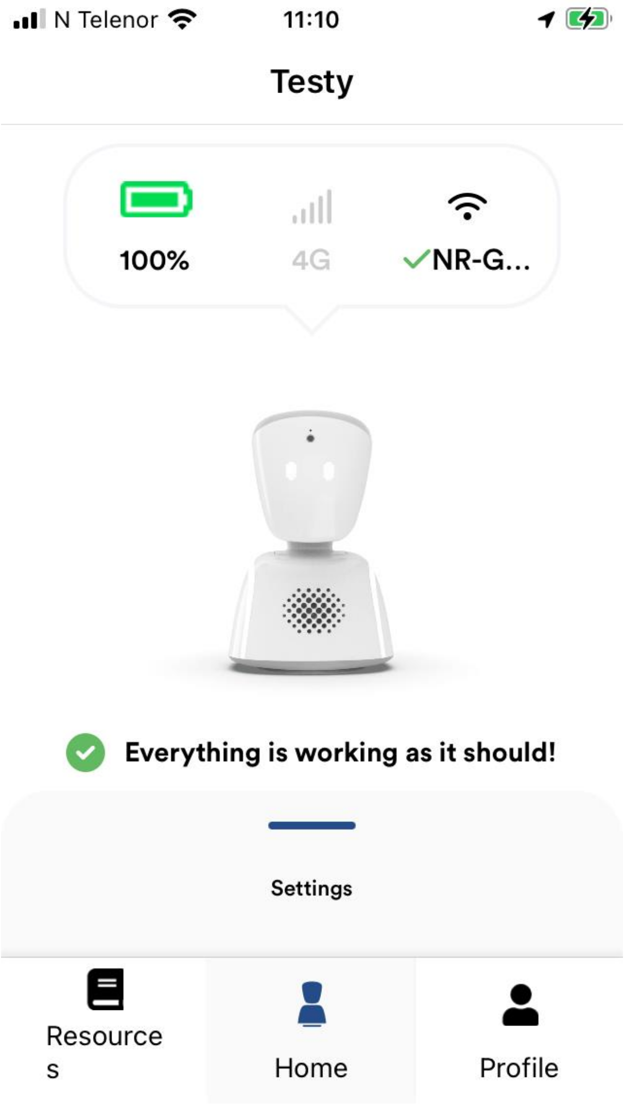
- Swipe to move ●
- Tap to move ●
- Arrow controls ○
- Customize arrows Tap
- Joystick control ○
- Customize joystick Medium Fixed

Figure 12: The settings screen (left) with the accessibility options (right) for controlling where the AV1 looks.

6.2.4 AV1 Assistant App

The Assistant App provides information about one or more AV1s and allows the person to make several changes. From the main screen (Figure 13) provides three tabs. Besides the main screen about the AV1, Resources provides information from the No Isolation web site. Profile provides information about the assistant (e.g., name, email address, phone number).

On the main screen, you need to swipe up to get to the settings. Otherwise, it seems to have much of the same functionality that you would have either on the AV1 Admin web page or in the settings section of the AV1 app. It seems the main reason for this is to let multiple people be able to perform administrative tasks with the robot and not overcomplicate the other interface or require access to the administrative website.



Everything is working as it should!

Figure 13: The AV1 Assistant App is for parents, teachers, or others can change settings instead of relying on an administrator.

6.2.5 The Robot itself

The face on the robot uses lights to indicate different emotions of the person controlling the robot. It has a high-quality camera, microphone, and speaker to allow the person controlling AV1 to interact with other people in the environment. Having both a 4G and Wi-Fi connection allows it to work in a variety of places even when there is no regular internet connection.

6.3 Select a sample

We looked at the admin tasks on the website, the main tasks on the AV1 App (connecting to the robot and using it) and the AV1 Assistant App.

6.4 The review

6.4.1 The robot itself

It is very easy to find the on/off button for the robot. There are pulsing lights to indicate the robot is booting up, but no audible signal. However, one does get notified from the app when the robot is available. Since having the app is required to use the robot, and the person is likely nowhere near the robot during the powering-on process, it makes sense that availability information is presented in the app.

In general, the robot is silent. This could be a potential accessibility issue for people with vision impairment that want to interact with the robot and be notified about changes in the robot facial expression or wanting to be called on. On the other hand, the robot is also meant to be in a classroom and additional noises may be distracting and generate unwanted attention for the person using the robot. Therefore, the visual cues (lights), seem appropriate and easily adaptable to the standard classroom context. All flashing lights are done at safe levels to not cause seizures. This follows Guideline 6 (*Flashing visual content*). If audible ways of getting attention are preferred, that can be accomplished by the person operating the robot (for example, by speaking). Interaction for people with hearing disabilities beyond the included lights on the AV1 would have to be accomplished another way.

6.4.2 AV1 admin Website

Overall, the website follows the WCAG guidelines well. One can navigate the web page well with only a keyboard and running a WCAG checker such as WAVE (WebAIM (Web Accessibility In Mind), 2022) shows only one error (a missing tag). In general, the web pages do not result in any problems if someone were going to work with it.

Many of the learning resources are PDFs and running a sample through Adobe Acrobat's PDF accessibility checker showed that the sampled PDFs had not been generated with any accessibility information. Of course, none of these are required, but it would be nice to have these accessible as well.

6.4.3 AV1 App

The welcome screen is readable by VoiceOver. Provided that you have the keyword a person who could start the registration process with an AV1. This is normally provided by a physical card in the box of the AV1, but you can also read it from the administration website.

After that, the process changes over to the chatbot interface. In these next screens, VoiceOver does not read the text initially. Instead, you just get a button that says "Enter Keyword". You can have VoiceOver start at the top of the screen and read the entire screen. Then, it will read the word bubbles, but this requires the person to constantly check and re-read the screen. So, although the dialog from the chat text is not picked up, the text input field is picked up and properly tagged. This mean that it requires some effort on a person dependent on VoiceOver to know what to do here. If the person presses the button to enter the keyword, there are no prompts in the app, so the person is typing without any feedback. So, it is difficult to know if the keyword is entered correctly.

Although the text bubbles from the chatbot are not read automatically, things such as the emoji are read aloud when they have focus. The first item in the chat is read aloud, and the buttons are correctly identified along with the response, but the person then needs go through the bubbles to follow the conversation and get back to any additional controls.

When one has registered a robot, entering the secret code is straight forward. VoiceOver is able to detect if the AV1 is off or connected, although the connection bubble is a clumsy as it only reads the text. You will get the battery percentage, followed by a network, followed by the wireless network. For example, VoiceOver will read text in Figure 10 as “99% 4G NR-Guest”. This could probably be presented instead as: “Battery 99%, 3 Bars 4G connected, Using Wireless Network NR-Guest with excellent reception”.

The main screen when connected to the robot has some things that work and some things that don’t. It correctly picks up the robot icon as a menu, and the mute and volume buttons work correctly. When you open the menu, however, VoiceOver reads the groups of buttons in two sections: the first with the row of emojis and the second with the hang-up, raise-hand, and rest buttons. Unfortunately, these are only presented as two blocks of items: One cannot choose any of these buttons. This makes it impossible to disconnect while using VoiceOver.

For moving the robots head using VoiceOver, one is best served by using the swipe method (the default) passed straight through to the application. That is, one double-taps and holds until VoiceOver makes a special sound and then you can swipe to move the robot’s head. Using the arrow keys method adds another layer of buttons that cannot be touched, and the joystick is not visible to VoiceOver either.

Although one cannot get to the hang up screen with VoiceOver, the current hang up screen also needs some work. The buttons are not recognized as buttons and closing the connection requires you to tap and hold the disconnect button. The tap and hold can be done via passing through VoiceOver, but it does require you to know where the button is.

The icon for the Settings page does not have a label, so one must tap the icon and hope that settings open. The Settings page has many additional, noisy labels. For example, the heading is read repeating the first word, that is “Object Object Settings”. Unfortunately, not all settings are accessible. For example, the screen for setting up a Wi-Fi connection is not readable by VoiceOver, all the fields for Wi-Fi are stuck in a rectangle that is not accessible. Also, the option to remove the robot (for example, if you are getting a new one) has some of the app text not read aloud.

6.4.4 AV1 Assistant App

The AV1 Assistant App shares some of the issues from above: it also reads only the text from the Robot’s word bubble, so you get the similar “100% 4G NR-Guest” read aloud. Similarly, its Wi-Fi dialog is also inaccessible to VoiceOver.

The biggest issue is that the settings dialog is accessed by swiping up on the main screen. This includes some information such as network strength and AV1 serial as well as options for changing the robot’s name, Wi-Fi connections, and more. Unfortunately, VoiceOver can “see” the items when navigating on the screen, but it cannot activate them. It also prioritizes them over the toolbar at the bottom of the page. This can be confusing if you are only using VoiceOver. If one knows one needs to swipe up, one can pass a swipe through and then the commands work.

The resources screen is accessible by VoiceOver, and it provides links to a variety of web pages on No Isolation’s website. These web pages, like others on No Isolation’s website, appear to have good compliance with WCAG.

7 Berntsen

7.1 Scope:

1. Robot name: Berntsen (pictured in Figure 14), which is a repurposed Double Robotics telepresence robot (Double 2 or Double 3) with additional software installed on it.

2. Scope of the robot: Berntsen is used as a platform for encouraging older people to be active. Most of the interaction should be automatic, but there are support functions done by others through the Patient Sky website. We will not look at PatientSky as that is its own complete system. The additional software provides a schedule for the person and an opportunity for a health worker or trainer to contact and talk with the person via the telepresence interface. There is also the possibility of calling others people. This, unfortunately, was not on the robot that we had access too. So, we will focus mostly on the telepresence interface.

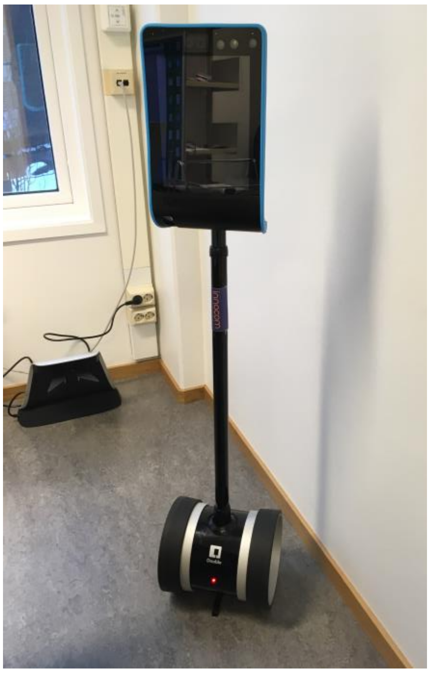


Figure 14: Double 3 Robot, which runs the Berntsen software.

3. We’ll use the HRI guidelines mostly, since it has a screen, we can examine it as well. We can also look at the telepresence interface for the controlling the robot, this is done on a web page or via an app on the phone or iPad.

4. VoiceOver on the iPad.

5. Primary goal is to find accessibility issues.

7.2 Explore the robot

7.2.1 About the robot

The Double 3 has the following features (taken from Double Robotics Website):

2 × 13 Megapixel Unified Pan/Tilt/Zoom Module

- One super wide-angle lens, one super zoom lens

- 30 FPS and Night Vision Mode

2 × Stereovision depth sensors (Intel® RealSense™ D430)

5 × Ultrasonic range finders

2 × Wheel encoders (2048 PPR each)

1 × Inertial Measurement Unit (9 DoF)

6 × Digital microphones with beamforming

8-watt full range speaker

9.7-inch LED-backlit multi-touch LCD

Wi-Fi - Intel Dual-Band Wireless-AC (2.4GHz, 5GHz)

Bluetooth 4.2

Remotely-adjustable height (47 to 60-inches tall)

Wi-Fi and Bluetooth connectivity

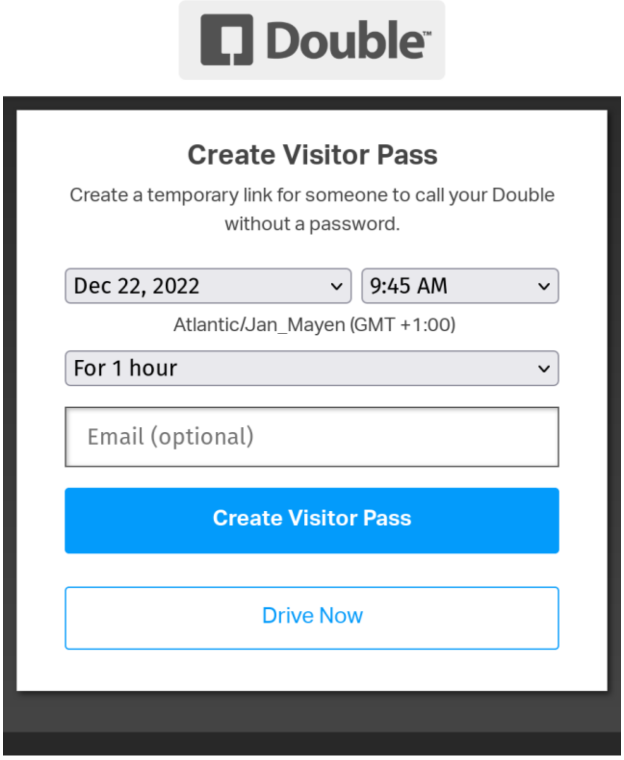
There is also an API available for adding your own functionality. This is where the Berntsen software would be added.

7.2.2 The robot software

The additional Berntsen software is not installed on the robot, but there is an alternate version available on the iPad. This is closer to an evaluation of the stock software that comes with the robot.

The robot has a display that is activated by being close to the robot. The display also is a touch screen and has buttons for creating a visitor pass, setting up an Internet connection, checking the battery level, and parking or unparking the robot. If you swipe up or down on the left or right side of the display, you can adjust the volume of the speaker or the robot’s height respectively.

The visitor pass displays a QR code that can be scanned by a phone in the area. This opens to a web page that offers you the ability to drive immediately or to send an invitation to an email address that can be used for a specific time slot (Figure 15).



Double

Create Visitor Pass

Create a temporary link for someone to call your Double without a password.

Dec 22, 2022 | 9:45 AM

Atlantic/Jan_Mayen (GMT +1:00)

For 1 hour

Email (optional)

Figure 15: Visitor Pass form for Double 3/Berntsen.

If you either choose to drive immediately or come into to the invitation at the proper time, you will get an HTML interface in your web browser that can be used to drive the robot (Figure 16). If you are a registered user of the robot, you can also control the robot via an iOS device. Either way, the interface works by choosing a spot on the floor that you want the robot to navigate to and it will navigate to the location as best it can. You can also adjust the height of the robot, go with a picture in picture set up, share a web page, or other things.

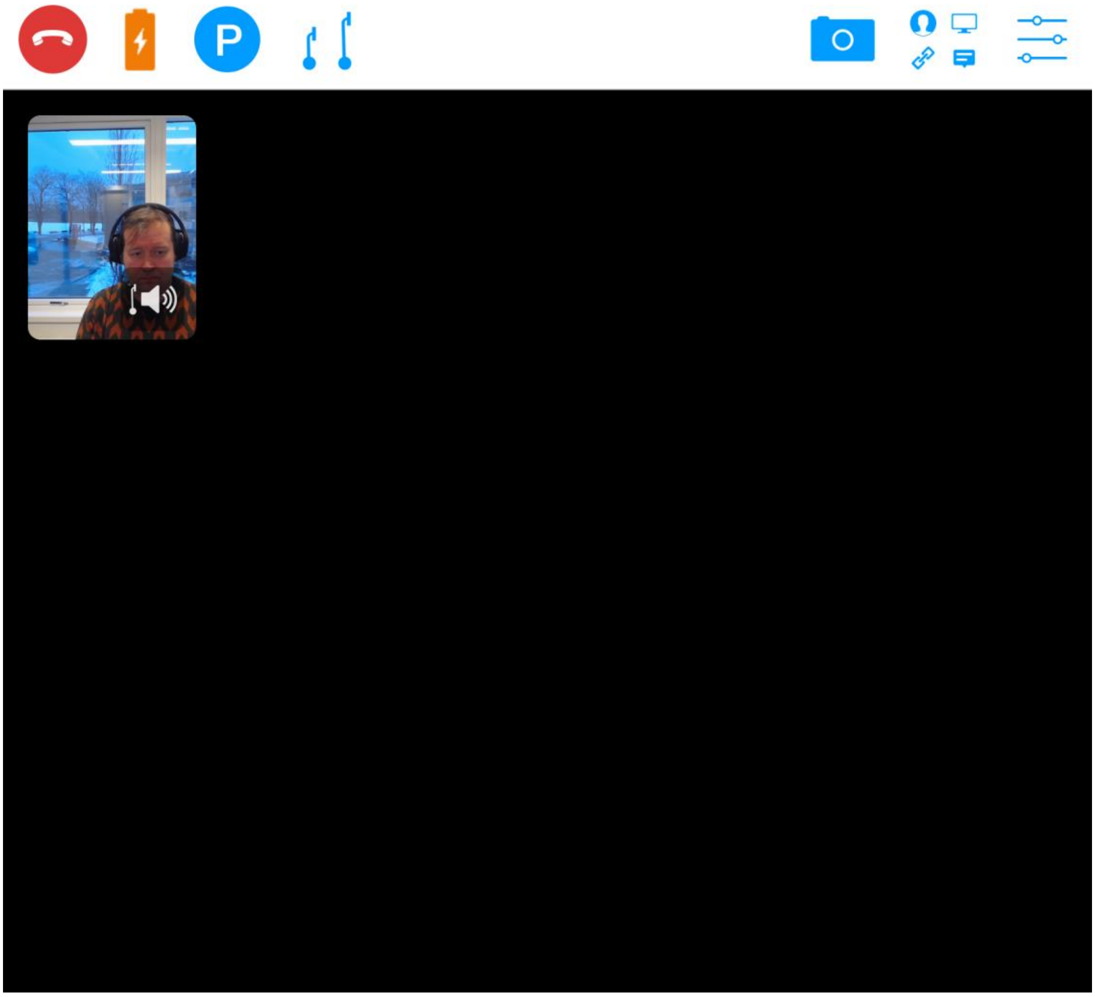


Figure 16: An example of the controller interface for the Double 3. The large black area is what is shown by the camera from the robot.

7.3 Select a sample

We will focus mostly on the invitation and control functionality since that is the software that we had access to.

7.4 The review

In general, if one is interacting with the person controlling the robot, it is straightforward assuming the volume levels are set OK and the light levels are OK for tablet (i.e., direct sunlight will wash out the strongest displays, but should not be a problem in an indoor environment). The person controlling the robot can also send text on the display or show a webpage, so it is possible to provide alternate communication to the person interacting with the robot. The controls for working the height and the volume can be manipulated without seeing the screen, assuming one knows the controls are there. Adjusting the height provides direct feedback, but there is no feedback for the volume if no audio is playing. The controller can also adjust these parameters, so it should not be a big issue.

Getting a visitor pass requires scanning the lower left side of the display. The code stays for a while, so it is possible to scan and open the link from it. The web page that you get to from scanning the QR code is simple but does not follow the form controls do not have labels and the two buttons at the bottom of the page have poor contrast. Using VoiceOver on the page is possible, but you need to have an idea of what needs to be done on the page.

The interface for the controller is, alas, not accessible. The contrast issues from the invitation page are also here for the buttons. Most of the buttons are also icon only with no text. The interface also requires you to visually place the position on the map that you want the robot to travel to. There is one advantage to this that the person doesn't need to "drive" the robot, but it requires an accessible interface for picking the spot to travel.

So, while people interacting with the robot can have the communication handled in multiple channels, it appears that the controller still needs sufficient eyesight and dexterity to control the robot.

8 Tiago

8.1 Scope

1. Robot name: TIAGo (Figure 17), which is a robot research platform (as opposed to an off-the-shelf product).

2. Scope of the robot: TIAGo is used as a research platform. The platform is sold by Pal Robotics (with HQ in Barcelona, Spain). The target audience are universities or research institutions that aim to develop, test, and conduct studies with the robot. The robot was sold in over 200 exemplars at different universities or other research institutions around the world. The robot can come equipped with one- or two robotic arms, and different design of the robotic grippers. A screen/tablet can also be bought or attached to the robot if desired. The robot is used for research in Artificial Intelligence and Machine Learning, Human-Robot Interaction, Manipulation, Perception, Navigation, Motion Planning. The fields of application cover: (future) factories, smart cities and IoT, Ambient Assistive Living (AAL), Benchmarking, User Studies.

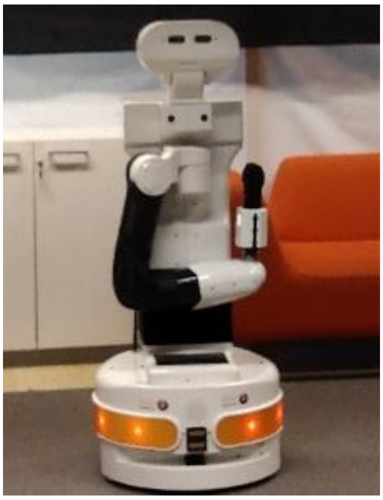


Figure 17: The TIAGo robot (Photo credit: Diana Saplacan).

3. Other information about the robot: The robot is often used in EU-projects as a research platform. In Norway, to our knowledge, there are three Institutions owning a such robot: University of Oslo (also a partner in the UD-Robots research project), Institute for Energy Technology (IFE), and University of Agder.

4. Design of the robot: the robot is formed from a mobile base, a torso, a head, and one or two robotic arms.

8.2 Explore

- Tablet: can be mounted on the robot (not available by default)
- Microphones: stereo microphone
- Speakers: yes
- Video & depth sensors: RGB and depth camera (thermal camera – mono and colored, can be bought and mounted on top of it)
- Lights: on the base (showing that the robot is ON)
- Degree of Freedom for the robotic arm: 7df
- End effector: parallel gripper (other possibilities are available to buy, or to 3D-print after a predefine design)
- Navigation: laser
- Battery autonomy: 4-5hs with 1 battery and 8-10 hs with 2 batteries
- Languages: available in several languages.
- SSD: 250 GB
- CPU: Intel i5
- Modalities of interaction: direct interaction, speech, motion
- Operating System: Ubuntu LTS, Real time OS (ROS)
- Simulation platform: Gazebo
- Height: 110-145 cm
- Arm payload: maximum 3 kg
- Mounting points: on the head (e.g., thermal camera), on the laptop tray, and on the mobile base
- Weight: Between 50-100 kg.

8.2.1 Experiments

TIAGo was used in experiments, on behalf of two other ongoing research projects, with a group of older users at an activity center. Some of the older users were using walking sticks, others were using wheelchairs, some walkers, and some did not have any additional assistive technologies. The experiments TIAGo was used are described below:

The first experiment had researchers show different types of cameras that TIAGo can use to detect persons or objects. The cameras included in the experiment were: RGB camera, thermal camera (mono-color and colored), and depth camera. The participants were asked a set of questions related to their perceived privacy when the robot uses different types of these cameras. In addition, other elements, such as handover of an object to the participants were also included. Besides these interactions, TIAGo was also interacting with the users through speech. Moreover, the participants were asked questions about their perceived safety. In this experiment, the researchers were controlling the robot through Wizard of Oz (Riek, 2012).

The second experiment took place together with the older users at the same activity center. The experiment was based on TIAGo approaching the participants at different heights and speeds, both in the physical environment, and in Augmented Reality (AR) by using a MetaQuest Virtual Reality System. The users could stop the robot at any time, by pressing a button on a controller. In this experiment, there was no speech interaction included. The speed and height of TIAGo was set manually, but the robot executed the task semi-autonomously (being pre-programmed).

8.3 Select a sample

We used the activities from the experiments above as a basis for looking at accessibility issues.

8.4 Report of results

A robot using a camera is not necessarily an accessibility problem as it can be part of its sensor system for navigation or other tasks; that is, the camera data is processed for its sensor purpose and then discarded. It can be an issue if the robot wants to preserve privacy and perhaps provide alerts to people that it is now recording something that will be reviewed by others.

The participants in the experiments had different levels of mobility, although the mobility of each participant wasn’t directly related to the activities in the experiment, it probably played some part on their opinions in the second experiment with the robot approaching at different heights and speeds.

AR provides interesting possibilities for providing information in the environment, but it also necessary to examine universal design issues with that as well. It has similar issues with Virtual Reality (Simon-Liedtke & Baraas, 2022) and is beyond the scope of this evaluation.

In general, TIAGo is designed to be programmed by researchers, programmers, or roboticists to perform a task, and possibly also used in user-studies. It is difficult for someone that does not have any technical knowledge to start using it. However, TIAGo can be controlled through a remote control (for example, a PlayStation-style controller), but any controller that is compatible with a PlayStation controller (or joystick) protocol can work. So, alternative switches could be used as well if the participant is meant to control the robot. Otherwise, it is up to the people planning, creating, and running the user study to ensure that accessibility issues are addressed in the study.

9 Closing remarks

The purpose of this document was to try and use the method suggested by Al-Qbilat (2022) that we have given the name ASAR-EM. Overall, the method proves good as an approach for examining accessibility issues with socially assistive robots. It helps to capture the context that the robot will be used and provides an opportunity to understand the robot's capabilities and the software that will be used. This can provide insights into other things that need to be considered (for example, considerations for classrooms in the case of the AV1 and the NAO or the reception area for Pepper). This does offer a good opportunity for creating a truly universally designed technology and service.

A small bonus for evaluating a robot in this way is that the results can provide a good starting point for considering the robot in other contexts. Of course, contexts are different, but the basic abilities of the robots do not change much, so a previous evaluation may provide a solid foundation for flagging potential issues or provide new items to consider for the new context. This makes the results more re-usable than compared to a typical WCAG-EM.

Finally, one theme that seemed to come up was difficulty in setting up the robot. While a robot for a certain context could be made accessible, much of the process for setting up the robot is not accessible and requires specialized knowledge. Granted, there is much that needs to be understood for each robot to make it work and the setting up most robots is currently targeted at technical users or admin users. Yet some effort could be done to make this more accessible so that less technical people are not dependent on others to have their robot set up. After all, it is possible to install operating systems with the use of assistive technology. So, one can hope that this may someday be possible with future robots.

References

Abou-Zahra, S., de Vries, H., & Hansma, M. (2022, August 8). *WCAG-EM Report Tool*. https://www.w3.org/WAI/eval/report-tool

Al-Qbilat, M. M. I. (2022). *Accessibility requirements for human-robot interaction for socially assistive robots* [Ph.D., Universidad Carlos III de Madrid. Departamento de Informática]. https://e-archivo.uc3m.es/handle/10016/35142

Riek, L. D. (2012). Wizard of Oz Studies in HRI: A Systematic Review and New Reporting Guidelines. *Journal of Human-Robot Interaction*, *1*(1), 119–136. https://doi.org/10.5898/JHRI.1.1.Riek

Schulz, T. (2014). Creating Universal Designed and Trustworthy Objects for the Internet of Things. In P. Zaphiris & A. Ioannou (Eds.), *Learning and Collaboration Technologies. Technology-Rich Environments for Learning and Collaboration* (pp. 206–214). Springer International Publishing. http://link.springer.com/chapter/10.1007/978-3-319-07485-6_21

Schulz, T., & Skeide Fuglerud, K. (2022). Creating a Robot-Supported Education Solution for Children with Autism Spectrum Disorder. In K. Miesenberger, G. Kouroupetroglou, K. Mavrou, R. Manduchi, M. Covarrubias Rodriguez, & P.

Penáz (Eds.), *Computers Helping People with Special Needs* (pp. 211–218). Springer International Publishing. https://doi.org/10.1007/978-3-031-08645-8_25

Simon-Liedtke, J. T., & Baraas, R. (2022). Towards eXtended Universal Design. In I. Garofolo, G. Bencini, & A. Arenghi (Eds.), *Studies in Health Technology and Informatics*. IOS Press. https://doi.org/10.3233/SHTI220865

W3C Working Group. (2014). *Website Accessibility Conformance Evaluation Methodology (WCAG-EM) 1.0* (E. Velleman & S. Abou-Zahra, Eds.). World Wide Web Consortium. http://www.w3.org/TR/WCAG-EM/

Walker, N. T., Ultes, S., & Lison, P. (2022). *GraphWOZ: Dialogue Management with Conversational Knowledge Graphs* (arXiv:2211.12852). arXiv. http://arxiv.org/abs/2211.12852

WebAIM (Web Accessibility In Mind). (2022). *WAVE Web Accessibility Evaluation Tools*. https://wave.webaim.org/