Department of Computer, Control and Management Engineering

Pepper as a Sign Language Tutor

Elective in AI - HRI Report

Professor:
Luca Iocchi

Students:
Amine Ahardane (2050689)
Ilaria Carifi (1999362)
Elisa Scandiuzzi (2069444)

Academic Year 2023/2024

Contents

1 Introduction **3**
- 1.1 Context and motivation . . . 3
- 1.2 Objectives . . . 3
- 1.3 Summary of the results . . . 4

2 Related work **4**

3 Solution **5**
- 3.1 Functional Architecture Components . . . 6
  - 3.1.1 Social Signal Perception Module . . . 6
  - 3.1.2 Memory . . . 7
  - 3.1.3 Mental Model . . . 7
  - 3.1.4 Knowledge Base . . . 7
  - 3.1.5 Social Reasoning Module . . . 7
  - 3.1.6 Social Signal Generation Module . . . 7

4 Implementation and Results **8**
- 4.1 Setting Up the Pepper Environment . . . 9
  - 4.1.1 Step 1: Installing Docker . . . 9
  - 4.1.2 Step 2: Downloading Packages . . . 9
  - 4.1.3 Step 3: Configuring the Host OS . . . 10
  - 4.1.4 Step 4: Starting the Emotion Server . . . 10
  - 4.1.5 Step 5: Running the Application . . . 10
- 4.2 System Components . . . 11
- 4.3 Module Implementations . . . 12
  - 4.3.1 Emotion Detection Module . . . 12
  - 4.3.2 Emotion Server . . . 12
  - 4.3.3 Robot Control Module . . . 13
  - 4.3.4 Engagement Zone Module . . . 14
- 4.4 Data Structures and Knowledge Representation . . . 15
- 4.5 User Mental Model . . . 15
- 4.6 Tools and Libraries . . . 16
- 4.7 Development Efforts . . . 16
- 4.8 Challenges Encountered . . . 16
- 4.9 Testing and Evaluation . . . 17

5 Results **17**
- 5.1 Intent Recognition and Interaction Initiation . . . 17
- 5.2 Positive Reinforcement on Correct Answers . . . 18
- 5.3 Assistance on Incorrect Answers . . . 18
- 5.4 Emotional Support and Adaptation . . . 18
- 5.5 Achievement of Objectives . . . 18

6 Experimental evaluation **19**

7 Conclusion **21**

All students have equally contributed to the project.

1 Introduction

With the development of robotics, interaction between humans and humanoid robots will become increasingly frequent, involving also users without specific training in the use of these robots. In this context, it is crucial that robots be equipped with social intelligence, enabling them to detect and express emotions, and also to act in an adaptive way by following social norms.

This project focuses on the analysis and development of a robotic application focusing on Human-Robot Interaction (HRI). The goal of the application is to support users in learning American Sign Language (ASL). In this project, the robot Pepper assumes the role of a teaching tutor, assisting users in acquiring the ASL alphabet through an interactive quiz displayed on the screen of the incorporated tablet. In addition to providing learning support, Pepper acts as an empathetic assistant, able to reassure and motivate the user at times when he or she appears sad or frustrated during the learning process. In addition, the robot builds a mental model of the user's acquired knowledge, analyzing the answers provided in order to identify learnt signs and less established ones, in a similar way to what a teacher would do. In this way Pepper is able to personalize its teaching, focusing on letters that require further reinforcement, improving learning effectiveness and making interaction more engaging and adaptive.

1.1 Context and motivation

The target audience of the application we developed are children who are able to read. However, there are no actual age limitations; in fact it is a learning game that can captivate everyone's attention.

We envision the robot positioned in waiting areas highly frequented by children (for example, in pediatrician's waiting room, vaccination centers, dental clinics, etc.) and with this in mind, we placed particular emphasis on providing entertainment. In order to make the educational game more engaging and to facilitate learning, we followed the principles proposed by Malone, which will be further elaborated in Chapter 2. While originally developed for computer games, these principles have been combined with additional techniques specifically suited to social robots.

The benefits of the developed application are twofold: first, it aims to entertain and distract children in the waiting room, helping to reduce their anxiety. Second, it serves as an educational tool to encourage the learning of sign language, with the goal of promoting greater integration of deaf individuals into society.

1.2 Objectives

The objectives of the project are mainly three:

1. To develop a Human-Robot interaction task where Pepper shows evident social abilities, being able to acknowledge human's mood and feelings and to react accordingly, providing adaptive emotional behavior to make the interlocutor feel understood and at ease.

2. To provide a sufficiently engaging interaction in a way that the kid can be distracted from a situation that may lead to anxiousness or alternatively boredom.

3. Last but not least to familiarize as many kids as possible with the basics of American Sign Language (ASL) alphabet.This objective stems from the desire to promote the integration of the deaf community by encouraging hearing individuals to learn ASL, thereby fostering communication between hearing and non-hearing individuals.

1.3 Summary of the results

In this project we successfully implemented an adaptive learning system for Pepper, where the robot uses real-time emotion detection to adjust its tone and pace based on user mood; for example, through the use of encouraging phrases or breaks as soon as it detects frustration.
In addition to this, we have reason to believe that Pepper's emotional behavior—such as being sad when the user is sad or angry, or offering comfort in the case of defeat—conveys empathy to the user, allowing to develop a sense of camaraderie with the robot. This undoubtedly enhances performance in the game and contributes to the goal of distracting from the surrounding situation. We managed, through the use of a face recognition module and of memory, and thanks to the possibility given to the interlocutor to choose every time how to proceed, to create a customized learning game to the specific user. An example is the possibility of selecting the level of difficulty or the increased frequency of appearance of the letters that the user gets wrong.

2 Related work

Examples of robotic assistants for learning sign language can already be found in the literature [1], [2]. However, our implementation is distinguished by a greater focus on social intelligence, with an emphasis on the robot's ability to reassure the child in situations of frustration and to adapt teaching based on the user's acquired knowledge.

Since the application's target audience consists of children, we considered it appropriate to place the interaction in a game context. The literature, indeed, indicates that educational games can be highly effective tools for promoting learning [3], [4].

To make the game more engaging, we applied principles outlined by Malone in [5], by setting the game's challenge within a fantasy context where an objective must be reached. The choice of side story was conducted considering

that educational activities with robots are most effective if the robot is seen as a peer [6]. The embodiment of the robot Pepper fits particularly well with such a choice being of comparable height to a child (1.20 m). Based on this, it was decided to call the robot the protagonist of the fictional story that frames the game that needs the child's help to proceed. Following the approach outlined in [7], the narrative includes an additional character, Marco, who is represented on the tablet. This virtual assistant employs emotional mirroring, displaying emotions that correspond to those of the user—for example, expressing sadness when the user loses and happiness when they win. This strategy was implemented not only to encourage empathy but also to overcome the limitations imposed by the Pepper's reduced facial expressiveness, enhancing the overall effectiveness of emotional communication. Following also the principles defined in [5] different levels of difficulty were defined, and the level proposed to the player varies according to the social cues detected by the robot and the mental representation [8] of the player that the robot reconstructs by analysing the rate of correct responses.

Our setting calls for the need of an enhanced human-robot interaction that exploits emotional and behavioural adaptations. Among the several studies on the matter, we were particularly interested in the one where Rossi et al. (2020) investigated how social assistive robots (SARs) could alleviate anxiety in pediatric healthcare by employing distraction strategies paired with emotional behaviors tailored to children's anxiety levels [9]. Their findings demonstrated that emotional adaptivity, such as switching between positive and negative valence behaviours, was effective in sustaining children's attention and reducing distress, particularly in challenging contexts like vaccinations. In fact in the study of [10] highlights how emotional and memory-based adaptations play a crucial role in maintaining user engagement over time. Our Pepper robot adopts similar emotionally responsive behaviours to address the emotional states of child interlocutors, improving engagement and reduce distress. Communication takes place through the use of multi-modal interactions [11], each interaction was conducted across multiple channels to improve robustness to failures and provide a better user experience. We also took into account proxemics, as a key factor in understanding the user's willingness to engage. By analyzing spatial proximity, it is possible to infer the level of interest or comfort the user feels, which is a useful information to understand their intent and adapt the interaction further [12].

3 Solution

In this section we will describe the functional architecture of our system, with emphasis on its components and how they interact with each other with the common purpose of allowing a smooth and engaging human-robot learning experience.

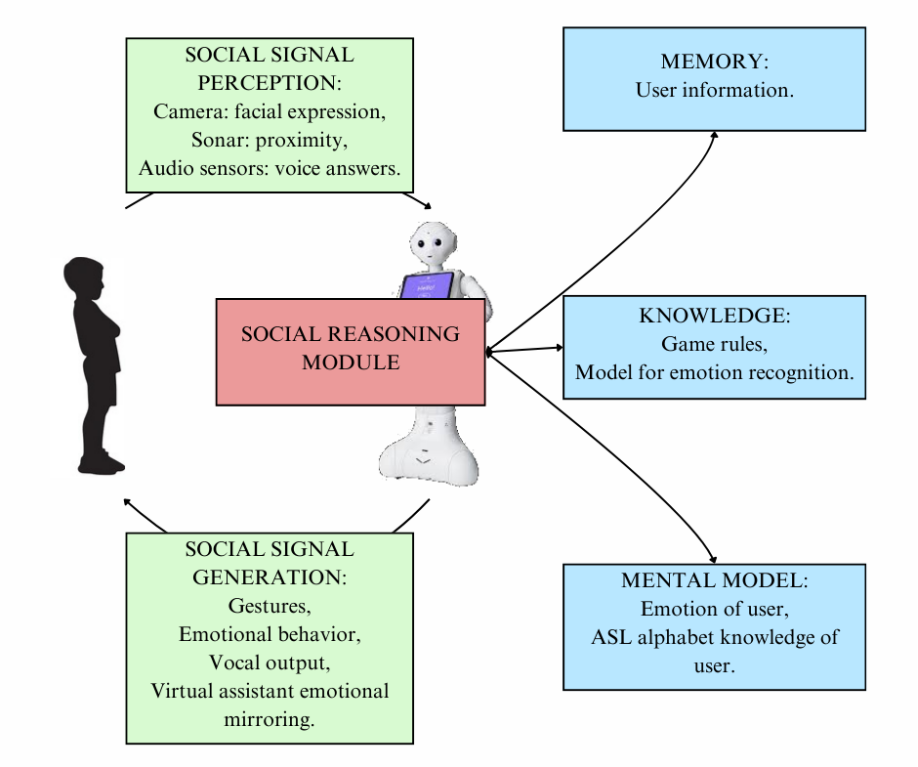

Figure 1: High-level Functional Architecture Overview. The diagram shows the interaction between components.

3.1 Functional Architecture Components

The core components of our system are described below and are schematized in Figure 1.

3.1.1 Social Signal Perception Module

This module enables the robot to perceive social signals from the human through audio sensors, a sonar and a camera.

Audio sensors allow for multimodal interaction, functioning as a support to the communication via tablet. The robot is able to understand the child's answers and to sense the silence. Through the use of a sonar Pepper monitors user proximity, categorizing distances into engagement zones (Zone 1, Zone 2, Zone 3) to determine the user's level of interaction. The camera plays an important role, as it allows the robot to detect emotions such as happiness, frustration, or confusion.

3.1.2 Memory

The memory module is used to store informations about the user.

- **Short-term memory** tracks responses within a session, enabling real-time adaptation.
- **Long-term memory** allows Pepper to personalize future interactions based on previous experiences. The robot will remember, for example, whether the interlocutor has already played or not and can adjust accordingly, either skipping the introduction or offering a tutorial if needed.

3.1.3 Mental Model

The system maintains a mental model of the user, storing information about their past interactions, emotional states, and learning progress with the ASL.

3.1.4 Knowledge Base

Contains predefined sign language vocabulary and the rules of the interactive game, with strategies to adapt the lesson based on user performance. In addition, the robot has the notions necessary for emotion recognition.

3.1.5 Social Reasoning Module

This module processes input from the social signal perception system in conjunction with the knowledge, memory and mental model modules, to assess the user's emotional state and engagement level along with their knowledge of the ASL. It is responsible for decision-making as it selects appropriate robot actions by filtering the input information through some predefined conditions.

Our system maintains a personalized difficulty progression based on past mistakes and successes. Depending on user performance, Pepper dynamically selects quiz questions, focusing on letters the user struggles with the most. In the meantime, thanks to the emotion recognition module, Pepper can, for instance, provide encouragement or offer breaks when frustration or disengagement is detected.

3.1.6 Social Signal Generation Module

This module is responsible for improving the interaction by allowing Pepper to communicate in a natural and engaging, indeed human-like, manner. This is managed through a combination of vocal output, gestures, and visual displays that have been exploited with the common purpose of achieving emotional behaviour.

The robot utilizes body movements (e.g. head tilts, nods, arm movements) to emphasize spoken messages and convey emotions. While the built-in tablet provides additional visual aids, both to storytelling and to the engagement of

the interlocutor. This is done by displaying pictures of Marco in different emotional states as the story is narrated and showing his reactions to the user's learning progress, thus providing further reinforcement and increasing empathy and motivation.

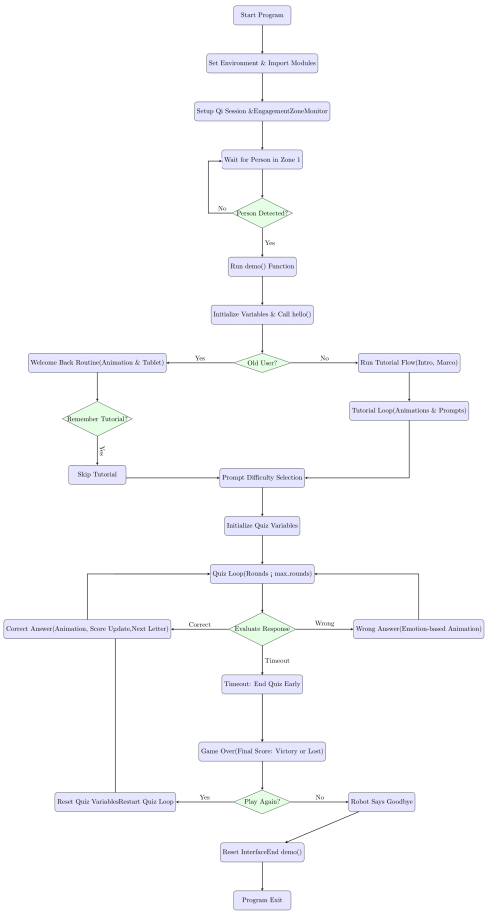

Figure 2: Overview of the Program's General Operation.

4 Implementation and Results

This section explains how specific modules in our functional architecture have been developed. Following a brief setup of the environment, we proceed by describing the data structures used, the libraries and tools that power each module, and which parts were written specifically for this project. Where helpful, we also include brief code excerpts.

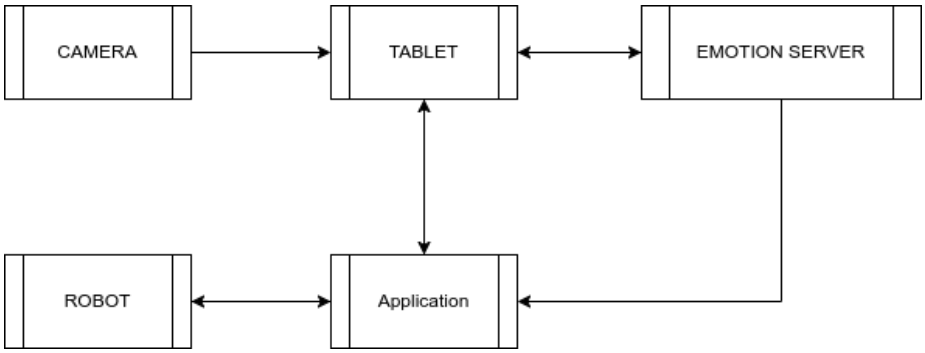

Figure 3: Overall Architecture Overview.

4.1 Setting Up the Pepper Environment

Our development and testing took place on Ubuntu 22.04 using the Choregraphe simulator. The steps below detail how to install Docker, pull the necessary repositories, build images, and configure the environment so you can replicate the setup.

4.1.1 Step 1: Installing Docker

Use the following commands to install Docker and add your user to the Docker group:

```
sudo apt install docker.io
sudo usermod -aG docker $USER
```

4.1.2 Step 2: Downloading Packages

First, clone the `pepper_tools` package:

```
mkdir -p $HOME/src/Pepper
cd $HOME/src/Pepper
git clone https://bitbucket.org/mtlazaro/pepper_tools.git
```

Then clone `modim`, build the Docker image, and run it:

```
cd $HOME/src/Pepper
git clone https://bitbucket.org/mtlazaro/modim.git
mkdir -p $HOME/playground
cd $HOME/src/Pepper/pepper_tools
git pull

cd $HOME/src/Pepper/docker
./build.bash
./run.bash X11
```

Next, clone our repository into the `playground` folder and access the container:

```
cd $HOME/playground
git clone https://github.com/Feedback02/pepper_hri.git

cd $HOME/src/Pepper/docker
docker exec -it pepperhri tmux a
./run_nginx.bash ~/playground/pepper_hri
```

4.1.3 Step 3: Configuring the Host OS

Add environment variables to your `.bashrc`:

```
gedit ~/.bashrc
# At the beginning of the file, add:
export MODIM_HOME=${HOME}/src
export MODIM_APP=$HOME/playground/pepper_hri
export PATH_PEPPER_TOOLS=$HOME/src/Pepper/pepper_tools
```

4.1.4 Step 4: Starting the Emotion Server

Navigate to the emotion server directory, build the Docker image, and run it:

```
cd $HOME/playground/emotion_server
docker build -t emotion_server_image .
docker run -d --name emotion_server_container \
  --net=host emotion_server_image
```

To confirm that the server is working, post an emotion:

```
curl -X POST -H "Content-Type: application/json" \
  -d '{"emotion": "happy"}' http://localhost:5000/emotion
```

Then retrieve the stored emotion:

```
curl http://localhost:5000/get_emotion
```

4.1.5 Step 5: Running the Application

1. **Access the Docker Container:**

   ```
   cd $HOME/src/Pepper/docker
   docker exec -it pepperhri tmux a
   ```

2. **Start NAOqi:**

   ```
   ./naoqi
   ```

3. **Launch the Choregraphe Simulator:**

```
./choregraphe
```

4. **Start the MODIM Server:**

   ```
   python ws_server.py -robot pepper
   ```

5. **Configure Choregraphe:** Go to `Edit` → `Preferences` → `Virtual Robot` and note the NAOqi port (e.g., 9559).

6. **Set the PEPPER_PORT Environment Variable:**

   ```
   export PEPPER_PORT=9559  # your port number
   ```

7. **Restart MODIM Server:**

   ```
   python ws_server.py -robot pepper
   ```

8. **Start the Nginx Server (in another terminal):**

   ```
   ./run_nginx.bash ~/playground/pepper_hri
   ```

9. **Access the Tablet Simulator:** Open `http://localhost` in a browser.

10. **Run the Main Application:**

    ```
    cd ~/playground/pepper_hri
    python demo1.py
    ```

4.2 System Components

For consistent deployment we used Docker. Docker isolates processes and dependencies, letting multiple applications run on the same host without conflicts. The main components are:

- **Docker Platform:** Containers the application for reproducible deployment.
- **Choregraphe:** A SoftBank Robotics tool to simulate Pepper's movement, speech, and sensors.
- **NAOqi Framework:** The middleware that manages Pepper's core capabilities (movement, speech, sensors).
- **ModIM Server:** Manages dialogue and multimodal interactions, deciding Pepper's simulated actions.
- **ModIM Client:** Runs on Pepper (or the simulator), relaying sensor data and executing server-generated commands.

- **Application:** Oversees the logic for user-robot interaction, game rules, and scenario flow.
- **Emotion Server:** Processes real-time emotion data from a webcam, passing it to the application so Pepper can adapt its behavior.
- **Ubuntu Environment:** Provides a stable foundation for Docker and Choregraphe.

Since each container shares the host network, we did not need additional configuration for cross-container communication.

4.3 Module Implementations

4.3.1 Emotion Detection Module

Overview This module uses `face-api.js` in a browser to detect facial expressions from a webcam feed in real time.

Implementation Details

- **Camera Access:** Handled via the `getUserMedia` API.
- **Model Loading:** Loads pre-trained `face-api.js` models asynchronously.
- **Emotion Analysis:** Calculates probabilities for each emotion and selects the most likely.
- **Data Transmission:** Sends results to the Emotion Server using AJAX.

```
async function detectEmotion() {
  const detections = await faceapi
    .detectAllFaces(video, new faceapi.TinyFaceDetectorOptions())
    .withFaceExpressions();
  if (detections.length > 0) {
    const { expressions } = detections[0];
    const maxEmotion = Object.keys(expressions).reduce((a, b) =>
      expressions[a] > expressions[b] ? a : b
    );
    // Send maxEmotion to the server
  }
}
```

4.3.2 Emotion Server

Overview A Flask application that stores and provides the latest emotion to the robot.

Implementation Details

- **Endpoints:**
  - `/emotion` (POST) updates the emotion state.
  - `/get_emotion` (GET) retrieves the current emotion.
- **Data Storage:** Maintains the latest emotion in memory.
- **CORS:** Enabled via `flask-cors` to allow cross-origin requests.
- **Containerization:** Packaged as a Docker image for consistent deployment.

```
from flask import Flask, request, jsonify
from flask_cors import CORS

app = Flask(__name__)
CORS(app)

latest_emotion = {"emotion": "none"}

@app.route('/emotion', methods=['POST'])
def receive_emotion():
    data = request.get_json()
    if 'emotion' in data:
        latest_emotion['emotion'] = data['emotion']
        return jsonify({"status": "success"}), 200
    return jsonify({"status": "failed"}), 400

@app.route('/get_emotion', methods=['GET'])
def get_emotion():
    return jsonify(latest_emotion), 200
```

4.3.3 Robot Control Module

Overview This module runs on Pepper (or its simulator) and adjusts Pepper's actions based on user emotions and interactions.

Implementation Details

- **MODIM Client:** Coordinates Pepper's speech and gestures.
- **Engagement Zone Monitoring:** Tracks user proximity to the robot.
- **Behavior Execution:** Calls predefined Choregraphe behaviors.
- **Emotion Data Integration:** Fetches current emotion from the Emotion Server to guide Pepper's responses.

```
def interaction():
    im.init()
    # If someone is detected in Zone 1
    if current_zones[0] > 0:
        emotion = get_latest_emotion()
        if emotion == 'happy':
            im.execute('happy_response')
        elif emotion == 'sad':
            im.execute('encourage_response')
        # Additional quiz or story logic
```

4.3.4 Engagement Zone Module

Overview Monitors the distance between the user and Pepper, categorizing proximity into different engagement zones.

Implementation Details

- **Data Acquisition:** Uses `ALMemory` to read sensor values.
- **Threaded Monitoring:** Continuously checks zone data in a background thread.
- **Shared Data:** Makes zone info accessible to other modules.

```
class EngagementZoneMonitor:
    def __init__(self, session):
        self.memory_service = session.service("ALMemory")
        self.engagementValueList = [
            "EngagementZones/PeopleInZone1",
            "EngagementZones/PeopleInZone2",
            "EngagementZones/PeopleInZone3"
        ]
```

```
def start_monitoring(self):
    # Starts a thread that reads zone data
def get_current_zones(self):
    # Returns the current zone counts
```

4.4 Data Structures and Knowledge Representation

- **User Interaction History:** Tracks whether the user has previously interacted with the robot. In the simulation version, this was managed using a boolean variable associated with the user, whereas with the real robot, it can be handled using Aldebaran's modules ALUserSession and ALUserInfo.
- **Engagement Zones:** Stored as a list containing the IDs of individuals present in each zone, allowing the system to determine the number of people in the zone.

4.5 User Mental Model

During the interaction, the robot creates a mental model of the user. Below are the data structures used to manage it:

- **Emotion Data:** Maintained as a simple dictionary reflecting the latest detected emotion.
- **Displayed Letters**: A list containing the letters that have already been shown during the game, helping to determine when there is a need to teach a new letter and when it can be assumed to be already known.
- **Letter correctness percentage**: A dictionary where each letter of the alphabet is a key, and the value is a number between 0 and 1, representing the percentage of times that letter has been incorrectly guessed. A value of 0 indicates the letter has always been guessed correctly, while a value of 1 means the letter has either never been guessed or has always been guessed incorrectly. This value reflects the robot's perception of the user's knowledge of the ASL alphabet. It is used in the game mechanics to prioritize requesting letters that the user has difficulty with, ensuring those letters are practised more frequently.

4.6 Tools and Libraries

- **Languages:** Python (backend, robot control) and JavaScript (frontend).
- **Libraries:**
  - `face-api.js` for facial recognition and emotion analysis.
  - `Flask` + `flask-cors` for the Emotion Server.
  - `MODIM` for dialog management.
  - `NAOqi SDK` for controlling Pepper's hardware and sensors.
- **Docker:** Containerizes the Emotion Server for consistent deployment.
- **Choregraphe:** Used to design Pepper's interactive behaviors.

4.7 Development Efforts

- **Custom Components:**
  - Frontend scripts for emotion detection.
  - Flask-based Emotion Server (Dockerized).
  - Robot control scripts integrating MODIM and NAOqi.
  - Engagement Zone monitor.
  - Choregraphe behaviors for sign language demonstrations.
- **External Libraries:**
  - `face-api.js` for emotion recognition.
  - `Flask` and `flask-cors` for the server.
  - `MODIM` for dialog flow.
  - `NAOqi` for Pepper's APIs.

4.8 Challenges Encountered

- **Integration:** Making sure each component (web frontend, Flask, robot) communicated smoothly.
- **Real-Time Performance:** Keeping emotion detection fast enough to guide Pepper's behavior without delays.
- **Concurrency:** Handling simultaneous threads, particularly for engagement zone tracking.

- **Environment Consistency:** Used Docker to avoid version mismatches during deployment.
- **Resource Constraints:** Managing CPU and memory usage with multiple processes running in parallel.

4.9 Testing and Evaluation

We tested the system with the Choregraphe simulated Pepper robot, focusing on:

- **Emotion Detection Accuracy:** Ensuring the system identifies basic facial expressions correctly in varied lighting.
- **Engagement Monitoring:** Confirming Pepper reacts appropriately when users approach or walk away.
- **Interaction Flow:** Verifying the storytelling and quiz features run smoothly and logically.
- **Behavior Execution:** Checking that Choregraphe behaviors trigger and complete as intended.

5 Results

The project achieved its objectives by creating an interactive learning experience where the Pepper robot teaches sign language to children through storytelling. Below are key results demonstrating how the objectives were met.

5.1 Intent Recognition and Interaction Initiation

Situation The user approaches the system without providing any explicit input, indicating a potential intention to interact.

Robot's Behaviour The proximity sensors detect the user's presence. The robot responds by waving its hand and saying:

"Hello! Come close, I need your help!"

This proactive approach encourages engagement without requiring a direct request from the user, demonstrating social awareness and adaptive intelligence.

5.2 Positive Reinforcement on Correct Answers

Situation The child correctly answers a quiz question about a sign learned in the story.

Robot's Behavior The robot enthusiastically replies:
"Great job! You're really getting the hang of this!"
It then introduces the next sign, keeping the child engaged.

5.3 Assistance on Incorrect Answers

Situation The child provides an incorrect answer during the quiz.

Robot's Behaviour The robot offers encouragement:
"That's okay, sometimes it's tricky. Let's review that sign together."
It then revisits the sign, making sure that the child understands before proceeding.

5.4 Emotional Support and Adaptation

Situation The child appears sad or frustrated.

Robot's Behaviour The robot detects the "sad" emotion and responds empathetically: *I see you're feeling upset, but don't worry, we'll figure it out together!*

This helps alleviate frustration and keeps the child motivated.

If the robot detects an "angry" emotion, it responds in a calm and reassuring manner:

"You're looking a bit frustrated, but that's okay! The more we try, the better we get!"

This helps the child regulate their emotions and creates a supportive environment for continued interaction.

If the robot detects a repeated mistake along with a negative mental state, it suggests a short break:

"I know learning new things can be tough, but you're doing great! Let's take a short break."

Along with this, the robot plays a short video to mitigate frustration and help the child regain focus before continuing the task.

5.5 Achievement of Objectives

The system successfully integrated emotion detection, engagement monitoring, and adaptive interaction strategies to create a personalized learning experience. The objectives outlined in Section 1.2 were met as follows:

- **Objective 1:** Demonstrated social intelligence through empathetic responses and personalized interaction flows. Developed a system capable of real-time emotion detection and engagement monitoring.
- **Objective 2:** Implemented adaptive behaviors in the robot, adjusting interactions based on the user's emotional state and engagement level.
- **Objective 3:** Created an educational narrative (Marco's story) that effectively teaches sign language to children.

6 Experimental evaluation

In the following section, we present an experiment designed to evaluate the effectiveness of the robot in interacting with children in a controlled environment. The aim is to analyze whether and to what extent the introduction of the robot can help mitigate anxiety and foster the language learning process through the use of social signals and mental models.

Below are the research questions to be investigated and the corresponding hypotheses formulated:

> **Research Question 1:** How does the presence of the robot affect the engagement of the game compared to interaction with just a tablet?
> **Hypothesis 1:** Interactions with the robot promote a higher level of perceived enjoyment by the user and prolongs the duration of the interaction compared to exclusive tablet use.
>
> **Research Question 2:** How does the presence of mental model knowledge modify learning effectiveness?
> **Hypothesis 2:** The presence of mental models associated with the user increases the number of letters learned.
>
> **Research Question 3:** How does the presence of social cues by the robot change the emotional state of the user?
> **Hypothesis 3:** The presence of social cues from the robot is associated with a decrease in user anxiety and stress.

The experiment considers three independent variables:

1. **Embodiment**: determines whether interaction occurs with exploiting the presence of the robot (Pepper robot) or not (tablet only).
   - Possible values: only tablet, Pepper robot.
2. **Social signals**: determines the presence or absence of social signals such as robot body movements or encouragement feedback.
   - Possible values: no social signals, generation of social signals.

3. **Mental model knowledge**: defines whether the robot uses a mental model of the user to personalize the interaction. In "with mental model knowledge" condition, the robot adapts its behaviour based on the participant's previous responses, selecting the next letters to teach based on the user's performance.

    - Possible values: no mental model knowledge, with mental model knowledge.

The dependent variables to be measured are:

1. **Duration of interaction**: evaluates whether the game is completed and, if not, the time of leaving.
2. **Learning effectiveness**: assesses how many new ASL letters the participant successfully acquires during the interaction.
3. **Ability to create and maintain a positive social atmosphere**: evaluates how the user's emotions are affected by the robot (e.g. decreases the user's stress level or anxiety).
4. **Enjoyment of interaction**: Assesses the perceived enjoyment of the subject.

The null hypotheses associated with our experiment are as follows:

1. Robot use has no impact on interaction duration or perceived enjoyment;
2. The presence of the mental model does not affect the number of letters learned;
3. The robot's use of social signals does not go to impact the user's level of anxiety or stress.

The experiment involves children aged 8 to 10 years. They are selected from different elementary schools in order to ensure variability in the background of the subjects. A between-subjects protocol is adopted, in which participants are randomly assigned to one of five experimental groups:

1. interaction only with tablet,
2. interaction with robots without social signals and without mental model knowledge,
3. interaction with robots with social signals and without mental model knowledge,
4. interaction with robots without social signals and with mental model knowledge,
5. interaction with robots with social signals and with mental model knowledge.

Each participant is given a preliminary questionnaire designed to assess prior knowledge of the sign alphabet and level of agitation. Next, the participant takes part in an individual interaction session with the robot or tablet, depending on the assigned experimental group. During the session, the interaction is recorded via video, and questions asked, answers given, and any interactions with the robot or tablet are tracked. At the end of the interaction, a second survey is administered to measure knowledge acquisition, any changes in emotional state and enjoyment perceived. The collected data are then processed to calculate the p-value. Based on the value obtained, conclusions are made: if the p-value is less than 0.05, the correlation between the variables is considered statistically significant, confirming the significance of the experimental factors considered.

7 Conclusion

Of this project, we appreciated the opportunity to explore topics specific to Human-Robot Interaction (HRI), such as Child-Robot Interaction and social signals. The interdisciplinary nature of the project allowed us to combine knowledge of robotics and artificial intelligence and integrate it with psychological knowledge related to human-robot interaction. We made new considerations compared to those made in other projects more focused on the technical part. We saw that its important to monitor and interpret the social signals provided by the user in order to be able to ensure a positive experience. Developing the project, we were able to better understand the importance of perceptions and the interpretation of social signals.

Future improvements can be done to the game dynamics and the robot's ability to analyze its surroundings. For example, the game could ask players to mimic a specific gesture, and by using vision recognition algorithms, the robot could provide feedback on the accuracy of the gesture performed. Also the robot's social skills could be improved, by collecting data with each interaction and apply a reinforcement learning technique to continuously improve, getting a final feedback from the child at each interaction. Other potential improvements include expanding emotion recognition to capture more nuanced emotional states. Not forgetting the importance of developing additional stories and learning modules that could improve the educational experience.

Potential critical issues in terms of child safety and safeguarding the robot are to be considered. Imagining as hypothetical application waiting rooms with high child attendance, we need to take into account the unpredictable nature of the users, who, being at an early age and lacking specific training, may not be fully aware of the correct ways to interact with the robotic system. The absence of proper monitoring could result in the risk of injury to children or damage to the robot, thus necessitating the implementation of effective safety strategies to mitigate these issues.

With reference to the above, in addition to the implementation of the necessary safety measures, it is also important to consider the economic implications of any damage to the robot, as well as the costs associated with the presence of

a possible dedicated supervisory operator.

To conclude on an important note, it is necessary to consider the ethical and social implications related to the introduction of robots in environments frequented by children, questioning the role that such technologies may take on in social and cognitive development, and how they may change their relationship with reality and interpersonal relationships.

References

[1] Pınar Uluer, Neziha Akalın, and Hatice Köse. A new robotic platform for sign language tutoring: Humanoid robots as assistive game companions for teaching sign language. *International Journal of Social Robotics*, 7:571–585, 2015.

[2] Mohammad Zakipour, Ali Meghdari, and Minoo Alemi. Rasa: A low-cost upper-torso social robot acting as a sign language teaching assistant. volume 9979, pages 630–639, 11 2016.

[3] Per Backlund and Maurice Hendrix. Educational games-are they worth the effort? a literature survey of the effectiveness of serious games. In *2013 5th international conference on games and virtual worlds for serious applications (VS-GAMES)*, pages 1–8. IEEE, 2013.

[4] Maria Virvou, George Katsionis, and Konstantinos Manos. Combining software games with education: Evaluation of its educational effectiveness. *Journal of Educational Technology & Society*, 8(2):54–65, 2005.

[5] Thomas Malone. What makes things fun to learn? a study of intrinsically motivating computer games. *Pipeline*, 6, 01 1981.

[6] Fumihide Tanaka and Shizuko Matsuzoe. Children teach a care-receiving robot to promote their learning: Field experiments in a classroom for vocabulary learning. *Journal of Human-Robot Interaction*, 1, 08 2012.

[7] Monica Perusquía-Hernández, Marisabel Cuberos Balda, David Antonio Gómez Jáuregui, Diego Paez-Granados, Felix Dollack, and Jose Victorio Salazar. Robot mirroring: Promoting empathy with an artificial agent by reflecting the user's physiological affective states. In *2020 29th IEEE International Conference on Robot and Human Interactive Communication (RO-MAN)*, pages 1328–1333, 2020.

[8] Luca Iocchi. Knowledge representation and planning in hri. Human-Robot Interaction 2024. Lecture slide.

[9] Silvia Rossi, Marwa Larafa, and Martina Ruocco. Emotional and behavioural distraction by a social robot for children anxiety reduction during vaccination. *International Journal of Social Robotics, Springer Nature*, 12, 01 2020.

[10] Omar Mubin Muneeb Imtiaz Ahmad and Joanne Orlando. Adaptive social robot for sustaining social engagement during long-term children–robot interaction. *International Journal of Human–Computer Interaction*, 33(12):943–962, 2017.

[11] Luca Iocchi. Multi-modal interaction. Human-Robot Interaction 2024. Lecture slide.

[12] Luca Iocchi. Motion control for hri. Human-Robot Interaction 2024. Lecture slide.