USI

Università della Svizzera italiana

Faculty of Informatics

Exploring the Learning by Teaching with Social Robots

Enabling Exploratory Studies on Learning by Teaching with Social Robots through Wizard-of-Oz Control

Davide Frova

Abstract

Abstract goes here ... You may include up to six keywords or phrases. Keywords should be separated with semicolons.
Keywords:

Contents

1 Introduction

Social robots are increasingly finding their way into learning environments, where their role as collaborators, tutors, or learning companions is being actively explored. In particular, the paradigm of *Learning by Teaching* (LbT) offers a promising framework in which children can reinforce their understanding and social-emotional skills by teaching a robot. This approach has the potential to enhance engagement, improve soft skills, and support inclusive education.

This bachelor project is part of the broader TESORO initiative, for which an SNSF funding application has been submitted and is currently awaiting approval. My contribution focuses on a practical implementation: the design and development of a web-based dashboard that enables a Wizard-of-Oz control of a social robot during an exploratory experiment involving turn-taking with children.

The experiment consists of a Lego-building task performed by two children taking turns. The robot, remotely controlled via the dashboard, intervenes when turn-taking violations occur, aiming to regulate the interaction and reinforce collaborative behavior.

This report presents the background of the project, the research and implementation goals, the technical architecture of the system, and the planned experimental study. The developed dashboard serves as a starting point for preliminary studies and lays the groundwork for later stages of the TESORO project where the robot will learn these regulatory behaviors from various operators, including researchers, teachers, or even the children themselves. Additionally, it could also serve other projects that are based on Human-Robot Interaction (HRI) since it will be developed in a way that is re-usable for different scenarios.

1.1 Report structure

The rest of the report is organized as follows: Section 2 presents the related work and theoretical framework; Section 3 outlines the experiment scenario and research goals; Section 4 describes the dashboard's architecture and implementation; Section 5 discusses the system's current status and outlines the upcoming evaluation; and Section 6 concludes with future directions.

2 State of the art

Explain all acronyms and abbreviations. For example, the first time an acronym is used, write it out in full and place the acronym in parentheses. When using the Graphical User Interface (GUI) version, the use may...

3 Experiment Design and Goals

The experiment designed for this project aims to explore how a social robot can support children's learning through turn-taking regulation. The overall investigation is structured into three progressive stages:

- **Step 1: Robot as Regulator** - Two children collaboratively build a Lego tower by taking turns. The robot observes and intervenes in case of turn violations, using multimodal cues such as LED lights, vocalizations, and gestures.

- **Step 2: Child as Regulator** - A child is tasked with regulating the interactions between the robot and another child that will perform a similar task to Step 1. This step exploits the LbT paradigm, where the regulator learns how to collaborate and keep turns.

- **Step 3: Robot Learns How and When to Intervene** - The robot applies the learned behavior during interactions with children in Step 1 and Step 2.

This report provides the foundational knowledge for implementing later stages of autonomous learning. The robot's interventions are controlled via a Wizard-of-Oz setup, and the primary goal is to assess the feasibility and effectiveness of such interventions in developing children's soft skills.

4 System Design and Implementation

4.1 System Architecture

Overview

The system architecture was designed to enable an operator to control a social robot in a Wizard-of-Oz fashion during experiments with children. The main components include a web-based dashboard for real-time control [2], a communication layer based on ROS2 (Robot Operating System 2) [6, 4, 3], and the RoboMaster EP robot [1]. The dashboard provides intuitive controls for triggering robot interventions and monitoring status, while the ROS2 integration ensures reliable communication and execution of commands. This architecture prioritizes usability, low latency, and safety, supporting the experimental requirements for responsive and effective interactions. The system was designed to facilitate iterative development and easy modifications, supporting a co-design approach as outlined in the TESORO project proposal [5]. This flexibility enables rapid adaptation of the dashboard based on feedback from stakeholders and participants during the study.

4.2 Component Description

Dashboard

The dashboard is a web-based interface developed using Next.js, with the user interface built on Material UI (MUI) components for a modern and consistent look. It allows the operator to monitor the robot's status and trigger interventions in real time. Communication with the robot is achieved via `roslibjs` [9], which enables the dashboard to send and receive ROS messages over the network.

Middleware Server

Due to a known limitation in `roslibjs` when interacting with ROS2 action servers, a minimal middleware server was implemented within the dashboard system. This server is built with Express and uses `rclnodejs` [7] to interface directly with ROS2, acting as a bridge between the dashboard and ROS2 action servers. This ensures reliable execution of complex robot actions and exposes additional functionalities to the web interface.

ROS Communication Layer

The communication layer is based on ROS2 and uses `rosbridge_server` [8] running on the Ubuntu machine that controls the robot ROS2 drivers. This layer handles the communication between the dashboard and the robot, allowing for real-time updates and command execution.

Robot Control and Action Servers

The Ubuntu machine runs several ROS2 nodes and action servers, including:

- `robomaster_ros` [4]: ROS2 drivers for the RoboMaster EP robot, providing low-level control and sensor access.

- `robomaster_hri` [3]: Custom packages for high-level robot control and human-robot interaction.

- Security and safety action servers: Allow the operator to immediately halt robot actions or trigger safety protocols as needed.

RoboMaster EP Robot

The RoboMaster EP robot [1] is the physical platform used for the experiment. It connects to the Ubuntu machine via its dedicated access point, ensuring a stable and isolated communication channel. The robot executes commands received from the ROS2 action servers and ROS2 nodes, and provides feedback to the dashboard.

Network and Deployment

All components are deployed on a local network to ensure low latency and reliability. The dashboard can be accessed from any device on the network, while the Ubuntu machine acts as the central hub for ROS2 communication and robot control. Since the RoboMaster EP robot connects wirelessly and provides odometry data, the system is not restricted to a laboratory setting. This flexibility allows experiments and interactions with children to take place in more neutral or familiar environments, which can be beneficial for naturalistic studies and participant comfort.

4.3 Safety Considerations

Ensuring the safety of participants and equipment was a primary concern throughout the system design. To address this, a dedicated ROS2 node was developed to act as an emergency stop mechanism. This node can immediately halt all ongoing robot actions and stop the RoboMaster EP's motors.

A prominent "Emergency Stop" button is integrated into the dashboard interface, allowing the operator to trigger this safety mechanism at any time during an experiment. When activated, the command is sent via the ROS2 communication layer to the safety node running on the Ubuntu machine, ensuring a rapid response. This feature is essential for experiments involving children, as it provides the operator with immediate control to prevent unintended or unsafe robot behavior. The system was tested to ensure that the emergency stop reliably overrides all other commands and brings the robot to a safe state.
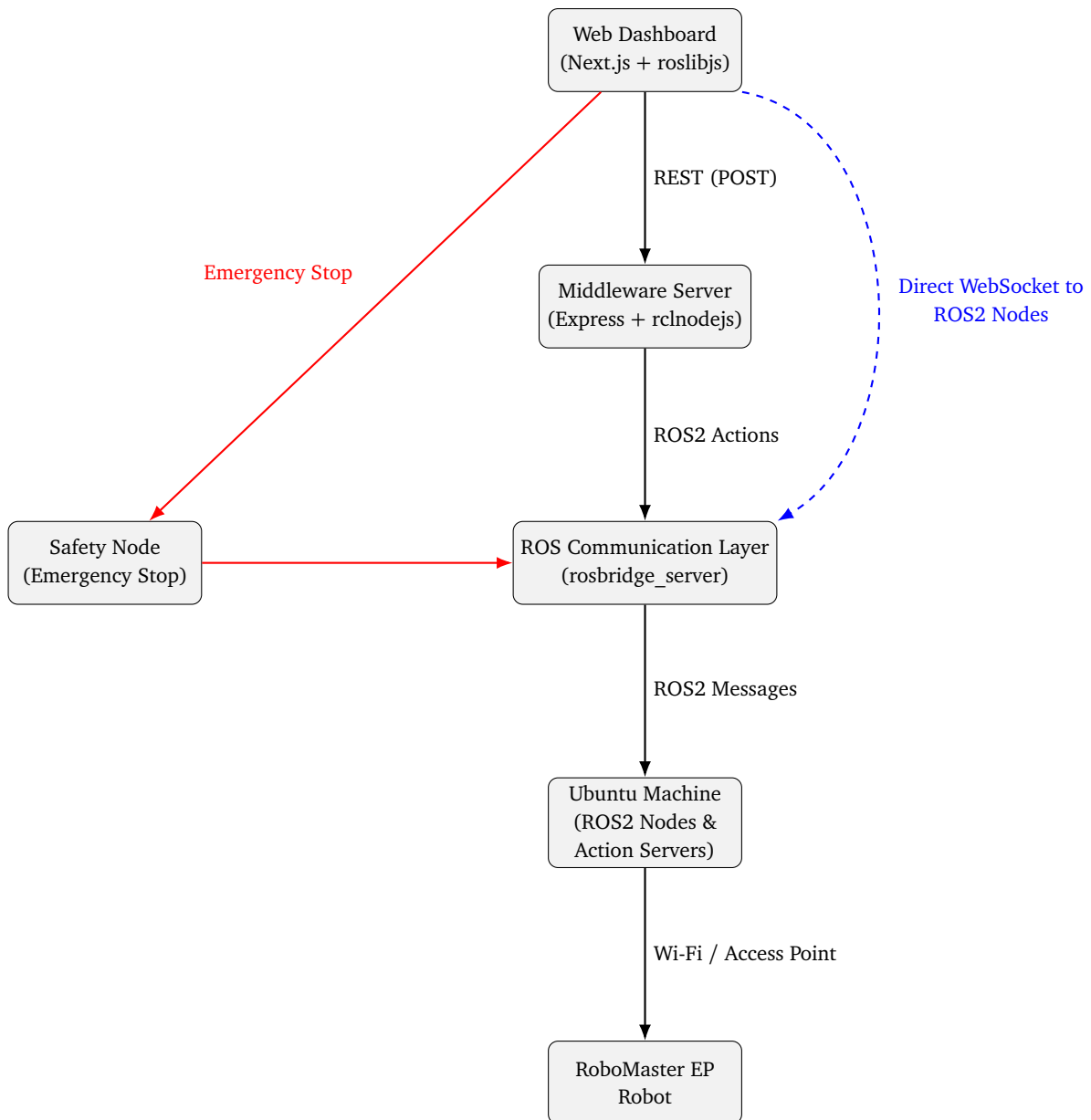
4.4 System Architecture Diagram

Figure 1. System architecture for Wizard-of-Oz control of the RoboMaster EP robot. The dashboard communicates with the middleware server via REST (POST request), and can also communicate directly with the ROS2 nodes using WebSocket (roslibjs).

4.5   Dashboard Interface

> Here we will describe the dashboard interface, including the main features and functionalities. We will also include screenshots of the dashboard to illustrate its design and layout.

5   Evaluation and Current Status

The current implementation of the dashboard is functional and has been tested with the DJI RoboMaster EP robot. It successfully supports all necessary actions for the experiment.

A pilot study is planned where the dashboard will be used to conduct an initial session of the Lego tower-building task with two children. The operator will use the dashboard to regulate turn-taking behavior. Key observations will include:

- Effectiveness of the robot's interventions.

- Children's responsiveness to the robot.

- Ease of use of the dashboard by the operator.

These observations will guide the refinement of both the experiment protocol and the dashboard interface.

5.1   Walkthrough with Experts

Setup and procedure

-

Key observations

-

Design implications

-

5.2   Walkthrough with High School Students

Setup and procedure

-

Key observations

-

Design implications

-

5.3   Pilot with Children

Setup and procedure

-

Key observations

-

Design implications

-

5.4 Summary of Key Iterative Changes

> Here we will summarize the key changes made to the dashboard based on the feedback received from the walkthroughs and pilot study. This section will highlight the iterative design process and how user feedback informed the development of the system.

6 Conclusions

This project presents the design and implementation of a web-based dashboard to enable Wizard-of-Oz control of a social robot used in a Learning by Teaching experiment with children. The system is part of the larger TESORO initiative and lays the groundwork for studying how social robots can assist in learning soft skills like turn-taking.

To be expanded

6.1 Future Work

Here we will write about future work like the ones mentioned in the TESORO project proposal like :
Long-term goal: enabling real-time learning from the child's regulatory actions and shifting toward semi-autonomous robot behavior.

References

[1] DJI. Robomaster EP. https://www.dji.com/ch/robomaster-ep/specs. Accessed: 2025-05-21.

[2] D. Frova. H.O.G.W.A.R.T.S – human operated gui for wizardry and remote telepresence system. https://github.com/frovaaa/hogwarts, 2025. Web-based dashboard for Wizard-of-Oz control.

[3] D. Frova. robomaster_hri: ROS2 packages for human-robot interaction with robomaster EP. https://github.com/frovaaa/robomaster_hri, 2025. Private repository. Fork of original codebase, expanded for navigation and interaction in HRI experiments.

[4] D. Frova. robomaster_ros: ROS2 drivers for robomaster EP. https://github.com/frovaaa/robomaster_ros, 2025. Fork of original codebase, expanded for this project.

[5] M. Landoni and A. Paolillo. TESORO: TEaching SOcial RObots to help children learn about themselves. SNSF Project Proposal. Unpublished project proposal, shared via personal communication, 2025.

[6] Open Source Robotics Foundation. Robot Operating System 2 (ROS 2). https://www.ros.org. Accessed: 2025-05-21.

[7] Robot Web Tools. rclnodejs. https://github.com/RobotWebTools/rclnodejs. Accessed: 2025-05-21.

[8] Robot Web Tools. rosbridge_server. https://github.com/RobotWebTools/rosbridge_suite/tree/ros2/rosbridge_server. Accessed: 2025-05-21.

[9] Robot Web Tools. roslibjs. https://github.com/RobotWebTools/roslibjs. Accessed: 2025-05-21.