Collaborative and Reproducible HRI Research Through a Web-Based Wizard-of-Oz Platform

Sean O'Connor and L. Felipe Perrone*

***Abstract*— Human-robot interaction (HRI) research plays a pivotal role in shaping how robots communicate and collaborate with humans. However, conducting HRI studies can be challenging, particularly those employing the Wizard-of-Oz (WoZ) technique. WoZ user studies can have technical and methodological complexities that may render the results irreproducible. We propose to address these challenges with HRIStudio, a modular web-based platform designed to streamline the design, the execution, and the analysis of WoZ experiments. HRIStudio offers an intuitive interface for experiment creation, real-time control and monitoring during experimental runs, and comprehensive data logging and playback tools for analysis and reproducibility. By lowering technical barriers, promoting collaboration, and offering methodological guidelines, HRIStudio aims to make human-centered robotics research easier and empower researchers to develop scientifically rigorous user studies.**

I. Introduction

Human-robot interaction (HRI) is an essential field of study for understanding how robots should communicate, collaborate, and coexist with people. The development of autonomous behaviors in social robot applications, however, offers a number of challenges. The Wizard-of-Oz (WoZ) technique has emerged as a valuable experimental paradigm to address these difficulties, as it allows experimenters to simulate a robot's autonomous behaviors. With WoZ, a human operator (the *"wizard"*) can operate the robot remotely, essentially simulating its autonomous behavior during user studies. This enables the rapid prototyping and continuous refinement of human-robot interactions postponing to later the full development of complex robot behaviors.

While WoZ is a powerful paradigm, it does not eliminate all experimental challenges. The paradigm is centered on the wizard who must carry out scripted sequences of actions. Ideally, the wizard should execute their script identically across runs of the experiment with different participants. Deviations from the script in one run or another may change experimental conditions significantly decreasing the methodological rigor of the larger study. This kind of problem can be minimized by instrumenting the wizard with a system that prevents deviations from the prescribed interactions with the participant. In addition to the variability that can be introduced by wizard behaviors, WoZ studies can be undermined by technical barriers related to the use of specialized equipment and tools. Different robots may be controlled or programmed through different systems requiring expertise with a range of technologies such as programming languages, development environments, and operating systems.

The elaboration and the execution of rigorous and reproducible WoZ experiments can be challenging for HRI researchers. Although there do exist solutions to support this kind of endeavor, they often rely on low-level robot operating systems, limited proprietary platforms, or require extensive custom coding, which can restrict their use to domain experts with extensive technical backgrounds. The development of our work was motivated by the desire to offer a platform that would lower the barriers to entry in HRI research with the WoZ paradigm.

Through the literature review described in the next section, we identified six categories of desirables to be included in a modern system that streamlines the WoZ experimental process: an environment that integrates all the functionalities of the system; mechanisms for the description of WoZ experiments which require minimal to no coding expertise; fine grained, real-time control of scripted experimental runs with a variety of robotic platforms; comprehensive data collection and logging; a platform-agnostic approach to support a wide range of robot hardware; and collaborative features that allow research teams to work together effectively.

The design and development of HRIStudio were driven by the desirables enumerated above and described in [5], our preliminary report. In this work, our main contribution is to demonstrate how a system such the one we are developing has significant potential to make WoZ experiments easier to carry out, more rigorous, and ultimately reproducible. The remainder of this paper is structured as follows. In Section II, we establish the context for our contribution through a review of recent literature. In Section III, we discuss the aspects of the WoZ paradigm that can lead to reproducibility challenges and in Section IV we propose solutions to address these challenges. Subsequently, in Section V, we describe our solution to create a structure for the experimental workflow. Finally, in Section VI, we conclude the paper with a summary of our contributions, a reflection on the current state of our project, and directions for the future.

II. Assessment of the State-of-the-Art

Over the last two decades, multiple frameworks to support and automate the WoZ paradigm have been reported in the literature. These frameworks can be categorized according to how they focus on four primary areas of interest, which we discuss below as we expose some of the most important contributions to the field.

*Both authors are with the Department of Computer Science at Bucknell University in Lewisburg, PA, USA. They can be reached at sso005@bucknell.edu and perrone@bucknell.edu

A. Technical Infrastructure and Architectures

The foundation of any WoZ framework lies in its technical infrastructure and architectural design. These elements determine not only the system's capabilities but also its longevity and adaptability to different research needs. Several frameworks have focused on providing robust technical infrastructures for WoZ experiments.

Polonius [4] utilizes the modular *Robot Operating System* (ROS) platform as its foundation, offering a graphical user interface for wizards to define finite-state machine scripts that drive robot behaviors. A notable feature is its integrated logging system that eliminates the need for post-experiment video coding, allowing researchers to record human-robot interactions in real-time as they occur. Polonius was specifically designed to be accessible to non-programming collaborators, addressing an important accessibility gap in HRI research tools.

OpenWoZ [3] takes a different approach with its runtime-configurable framework and multi-client architecture, enabling evaluators to modify robot behaviors during experiments without interrupting the flow. This flexibility allows for dynamic adaptation to unexpected participant responses, though it requires programming expertise to create customized robot behaviors. The system's architecture supports distributed operation, where multiple operators can collaborate during an experiment.

B. Interface Design and User Experience

The design of an interface for the wizard to control the execution of an experiment is of paramount importance. The qualities of the interface can significantly impact faithfulness of the experiment's execution, the quality of data collected, and the longevity of the tool itself. *NottReal* [7] exemplifies careful attention to interface design in its development for voice user interface studies. The system makes it easier for the wizard to play their role featuring tabbed lists of pre-scripted messages, slots for customization, message queuing capabilities, and comprehensive logging. Its visual feedback mechanisms mimic commercial voice assistants, providing participants with familiar interaction cues such as dynamic "orbs" that indicate the system is listening and processing states.

WoZ4U [11] prioritizes usability with a GUI specifically designed to make HRI studies accessible to non-programmers. While its tight integration with Aldebaran's Pepper robot constrains generalizability, it demonstrates how specialized interfaces can lower barriers to entry for conducting WoZ studies with specific platforms.

C. Domain Specialization vs. Generalizability

A key tension in WoZ framework development exists between domain specialization and generalizability. Some systems are designed for specific types of interactions or robot platforms, offering deep functionality within a narrow domain. Others aim for broader applicability across various robots and interaction scenarios, potentially sacrificing depth of functionalities for breadth.

Pettersson and Wik's [6] systematic review identified this tension as central to the longevity of WoZ tools, that is, their ability to remain operational despite changes in underlying technologies. Their analysis of 24 WoZ systems revealed that most general-purpose tools have a lifespan of only 2-3 years. Their own tool, Ozlab, achieved exceptional longevity (15+ years) through three factors: (1) a truly general-purpose approach from inception, (2) integration into HCI curricula ensuring institutional support, and (3) a flexible wizard interface design that adapts to specific experimental needs rather than forcing standardization.

D. Standardization Efforts and Methodological Approaches

The tension between specialization and generalizability has led to increased interest in developing standardized approaches to WoZ experimentation. Recent efforts have focused on developing standards for HRI research methodology and interaction specification. Porfirio et al. [8] proposed guidelines for an *interaction specification language* (ISL), emphasizing the need for standardized ways to define and communicate robot behaviors across different platforms. Their work introduces the concept of *Application Development Environments* (ADEs) for HRI and details how hierarchical modularity and formal representations can enhance the reproducibility of robot behaviors. These ADEs would provide structured environments for creating robot behaviors with varying levels of expressiveness while maintaining platform independence.

This standardization effort addresses a critical gap identified in Riek's [10] systematic analysis of published WoZ experiments. Riek's work revealed concerning methodological deficiencies: 24.1% of papers clearly described their WoZ simulation as part of an iterative design process, 5.4% described wizard training procedures, and 11% constrained what the wizard could recognize. This lack of methodological transparency hinders reproducibility and, therefore, scientific progress in the field.

Methodological considerations extend beyond wizard protocols to the fundamental approaches in HRI evaluation. Steinfeld et al. [12] introduced a complementary framework to the traditional WoZ method, which they termed "the Oz of Wizard." While WoZ uses human experimenters to simulate robot capabilities, the Oz of Wizard approach employs simplified human models to evaluate robot behaviors and technologies. Their framework systematically describes various permutations of real versus simulated components in HRI experiments, establishing that both approaches serve valid research objectives. They contend that technological advances in HRI constitute legitimate research even when using simplified human models rather than actual participants, provided certain conditions are met. This framework establishes an important lesson for the development of new WoZ platforms like HRIStudio which must balance standardization with flexibility in experimental design.

The interdisciplinary nature of HRI creates methodological inconsistencies that Belhassein et al. [1] examine in depth. Their analysis identifies recurring challenges in HRI user studies: limited participant pools, insufficient reporting of wizard protocols, and barriers to experiment replication. They note that self-assessment measures like questionnaires, though commonly employed, often lack proper validation for HRI contexts and may not accurately capture the participants' experiences. Our platform's design goals align closely with their recommendations to combine multiple evaluation approaches, thoroughly document procedures, and develop validated HRI-specific assessment tools.

Complementing these theoretical frameworks, Fraune et al. [2] provide practical methodological guidance from an HRI workshop focused on study design. Their work organizes expert insights into themes covering study design improvement, participant interaction strategies, management of technical limitations, and cross-field collaboration. Key recommendations include pre-testing with pilot participants and ensuring robot behaviors are perceived as intended. Their discussion of participant expectations and the "novelty effect" in first-time robot interactions is particularly relevant for WoZ studies, as these factors can significantly influence experimental outcomes.

E. Challenges and Research Gaps

Despite these advances, significant challenges remain in developing accessible and rigorous WoZ frameworks that can remain usable over non-trivial periods of time. Many existing frameworks require significant programming expertise, constraining their usability by interdisciplinary teams. While technical capabilities have advanced, methodological standardization lags behind, resulting in inconsistent experimental practices. Few platforms provide comprehensive data collection and sharing capabilities that enable robust meta-analyses across multiple studies. We are challenged to create tools that provide sufficient structure for reproducibility while allowing the flexibility needed for the pursuit of answers to diverse research questions.

HRIStudio aims to address these challenges with a platform that is robot-agnostic, methodologically rigorous, and eminently usable by those with less honed technological skills. By incorporating lessons from previous frameworks and addressing the gaps identified in this section, we designed a system that supports the full lifecycle of WoZ experiments, from design through execution to analysis, with an emphasis on usability, reproducibility, and collaboration.

III. Reproducibility Challenges in WoZ Studies

Reproducibility is a cornerstone of scientific research, yet it remains a significant challenge in HRI studies, particularly those centered on the Wizard-of-Oz methodology. Before detailing our platform design, we first examine the critical reproducibility issues that have informed our approach.

The reproducibility challenges affecting many scientific fields are particularly acute in HRI research employing WoZ techniques. Human wizards may respond differently to similar situations across experimental trials, introducing inconsistency that undermines reproducibility and the integrity of collected data. Published studies often provide insufficient details about wizard protocols, decision-making criteria, and response timing, making replication by other researchers nearly impossible. Without standardized tools, research teams create custom setups that are difficult to recreate, and ad-hoc changes during experiments frequently go unrecorded. Different data collection methodologies and metrics further complicate cross-study comparisons.

As previously discussed, Riek's [10] systematic analysis of WoZ research exposed significant methodological transparency issues in the literature. These documented deficiencies in reporting experimental procedures make replication challenging, undermining the scientific validity of findings and slowing progress in the field as researchers cannot effectively build upon previous work.

We have identified five key requirements for enhancing reproducibility in WoZ studies. First, standardized terminology and structure provide a common vocabulary for describing experimental components, reducing ambiguity in research communications. Second, wizard behavior formalization establishes clear guidelines for wizard actions that balance consistency with flexibility, enabling reproducible interactions while accommodating the natural variations in human-robot exchanges. Third, comprehensive data capture through time-synchronized recording of all experimental events with precise timestamps allows researchers to accurately analyze interaction patterns. Fourth, experiment specification sharing capabilities enable researchers to package and distribute complete experimental designs, facilitating replication by other teams. Finally, procedural documentation through automatic logging of experimental parameters and methodological details preserves critical information that might otherwise be omitted in publications. These requirements directly informed HRIStudio's architecture and design principles, ensuring that reproducibility is built into the platform rather than treated as an afterthought.

IV. The Design and Architecture of HRIStudio

Informed by our analysis of both existing WoZ frameworks and the reproducibility challenges identified in the previous section, we have developed several guiding design principles for HRIStudio. Our primary goal is to create a platform that enhances the scientific rigor of WoZ studies while remaining accessible to researchers with varying levels of technical expertise. We have been driven by the goal of prioritizing "accessibility" in the sense that the platform should be usable by researchers without deep robot programming expertise so as to lower the barrier to entry for HRI studies. Through abstraction, users can focus on experimental design without getting bogged down by the technical details of specific robot platforms. Comprehensive data management enables the system to capture and store all generated data, including logs, audio, video, and study materials. To facilitate teamwork, the platform provides collaboration support through multiple user accounts, role-based access control, and data sharing capabilities that enable effective knowledge transfer while restricting access to sensitive data. Finally, methodological guidance is embedded throughout the platform, directing users toward scientifically sound practices through its design and documentation. These principles directly address the reproducibility requirements identified earlier, particularly the need for standardized terminology, wizard behavior formalization, and comprehensive data capture.

We have implemented HRIStudio as a modular web application with explicit separation of concerns in accordance with these design principles. The structure of the application into client and server components creates a clear separation of responsibilities and functionalities. While the client exposes interactive elements to users, the server handles data processing, storage, and access control. This architecture provides a foundation for implementing data security through role-based interfaces in which different members of a team have tailored views of the same experimental session.

As shown in Figure 1, the architecture consists of three main functional layers that work in concert to provide a comprehensive experimental platform. The *User Interface Layer* provides intuitive, browser-based interfaces for three components: an *Experiment Designer* with visual programming capabilities for one to specify experimental details, a *Wizard Interface* that grants real-time control over the execution of a trial, and a *Playback & Analysis* module that supports data exploration and visualization.

The *Data Management Layer* provides database functionality to organize, store, and retrieve experiment definitions, metadata, and media assets generated throughout an experiment. Since HRIStudio is a web-based application, users can access this database remotely through an access control system that defines roles such as *researcher*, *wizard*, and *observer* each with appropriate capabilities and constraints. This fine-grained access control protects sensitive participant data while enabling appropriate sharing within research teams, with flexible deployment options either on-premise or in the cloud depending on one's needs. The layer enables collaboration among the parties involved in conducting a user study while keeping information compartmentalized and secure according to each party's requirements.

The third major component is the *Robot Integration Layer*, which is responsible for translating our standardized abstractions for robot control to the specific commands accepted by different robot platforms. HRIStudio relies on the assumption that at least one of three different mechanisms is available for communication with a robot: a RESTful API, standard communication structures provided by ROS, or a plugin that is custom-made for that platform. The *Robot Integration Layer* serves as an intermediary between the *Data Management Layer* with *External Systems* such as robot hardware, external sensors, and analysis tools. This layer allows the main components of the system to remain "robot-agnostic" pending the identification or the creation of the correct communication method and changes to a configuration file.

In order to facilitate the deployment of our application, we leverage containerization with Docker to ensure that every component of HRIStudio will be supported by their system dependencies on different environments. This is an important step toward extending the longevity of the tool and toward guaranteeing that experimental environments remain consistent across different platforms. Furthermore, it allows researchers to share not only experimental designs, but also their entire execution environment should a third party wish to reproduce an experimental study.

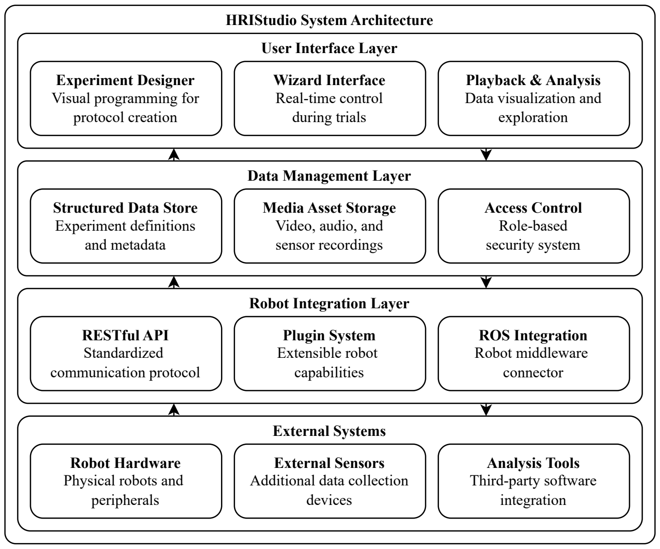

HRIStudio System Architecture
User Interface Layer
Experiment Designer
Visual programming for protocol creation
Wizard Interface
Real-time control during trials
Playback & Analysis
Data visualization and exploration
Data Management Layer
Structured Data Store
Experiment definitions and metadata
Media Asset Storage
Video, audio, and sensor recordings
Access Control
Role-based security system
Robot Integration Layer
RESTful API
Standardized communication protocol
Plugin System
Extensible robot capabilities
ROS Integration
Robot middleware connector
External Systems
Robot Hardware
Physical robots and peripherals
External Sensors
Additional data collection devices
Analysis Tools
Third-party software integration

Fig. 1. HRIStudio's three-layer architecture.

V. Experimental Workflow Support

The experimental workflow in HRIStudio directly addresses the reproducibility challenges identified in Section III by providing standardized structures, explicit wizard guidance, and comprehensive data capture. This section details how the platform's workflow components implement solutions for each key reproducibility requirement.

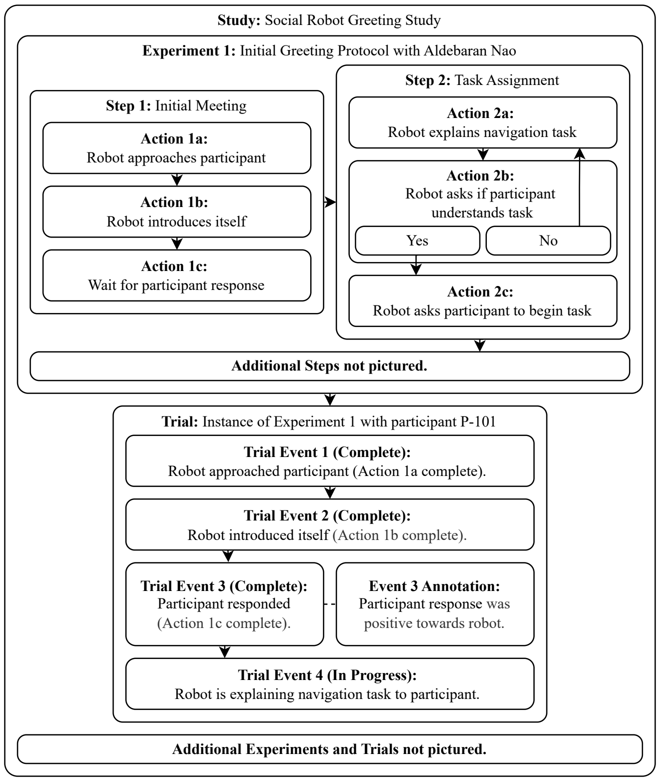
A. Embracing a Hierarchical Structure for WoZ Studies

HRIStudio defines its own standard terminology with a hierarchical organization of the elements in WoZ studies as follows.

- At the top level, an experiment designer defines a *study* element, which comprises one or more *experiment* elements.
- Each *experiment* specifies the experimental protocol for a discrete subcomponent of the overall study and comprises one or more *step* elements, each representing a distinct phase in the execution sequence. The *experiment* functions as a parameterized template.
- Defining all the parameters in an *experiment*, one creates a *trial*, which is an executable instance involving a specific participant and conducted under predefined conditions. The data generated by each *trial* is recorded by the system so that later one can examine how the experimental protocol was applied to each participant. The distinction between experiment and trial enables a clear separation between the abstract protocol specification and its concrete instantiation and execution.

- Each *step* encapsulates instructions that are meant either for the wizard or for the robot thereby creating the concept of "type" for this element. The *step* is a container for a sequence of one or more *action* elements.
- Each *action* represents a specific, atomic task for either the wizard or the robot, according to the nature of the *step* element that contains it. An *action* for the robot may represent commands for input gathering, speech, waiting, movement, etc., and may be configured by parameters specific for the *trial*.

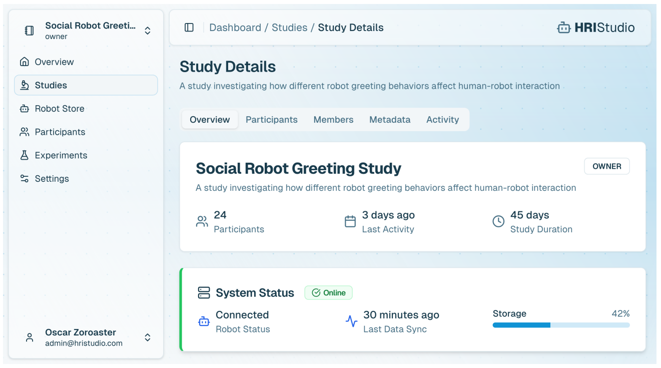
Figure 2 illustrates this hierarchical structure through a fictional study. In the diagram, we see a "Social Robot Greeting Study" containing an experiment with a specific robot platform, steps containing actions, and a trial with a participant. Note that each trial event is a traceable record of the sequence of actions defined in the experiment. HRIStudio enables researchers to collect the same data across multiple trials while adhering to consistent experimental protocols and recording any reactions the wizard may inject into the process.



Study: Social Robot Greeting Study
Experiment 1: Initial Greeting Protocol with Aldebaran Nao
Step 1: Initial Meeting
Action 1a: Robot approaches participant
Action 1b: Robot introduces itself
Action 1c: Wait for participant response
Step 2: Task Assignment
Action 2a: Robot explains navigation task
Action 2b: Robot asks if participant understands task
Yes
No
Action 2c: Robot asks participant to begin task
Additional Steps not pictured.
Trial: Instance of Experiment 1 with participant P-101
Trial Event 1 (Complete): Robot approached participant (Action 1a complete).
Trial Event 2 (Complete): Robot introduced itself (Action 1b complete).
Trial Event 3 (Complete): Participant responded (Action 1c complete).
Event 3 Annotation: Participant response was positive towards robot.
Trial Event 4 (In Progress): Robot is explaining navigation task to participant.
Additional Experiments and Trials not pictured.

Fig. 2. Hierarchical organization of a sample user study in HRIStudio.

This standardized hierarchical structure creates a common vocabulary for experimental elements, eliminating ambiguity in descriptions and enabling clearer communication among researchers. Our approach aligns with the guidelines proposed by Porfirio et al. [8] for an HRI specification language, particularly in regards to standardized formal representations and hierarchical modularity. Our system uses the formal study definitions to create comprehensive procedural documentation requiring no additional effort by the researcher.



Study Details
A study investigating how different robot greeting behaviors affect human-robot interaction

Social Robot Greeting Study
A study investigating how different robot greeting behaviors affect human-robot interaction

24 Participants
3 days ago Last Activity
45 days Study Duration

System Status Online
Connected Robot Status
30 minutes ago Last Data Sync
Storage 42%

Fig. 3. Summary view of system data on an example study.

Beyond this documentation, a study definition can be shared with other researchers for the faithful reproduction of experiments.

Figure 3 shows how the system displays the data of an experimental study in progress. In this view, researchers can inspect summary data about the execution of a study and its trials, find a list of human subjects ("participants") and go on to see data and documents associated with them such as consent forms, find the list of teammates collaborating in this study ("members"), read descriptive information on the study ("metadata"), and inspect an audit log that records work that has been done toward the completion of the study ("activity").

B. Collaboration and Knowledge Sharing

Experiments are reproducible when they are thoroughly documented and when that documentation is easily disseminated. To support this, HRIStudio includes features that enable collaborative experiment design and streamlined sharing of assets generated during experimental studies. The platform provides a dashboard that offers an overview of project status, details about collaborators, a timeline of completed and upcoming trials, and a list of pending tasks.

As previously noted, the *Data Management Layer* incorporates a role-based access control system that defines distinct user roles aligned with specific responsibilities within a study. This role structure enforces a clear separation of duties and enables fine-grained, need-to-know access to study-related information. This design supports various research scenarios, including double-blind studies where certain team members have restricted access to information. The pre-defined roles are as follows:

- *Administrator*, a "super user" who can manage the installation and the configuration of the system,
- *Researcher*, a user who can create and configure studies and experiments,
- *Observer*, a user role with read-only access, allowing inspection of experiment assets and real-time monitoring of experiment execution, and
- *Wizard*, a user role that allows one to execute an experiment.

For maximum flexibility, the system allows additional roles with different sets of permissions to be created by the administrator as needed.

The collaboration system allows multiple researchers to work together on experiment designs, review each other's work, and build shared knowledge about effective methodologies. This approach also enables the packaging and dissemination of complete study materials, including experimental designs, configuration parameters, collected data, and analysis results. By making all aspects of the research process shareable, HRIStudio facilitates replication studies and meta-analyses, enhancing the cumulative nature of scientific knowledge in HRI.

C. *Visual Experiment Design*

HRIStudio implements an *Experiment Development Environment* (EDE) that builds on Porfirio et al.'s [8] concept of Application Development Environment. Figure 4 shows how this EDE is implemented as a visual programming, drag-and-drop canvas for sequencing steps and actions. In this example, we see a progression of steps ("Welcome" and "Robot Approach") where each step is customized with specific actions. Robot actions issue abstract commands, which are then translated into platform-specific concrete commands by components known as *plugins*, which are tailored to each type of robot and discussed later in this section.

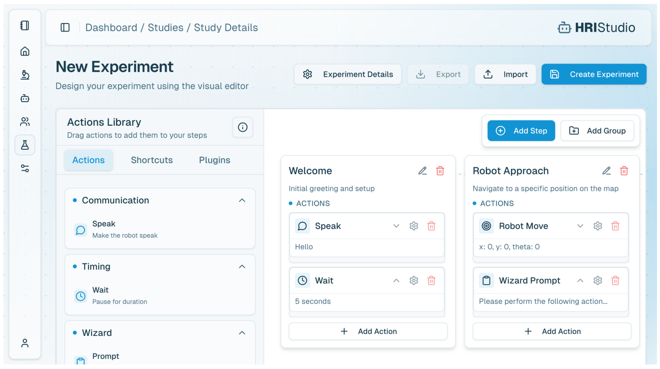


Fig. 4. View of experiment designer.

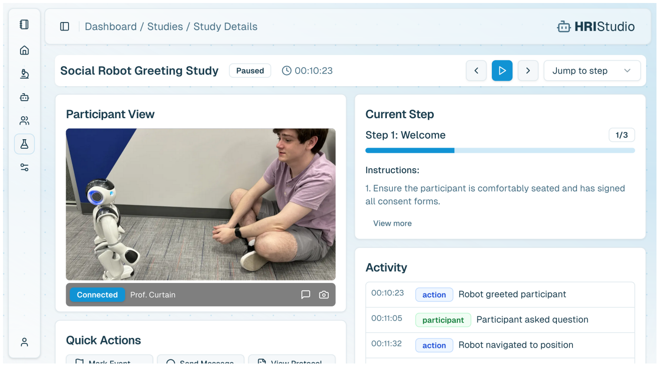
Our EDE was inspired by Choregraphe [9] which enables researchers without coding expertise to build the steps and actions of an experiment visually as flow diagrams. As in Choreograph, the design of an experiment is expressed *graphically* through the selection and the interconnection of components from a library, The robot control components presented in the user interface are automatically added to the inventory of options according to the experiment configuration, which specifies the robot to be used. We expect that this will make experiment design more accessible to those with reduced programming skills while maintaining the expressivity required for sophisticated studies. Conversely, to support those without knowledge of best practices for WoZ studies, the EDE offers contextual help and documentation as guidance for one to stay on the right track.

D. *The Wizard Interface and Experiment Execution*

We built into HRIStudio an interface for the wizard to execute experiments and to interact with them in real time. It is important to point out that our tool in no way obviates the need for the role of the wizard, who remains the *"human-in-the-loop."* The wizard continues to be essential to the execution of the experiment: their role is to advance one step at a time the experimental sequence of events expressed in a script and to intervene in case the subject of an experiment presents an unexpected reaction. In that event, the wizard can interrupt the execution of the experimental script and use the system to carry out and to record the *ad hoc* interaction.

In the development of this component, we drew on lessons from Pettersson and Wik's [6] work on WoZ tool longevity. From them we have learned that a significant factor that determines the short lifespan of WoZ tools is the trap of a fixed, one-size-fits-all wizard interface. Following the principle incorporated into their Ozlab, we have incorporated into our framework functionality that allows the wizard interface to be adapted to the specific needs of each experiment. One can configure wizard controls and visualizations for their specific study, while keeping other elements of the framework unchanged.

Figure 5 shows the wizard interface for the fictional experiment "Initial Greeting Protocol." This view shows the current step with an instruction for the wizard that corresponds to an action they will carry out. These instructions are presented one at a time so as not to overwhelm the wizard, but one can also use the "View More" button when it becomes desirable to see the complete experimental script. The view also includes a window for the captured video feed showing the robot and the participant, a timestamped log of recent events, and various interaction controls for unscripted actions that can be applied in real time ("quick actions"). By following the instructions which are provided incrementally, the wizard is guided to execute the experimental procedure consistently across its different trials with different participants. To provide live monitoring functionalities to users in the role of *observer*, a similar view is presented to them without the controls that might interfere with the execution of an experiment.



1. Ensure the participant is comfortably seated and has signed all consent forms.

```
00:10:23  action       Robot greeted participant
00:11:05  participant  Participant asked question
00:11:32  action       Robot navigated to position
```

Fig. 5. View of HRIStudio's wizard interface during experiment execution.

When a wizard initiates an action during a trial, the system executes a three-step process to implement the command. First, it translates the high-level action into specific API calls as defined by the relevant plugin, converting abstract experimental actions into concrete robot instructions. Next, the system routes these calls to the robot's control system through the appropriate communication channels. Finally, it processes any feedback received from the robot, logs this information in the experimental record, and updates the experiment state accordingly to reflect the current situation. This process ensures reliable communication between the wizard interface and the physical robot while maintaining comprehensive records of all interactions.

E. *Robot Platform Integration*

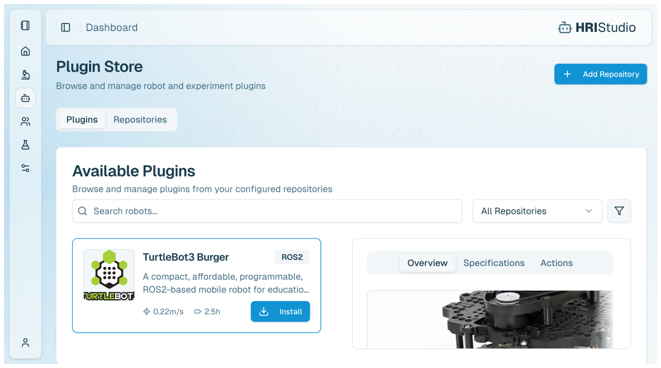
The three-step process described above relies on a modular, two-tier system for communication between HRIStudio and each specific robot platform. The EDE offers an experiment designer a number of pre-defined action components representing common tasks and behaviors such as robot movements, speech synthesis, and sensor controls. Although these components can accept parameters for the configuration of each action, they exist at a higher level of abstraction. When actions are executed, the system translates these abstractions so that they match the commands accepted by the robot selected for the experiment. This translation is achieved by a *plugin* for the specific robot, which serves as the communication channel between HRIStudio and the physical robots.

Each robot plugin contains detailed action definitions with multiple components: action identifiers and metadata such as title, description, and a graphical icon to be presented in the EDE. Additionally, the plugin is programmed with parameter schemas including data types, validation rules, and default values to ensure proper configuration. For robots running ROS2, we support mappings that connect HRIStudio to the robot middleware. This integration approach ensures that HRIStudio can be used with any robot for which a plugin has been built.

As shown in Figure 6, we have developed a *Plugin Store* to aggregate plugins available for an HRIStudio installation. Currently, it includes a plugin specifically for the TurtleBot3 Burger (illustrated in the figure) as well as a template to support the creation of additional plugins for other robots. Over time, we anticipate that the Plugin Store will expand to include a broader range of plugins, supporting robots of diverse types. In order to let users of the platform know what to expect of the plugins in the store, we have defined three different trust levels:

- *Official* plugins will have been created and tested by HRIStudio developers.
- *Verified* plugins will have different provenance, but will have undergone a validation process.
- *Community* plugins will have been developed by third-parties but will not yet have been validated.

The Plugin Store provides access to the source version control *repositories* which are used in the development of plugins allowing for the precise tracking of which plugin versions are used in each experiment. This system enables community contributions while maintaining reproducibility by documenting exactly which plugin versions were used for any given experiment.



Browse and manage robot and experiment plugins

Browse and manage plugins from your configured repositories

A compact, affordable, programmable, ROS2-based mobile robot for educatio...

Fig. 6. The Plugin Store for plugin selection.

F. *Comprehensive Data Capture and Analysis*

We have designed HRIStudio to create detailed logs of experiment executions and to capture and place in persistent storage all the data generated during each trial. The system keeps timestamped records of all executed actions and experimental events so that it is able to create an accurate timeline of the study. It collects robot sensor data including position, orientation, and various sensor readings that provide context about the robot's state throughout the experiment.

The platform records audio and video of interactions between a robot and participant, enabling post-hoc analysis of verbal and non-verbal behaviors. The system also records wizard decisions and interventions, including any unplanned actions that deviate from the experimental protocol. Finally, it saves with the experiment the observer notes and annotations, capturing qualitative insights from researchers monitoring the study. Together, these synchronized data streams provide a complete record of experimental sessions.

Experimental data is stored in structured formats to support long-term preservation and seamless integration with analysis tools. Sensitive participant data is encrypted at the database level to safeguard participant privacy while retaining comprehensive records for research use. To facilitate analysis, the platform allows trials to be studied with "playback" functionalities that allow one to review the steps in a trial and to annotate any significant events identified.

VI. Conclusion and Future Directions

Although Wizard-of-Oz (WoZ) experiments are a powerful method for developing human-robot interaction applications, they demand careful attention to procedural details. Trials involving different participants require wizards to consistently execute the same sequence of events, accurately log any deviations from the prescribed script, and systematically manage all assets associated with each participant. The reproducibility of WoZ experiments depends on the thoroughness of their documentation and the ease with which their experimental setup can be disseminated.

To support these efforts, we drew on both existing literature and our own experience to develop HRIStudio, a modular platform designed to ease the burden on wizards while enhancing the reproducibility of experiments. HRIStudio maintains detailed records of experimental designs and results, facilitating dissemination and helping third parties interested in replication. The platform offers a hierarchical framework for experiment design and a visual programming interface for specifying sequences of events. By minimizing the need for programming expertise, it lowers the barrier to entry and broadens access to WoZ experimentation.

HRIStudio is built using a variety of web application and database technologies. The primary programming language we use is JavaScript with the support of libraries such as *React*, *React Flow*, and *Auth.js*. Additionally, we leverage *MinIO/S3* for distributed object storage, *tRPC* for typesafe APIs between client and server, and *Postgres* for database implementation. To simplify deployment and reduce system dependencies introduced by relying on multiple technologies, we are containerizing the platform and developing comprehensive, interface-integrated documentation to guide users through installation and operation.

Our next development phase focuses on enhancing execution and analysis capabilities, including advanced wizard guidance, dynamic adaptation, and improved real-time feedback. We are also implementing playback functionality for reviewing synchronized data streams and expanding integration with hardware commonly used HRI research. An advanced stage in this project will be to conduct user studies to evaluate the assumptions we made in the design and in the implementation of the system.

Ongoing engagement with the research community has played a key role in shaping HRIStudio. Feedback from the reviewers of our RO-MAN 2024 late breaking report and conference participants directly influenced our design choices, particularly around integration with existing research infrastructures and workflows. We look forward to creating more systematic opportunities to engage researchers to guide and refine our development as we prepare for an open beta release.

REFERENCES

[1] K. Belhassein, G. Buisan, A. Clodic, and R. Alami. Towards methodological principles for user studies in Human-Robot interaction. In *Test Methods and Metrics for Effective HRI in Collaborative Human-Robot Teams Workshop, ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, Daegu, South Korea, March 2019.

[2] Marlena R. Fraune, I. Leite, N. Karatas, A. Amirova, A. Legeleux, A. Sandygulova, A. Neerincx, G. Dilip Tikas, H. Gunes, M. Mohan, N. I. Abbasi, S. Shenoy, B. Scassellati, E.J. de Visser, and T. Komatsu. Lessons learned about designing and conducting studies from HRI experts. *Frontiers in Robotics and AI*, 8, 2022.

[3] Guy Hoffman. OpenWoZ: A runtime-configurable Wizard-of-Oz framework for Human-Robot interaction. In *Proceedings of the 2016 AAAI Spring Symposium*. AAAI, 2016.

[4] David V. Lu and W.D. Smart. Polonius: A Wizard of Oz interface for HRI experiments. In *Proceedings of the 6th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, 2011.

[5] Sean O'Connor and L.F. Perrone. HRIStudio: A framework for Wizard-of-Oz experiments in Human-Robot interaction studies. In *33rd IEEE International Conference on Robot and Human Interactive Communication (RO-MAN) (late breaking report)*, Pasadena, CA, USA, August 2024.

[6] John Sören Pettersson and M. Wik. The longevity of general purpose Wizard-of-Oz tools. In *Proceedings of the Annual Meeting of the Australian Special Interest Group for Computer Human Interaction (OzCHI '15)*, New York, NY, USA, 2015. ACM.

[7] Martin Porcheron, J.E. Fischer, and M. Valstar. NottReal: A tool for voice-based Wizard of Oz studies. In *Proceedings of the 2nd Conference on Conversational User Interfaces*, CUI '20, New York, NY, USA, 2020. ACM.

[8] David Porfirio, M. Roberts, and L.M. Hiatt. Guidelines for a Human-Robot Interaction specification language. In *Proceedings of the 32nd IEEE International Conference on Robot and Human Interactive Communication (RO-MAN)*, 2023.

[9] E. Pot, J. Monceaux, R. Gelin, and B. Maisonnier. Choregraphe: A graphical tool for humanoid robot programming. In *Proceedings of the 18th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)*, 2009.

[10] Laurel D. Riek. Wizard of Oz studies in HRI: A systematic review and new reporting guidelines. *Journal of Human-Robot Interaction*, 1(1):119–136, July 2012.

[11] Finn Rietz, A. Sutherland, S. Bensch, S. Wermter, and T. Hellström. WoZ4U: An open-source Wizard-of-Oz interface for easy, efficient and robust HRI experiments. *Frontiers in Robotics and AI*, 8, 2021.

[12] Aaron Steinfeld, O. C. Jenkins, and B. Scassellati. The Oz of Wizard: Simulating the human for interaction research. In *Proceedings of the 4th ACM/IEEE International Conference on Human Robot Interaction (HRI)*, La Jolla, California, USA, 2009. ACM.