UNIVERSITÀ DEGLI STUDI DI BRESCIA

DIPARTIMENTO DI INGEGNERIA DELL'INFORMAZIONE

DOTTORATO DI RICERCA IN TECHNOLOGY FOR HEALTH

SSD: ING-INF/05

CICLO XXXVIII

HUMAN-ROBOT COLLABORATION IN HEALTHCARE: NEW PROGRAMMING AND INTERACTION TECHNIQUES

Dottorando:
Luigi Gargioni
Matricola 89467

Supervisore: Prof.ssa Daniela Fogli, Università degli Studi di Brescia
Co-Supervisore: Prof. Pietro Baroni, Università degli Studi di Brescia

Finanziato dall'Unione Europea
NextGenerationEU

Abstract

The integration of robotic systems into complex, human-centric environments, such as healthcare, necessitates a paradigm shift towards truly collaborative and intuitive interaction. This thesis presents a comprehensive research project dedicated to advancing Human-Robot Interaction (HRI) by addressing two critical limitations: the difficulty that domain experts face in defining robot tasks, and the lack of robot adaptability during human-robot collaborative execution.

The first major contribution is framed in the End-User Development (EUD) research area. I propose and evaluate multimodal environments that enable domain experts, specifically healthcare professionals, to define, modify, and validate robot tasks using natural language and graphic interfaces. This approach leverages Large Language Models (LLMs) to translate high-level user goals into robot task representations, significantly lowering the entry barrier for robot programming. The approach is demonstrated through the creation of PRAISE (Pharmaceutical Robotic and AI System for End Users), a domain-specific EUD environment for pharmaceutical compounding, and further extended to ADMIRABLE (A Domain-independent Multimodal Interface for Robot Adaptable Behavior Leveraging End-user development), a domain-independent system that includes a Digital Twin of the robot for simulation and safety validation.

The second core contribution is an adaptive Human-Robot Collaboration (HRC) architecture. I propose a hybrid architecture that combines deterministic, model-based reasoning with the generative and context-aware capabilities of LLMs. By integrating explicit representations of collaborative tasks and situational awareness, the architecture enables the robot to continuously monitor the environment, anticipate human actions, and detect deviations from planned workflows. This combination of structured and flexible reasoning allows the robot to generate adaptive, context-sensitive guidance and collaborative support actions, enhancing trust, efficiency, and fluency in dynamic human-robot interactions.

Together, these contributions enable more intuitive task definition and validation, and more adaptive and context-sensitive collaboration between humans and robots, supporting flexible and human-centered robotic assistants for real-world deployment.

Keywords
Human-Computer Interaction, Human-Robot Collaboration, End-User Development, Large Language Models

Sommario

L'integrazione dei sistemi robotici in contesti reali e centrati sulla persona, come quelli sanitari, richiede un cambio di prospettiva verso un'interazione più collaborativa e intuitiva fra umano e robot. Questa tesi presenta un progetto di ricerca volto a far progredire la Human-Robot Interaction (HRI), affrontando due limiti principali: la difficoltà per gli esperti del dominio nel definire i compiti da affidare al robot e la scarsa capacità dei robot di adattare il loro comportamento durante l'esecuzione di attività collaborative con gli esseri umani.

Il primo contributo si colloca nell'ambito dell'End-User Development (EUD). In questa prima parte vengono sviluppati e valutati degli ambienti multimodali che permettono agli esperti del dominio, in particolare ai professionisti sanitari, di definire, modificare e validare compiti robotici usando il linguaggio naturale ed interfacce grafiche dedicate. Questo approccio sfrutta i Large Language Model (LLM) per tradurre gli obiettivi espressi dagli utenti in rappresentazioni operative del compito, riducendo in modo significativo la barriera d'ingresso della programmazione robotica. L'approccio è stato validato attraverso lo sviluppo di PRAISE (Pharmaceutical Robotic and AI System for End Users), un ambiente EUD pensato per le preparazioni farmaceutiche, e successivamente esteso ad ADMIRABLE (A Domain-independent Multimodal Interface for Robot Adaptable Behavior Leveraging End-user development), un sistema svincolato dal contesto applicativo, ma al contempo adattabile ad esso da parte dell'utente, dotato di un gemello digitale del robot per simulare e validare in sicurezza i compiti definiti.

Il secondo contributo è un'architettura adattiva per la Human-Robot Collaboration (HRC). Viene proposta un'architettura ibrida che integra il ragionamento deterministico basato su modelli simbolici con le capacità generative e di interpretazione contestuale degli LLM. Combinando rappresentazioni esplicite di compiti collaborativi con la comprensione del contesto, l'architettura consente al robot di monitorare continuamente l'ambiente, anticipare le azioni umane e rilevare eventuali deviazioni dai flussi di lavoro previsti. L'integrazione fra ragionamento strutturato e ragionamento flessibile permette al robot di fornire indicazioni ed interventi contestuali, migliorando la fiducia, l'efficienza e la fluidità dell'interazione.

Nel complesso, questi contributi rendono più naturale la definizione e la validazione dei compiti robotici e favoriscono una collaborazione tra persone e robot più adattabile, adattiva e consapevole del contesto. Il lavoro punta così allo sviluppo di assistenti robotici flessibili, realmente orientati all'uomo e pronti per essere impiegati nel mondo reale.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

This thesis illustrates the study and work carried out at the Information Engineering department of the University of Brescia during my Ph.D. in Technology for Health, cohort XXXVIII. My work was done under the supervision of Prof. Daniela Fogli and Prof. Pietro Baroni at the University of Brescia. During my research visit period at LAAS-CNRS in Toulouse (France), I worked under the supervision of Dr. Rachid Alami in the Robotics and InteractionS (RIS) team.

The PhD scholarship is co-funded by the Italian Ministry, Piano Nazionale di Ripresa e Resilienza (PNRR), and Antares Vision S.p.A.

1.1 Context

The field of robotics is currently undergoing rapid expansion, driven by significant technological advancements across multiple disciplines. In industrial automation, innovations in control systems, materials, and technical standards are transforming manufacturing processes. Robots in this domain are capable of executing highly precise and repeatable operations at high speed over long periods, making them ideal for tasks such as precision welding, heavy load handling, and complex assembly line operations. In parallel, the rise of collaborative robots (cobots) is reshaping the way humans and robots interact. Designed to operate safely alongside human workers without the need for physical barriers, cobots are enabling more flexible, adaptive, and human-centric manufacturing environments.

Furthermore, recent advances in Artificial Intelligence (AI) have introduced a new wave of computational capabilities to mobile robotics [144], including machine learning, automated planning, and computer vision. These developments have enabled mobile robots to navigate both known and unknown environments autonomously. Applications include hazardous area exploration, such as radioactive zones, and object recognition in sensitive contexts like airports.

Another area undergoing significant transformation is social robotics [99]. This is largely driven by the advent of Large Language Models (LLMs), which have acquired conversational abilities far beyond traditional systems. Social robots are now capable of engaging in human-like interactions aimed at acting as a conversational partner, supporting mental well-being, and facilitating cognitive exercises.

While the robotics field has shown impressive progress and earned substantial attention, challenges persist, particularly in the domains of collaborative and social robotics, where interaction with humans remains a central aspect. Human-Robot Interaction (HRI) occurs at multiple levels: from task definition to real-time task execution and adaptation. These interactions often require a robot to understand and respond to human actions, preferences, and emotional states, for which traditional model-based robotic systems are not well-suited.

In collaborative robotics, one key challenge lies in enabling domain experts to define the tasks that robots must perform. Given the increasing versatility of collaborative robots, such systems must be adaptable to frequent changes in production workflows. This demand aligns with the principles of End-User Development (EUD) [94, 12], which promotes empowering end users to modify or extend technological artefacts. In this context, end-user robot programming has emerged as a growing field of research, aiming to make robotic systems programmable by users without formal expertise in robotics. Two main approaches have been identified in end-user robot programming: (i) programming by demonstration [97], which requires multiple iterations for skill generalization, and (ii) visual programming interfaces [130], designed to lower the cognitive and technical barriers for users. In parallel, the ubiquity of natural language in human communication has inspired efforts to enable natural language programming for robots. This paradigm has been applied to tasks such as navigation, assembly, and manipulation in industrial contexts [106], and is now being explored for social and service robotics [48].

Although HRI shares many objectives with Human-Computer Interaction (HCI), some of the main challenges in building a human-aware collaboration with robots come from the field of autonomous robotics. These challenges often involve how robots make decisions and interpret their surroundings, areas traditionally explored through research on autonomous behavior and artificial cognition. Several studies have addressed the architectural and cognitive challenges involved in designing adaptive and interactive robotic systems [145]. These systems typically integrate decisional components responsible for high-level reasoning, such as planning and situation assessment, with functional components that manage real-time perception and action [46]. Effective integration of these elements is essential to allow robots to interact meaningfully with dynamic environments and human partners. Despite this progress, current architectures often depend on predefined models of human mental states, like beliefs, desires, and intentions, limiting their ability to adapt to unpredictable behavior. This model-based limitation introduces uncertainty and hinders the robot's capacity to anticipate or respond to nuanced human actions. Building accurate, generalizable models of human cognition remains an open challenge. Moreover, the growing complexity and diversity of potential interaction scenarios amplify the difficulty of defining universal metrics for the explainability and acceptability of robot plans.

In this context, my PhD research is situated within the field of HRI. It focuses on two complementary aspects: on one hand, (i) enhancing task definition mechanisms, particularly through EUD methodologies and natural language interfaces; on the other hand, (ii) improving collaboration during task execution by enabling robots to adapt in real time to human behavior, preferences, and environmental changes.

1.2 Research Objectives and Contributions

Building on the themes discussed above, this thesis is structured around two main research objectives. These objectives are designed to address current limitations in the field of human-robot interaction, particularly in relation to end-user involvement and human-aware task execution. *Objective 1* was explored in collaboration with Prof. Daniela Fogli and Prof. Pietro Baroni at the University of Brescia. This phase of the research focused on enabling domain experts to define robot tasks without requiring technical expertise. *Objective 2* was developed during a research visit period at LAAS-CNRS in Toulouse (France), under the supervision of Dr. Rachid Alami, and with continued support from Prof. Daniela Fogli. This part of the work concentrated on investigating different types of models of human behavior and intentions in robotic planning and execution.

1.2.1 Objective 1: End-User Development for Robot Task Definition

This objective aimed to explore and support the ability of end users to autonomously define, edit, and verify robotic tasks. This encompassed not only the specification of task logic but also the identification and modeling of relevant domain items of the real-world environment. This objective is based on the idea that giving end users more control can make robotic systems easier to use and more adaptable to different and dynamic situations. Specifically, the goal is to investigate design approaches and interaction paradigms that facilitate such empowerment, including natural language interfaces and multimodal interaction techniques. Key aspects of interest include:

1. Supporting the definition of both tasks and domain elements directly by end users, reducing dependency on technical developers.
2. Exploring the viability and limitations of natural language as an approach for task specification.
3. Evaluating the effectiveness of multimodal interaction in enhancing the end-user experience across the different stages of the task definition workflow.

Research Questions

To guide this investigation, the following research questions were formulated:

- **RQ1**: To what extent can domain experts define domain-specific tasks and items through dedicated user interfaces designed for non-programmers?
- **RQ2**: What are the capabilities and limitations of natural language interfaces in supporting users to accurately specify, validate, and refine structured robotic tasks, and how can multimodal interaction enhance this process during task definition?

Methodology

To address the research questions, several interaction design activities were carried out. Whenever feasible, the investigations focused on use cases in the healthcare domain, particularly on galenic formulations, which are personalized medications manually prepared by pharmacists to meet specific patient needs, such as allergies, tailored dosages, or specific administration forms.

The healthcare domain was selected as a reference context since this PhD project is part of the *Technology for Health* doctoral program at the University of Brescia. The interaction design activities revolved around two main projects:

- **Project 1 - Preparation of Personalized Medicines through Collaborative Robots**: In [53], a survey was conducted to investigate how medications are organized and dispensed in pharmacies and patients' homes, with a specific focus on the use of pill dispensers. This initial analysis offered a contextual analysis of pharmacy workflows and provided the opportunity to explore the feasibility of a future end-to-end service for galenic preparation and delivery. Subsequently, the next step of this activity [55] focused on applying a human-centred methodology to design an EUD environment by involving representative end users (experts in the pharmaceutical sector) from system ideation to its evaluation. The designed EUD environment is based on the starting point of the PhD project, namely the CAPIRCI application [47]. CAPIRCI supports non-technical users in defining robot tasks, introducing the possibility to verify and modify natural language-generated programs using a block-based interface, thereby providing the conceptual basis for the subsequent exploration of how such approaches can be adapted to the pharmaceutical context. The analysis of pharmacy workflows identified repetitive and low-value tasks that could be effectively delegated to collaborative robots, allowing pharmacists to focus on higher-priority activities. The primary conclusions of the work pertain to the design implications associated with AI-enabled EUD for collaborative robots. Based on these findings, in [51], low-fidelity prototypes of an interactive system were developed. Following the emerged implications and the designed mockups, a web-based prototype has been developed [57]. The purpose of the application is to provide support to end users (i.e., pharmacists) in defining robot programs that are suitable for the specific case of galenic preparations. The application is conceived as an EUD environment, which implements a multimodal interaction approach based on a natural language interface leveraging LLMs and a domain-oriented graphical interface to check and revise the robot programs created. A user study conducted with nine pharmacists has demonstrated the validity of the approach with positive feedback.

- **Project 2 - Natural Language and Multimodal Interfaces for End-User Robot Programming**: Building on the insights gained from the design and development of the previous system for galenic formulation preparations, a new domain-independent EUD environment was introduced. The goal of this work was to begin an exploration of the capabilities of the new technologies based on LLMs in identifying user intents and structuring the desired output. Extending the previous work, [50] presented a prototype environment that allows users with no technical background to define domain-specific elements (objects, actions, locations) and create pick-and-place tasks using a combination of natural language and graphical programming via Blockly. The goal was to assess the feasibility of multimodal interaction in defining and revising robot tasks. As a continuation of the previous work, a new prototype integrated a digital twin into the application. Once a task is defined, it can be simulated virtually, allowing users to preview the robot's behavior and identify potential execution issues before deployment. A user study conducted with twelve users confirmed that, with this EUD environment, the end user can define a robot task with a multimodal approach without writing code and verify it by observing its execution in a safe and controlled environment before running it on the actual robot.

Main Findings

These projects advance the field of end-user robot programming by introducing a human-centered AI approach that integrates LLMs, multimodal interaction, and digital twins. The resulting workflow empowers domain experts to conceptualize, validate, and refine robot tasks through intuitive, accessible tools, without requiring programming or robotics expertise.

Central to the methodology is the active involvement of end users throughout the design process, ensuring that the environment aligns with their practices, expectations, and cognitive models.

The activities carried out aim to demonstrate that, when properly supported, non-technical users can effectively engage in robot task definition, ultimately enabling more adaptive, usable, and trustworthy robotic systems in real-world environments.

1.2.2 Objective 2: Human-Robot Collaboration for Task Accomplishment

The second objective of the PhD project was to investigate how a robot can effectively collaborate with a human partner by taking into account human beliefs, intentions, and emotional states during task execution. This goal addresses the limitations of rigid model-based approaches, which often fall short when facing the complexity and unpredictability of real-world human behavior.

Key challenges addressed in this objective include:

1. Overcoming the limitations of traditional model-based approaches to robotics when interacting with humans, whose behavior is often non-deterministic and emotionally driven.

2. Maintaining a robust and rigorous task planning framework, even in dynamic and uncertain environments.

Research Questions

To guide this investigation, the following research question was formulated:

- **RQ3**: How can robots adapt to the unpredictable behavior of human partners in collaborative scenarios, while maintaining task control and ensuring safety during execution?

Methodology

To address RQ3, a hybrid system architecture was designed that integrates deterministic techniques (e.g., task planning and rule-based approach) with the flexibility of LLMs. The preliminary conceptualization was introduced in [54], and a complete description of the architecture, along with a user study, is provided in a paper currently under review in the *International Journal of Social Robotics*.

The approach is based on the concept of *shared goals* between the human and the robot. A task is defined as a cooperative activity aimed at achieving a mutually agreed-upon goal. These tasks may vary in complexity, from well-defined, rule-based sequences to more interactive and adaptive activities requiring continuous real-time assessment.

The assumed application context is a coaching scenario where the robot assists a person in completing a task composed of several activities. To illustrate this scenario, it can be considered a patient in a care facility who is required to take medications and engage in both cognitive and physical activities. Effective collaboration requires adapting the robot's actions not only to the requirements of the task but also to the human's evolving behavior, preferences, and affective state.

The proposed architecture is modular and designed to ensure seamless collaboration, adaptive behavior, and safe execution across the different components. The integrated pipeline ensures that task execution remains safe, adaptable, and centered on human-robot mutual understanding.

Main Findings

The developed architecture contributes to the advancement of HRI by blending deterministic techniques with the contextual flexibility offered by LLMs. The system is capable of adapting robot behavior in real time, maintaining alignment with human intentions and adjusting to unexpected changes in the environment or in user input.

A user study with 14 participants confirmed the system's capacity to dynamically adapt execution strategies in response to ongoing interaction and ensure safety and reliability despite behavioral uncertainty on the human side.

By addressing both the need for structured reasoning and the unpredictability inherent in human collaboration, this work moves toward more natural, effective, and human-centered robotic systems.

1.3 Structure of the Thesis

The thesis is organized into eight chapters, beginning with Chapter 1, which introduces the research context, defines the two main objectives (*"End-User Development for Robot Task Definition"* and *"Human-Robot Collaboration for Task Accomplishment"*), presents the research questions, and outlines the core contributions.

The core of the thesis is divided into two main parts:

1. Part I focuses on End-User Development: Chapter 2 reviews the state-of-the-art in robot programming for end users, with an emphasis on NLP and multimodal approaches; Chapter 3 describes the initial CAPIRCI platform and its early natural language/graphic interface; Chapter 4 documents the design, implementation, and user evaluation of PRAISE, a domain-specific EUD environment for compounding tasks in the healthcare domain; and Chapter 5 generalizes these EUD concepts by presenting ADMIRABLE, a domain-independent multimodal approach that integrates an LLM, a drag-and-drop graphic interface and a Digital Twin for virtual validation.

2. Part II addresses Human-Robot Collaboration: Chapter 6 explores relevant HRI concepts, including social robots and human-aware task planning; Chapter 7 details the novel hybrid architecture developed for adaptive robot coaching.

Finally, Chapter 8 provides concluding remarks, summarizing the main results, proposing future research directions, and reporting two related projects.

Part I

End-User Development for Robot Task Definition

Chapter 2

Approaches to Defining Robotic Tasks: The State of the Art

2.1 Collaborative Robotics

Collaborative robots [34], also known as cobots, represent a paradigm shift from traditional industrial automation. Unlike conventional industrial robots, typically heavy, fast, and confined within safety fences, collaborative robots are explicitly designed to share workspaces and physically interact with humans. Cobots are characterized by reduced payloads and speeds, lightweight and rounded mechanical structures, and the integration of force and torque sensors, which together ensure compliance with international safety standards such as ISO 10218[1,2] and ISO/TS 15066[3]. These ISO standards formally define modes of safe collaboration, including power and force limiting, speed monitoring, and safety stops.

From an operational perspective, cobots trade extreme speed and payload capacity for ease of use and installation, safety, and reconfigurability. These characteristics have facilitated their deployment in a wide range of industries [138, 96], including automotive [121], packaging [78], and healthcare [2]. According to the literature review presented by Keshvarparast et al. [83], common applications include assembly [7], loading/unloading of components [84], and ergonomically challenging tasks that benefit from human oversight and flexibility [104]. Cobots feature a relatively low cost of deployment and flexibility that make them particularly attractive to small and medium-sized enterprises (SMEs), which may lack the resources for large-scale automation infrastructures [77].

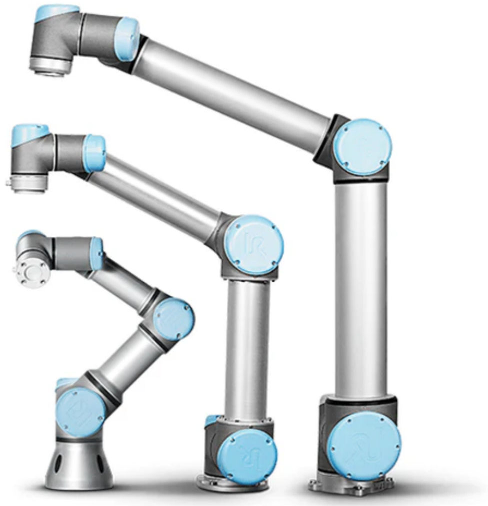
A growing volume of research emphasizes the contribution of collaborative robotics to sustainable manufacturing. By enabling more flexible and adaptable production processes [59], cobots support more efficient operations, reducing material waste and energy consumption [65], and lower ergonomic risks for workers [30].

Several manufacturers have developed collaborative robots. Figure 2.1 illustrates representative examples, each demonstrating different design philosophies and target applications within the collaborative robotics landscape.

The Denso Cobotta[4] (Figure 2.1a) is a compact 6-axis collaborative robot designed specifically for light assembly tasks and educational applications, featuring integrated

[1] ISO 10218, Part 1 - https://www.iso.org/standard/73933.html
[2] ISO 10218, Part 2 - https://www.iso.org/standard/73934.html
[3] ISO/TS 15066 - https://www.iso.org/standard/62996.html
[4] Denso Cobotta - https://www.densorobotics.com/products/collaborative/cobotta

(a) Denso Cobotta



(b) Universal Robots UR Series

(c) ABB YuMi

(d) KUKA iiwa

Figure 2.1: Overview of leading collaborative robot platforms representing different approaches to collaborative robotics, from compact educational platforms to advanced dual-arm systems for precision assembly

vision systems and intuitive programming interfaces. Universal Robots (UR) has established itself as a market leader with their UR series[5] (Figure 2.1b), including models like the UR3, UR20, and UR30, which offer different payload capacities. The UR series spans a wide range of payload capabilities, from small-scale applications with the UR3 (approximately 3 kg) to heavier collaborative tasks with models such as the UR20 and UR30, offering payloads of approximately 20 kg and 30 kg, respectively. This range covers most light- to medium-duty needs. ABB's YuMi[6] (Figure 2.1c) is a dual-arm collaborative robot designed for small parts assembly, featuring inherent safety through its lightweight design and advanced control systems. KUKA's iiwa (intelligent industrial work assistant)[7] (Figure 2.1d) represents another significant advancement in collaborative robotics, incorporating sensitive robotics technology that enables the robot to feel and react to forces, making it suitable for delicate assembly tasks and direct human-robot collaboration.

Cobots are designed to work alongside humans and offer greater flexibility in deployment. To fully exploit this flexibility, programming these robots must be accessible to end users. Intuitive interfaces must allow operators without coding or robotics knowledge to define cobot tasks and adapt workflows quickly. Making cobots programming accessible not only reduces downtime and integration costs but also expands the range of applications where automation can be effectively applied.

However, the challenge of programming remains a critical barrier. While collaborative robots are designed with usability in mind, they still inherit many of the limitations of traditional robot programming paradigms, which rely heavily on specialized languages and technical expertise. This makes it necessary to examine how different kinds of robots (e.g., industrial robots, cobots, drones, etc.) are currently programmed and what constraints hinder broader accessibility.

2.2 Robot Programming Outside the Academia

Today's industrial robots are typically programmed by specialists using either online (e.g., through a teach pendant) or offline (e.g., using a simulator to program and test the robot virtually before uploading the code to the real robot) methods. In practice, a robot integrator often defines a sequence of motions and actions in a vendor-specific language, then downloads it to the robot controller [23]. Major robot brands provide proprietary scripting languages (e.g., Denso's PacScript, Universal Robot's URScript, ABB's RAPID, KUKA's KRL, Fanuc's Karel, Yaskawa's INFORM, etc.), and these languages differ in syntax and capabilities. Programming a robot typically requires deep technical knowledge: programmers must understand the robot's kinematics, coordinate frames, and control modes, and write or fine-tune code in the manufacturer's interface. Any change to the robot's task often entails interrupting the program, reteaching waypoints or rewriting code, then resimulating and testing. The reliance on specialized languages and tools makes the process slow and inflexible.

For the above reasons, traditional industrial robot programming methods often pose high usage thresholds, resulting in steep learning curves and limited interoperability. For non-industrial robots (e.g., collaborative robots, service robots, drones, etc.), programming still requires software expertise: developers commonly use languages like C++ or

[5]Universal Robots - https://www.universal-robots.com/products/ur-series

[6]ABB's YuMi - https://new.abb.com/products/robotics/robots/collaborative-robots/yumi

[7]KUKA's iiwa - https://www.kuka.com/en-us/products/robotics-systems/industrial-robots/lbr-iiwa

Python to exploit vendor Software Development Kits (SDKs) or robotics frameworks. Open-source systems, such as the Robot Operating System (ROS)[8] [123], provide high-level Application Programming Interfaces (APIs) and integration features for various robot models. However, even with ROS, a programmer must handle middleware, simulation, and calibration aspects.

In summary, all current robot programming approaches demand professional expertise. The code is often vendor-locked, meaning that there is a lack of a universal robot language, and programmers must manage low-level details, like motion trajectories and safety checks. Even modern attempts to unify control (e.g., ROS-Industrial[9] [134]) acknowledge that many vendor interfaces and safety constraints remain unsupported. Thus, the need for a technical specialist remains a major bottleneck in deploying and reprogramming robots in the field.

These limitations are particularly evident in the context of collaborative robotics, where the users are domain experts rather than programmers. This has motivated the emergence of scientific research on end-user robot programming, which aims to make robot control accessible to domain experts.

2.3 End-User Robot Programming

End users of collaborative robots are usually all those humans whose activities can be supported and alleviated in terms of fatigue and repetitiveness. Such end users are not necessarily proficient in programming and robotics, nor do they want to be. Rather, they are often experts in their application domain, even though they could be afraid to be replaced by automation; in most cases, their expertise is still critical for obtaining correct and complete products and workflows, and thus the use of cobots is aimed at empowering rather than replacing these humans.

End-User Programming (EUP) methods and techniques have been proposed in the last twenty years to provide a set of techniques that enable end users to create their own programs [86]. More recently, end-user robot programming has emerged as a research area that aims to empower end users of robot technologies, who are not experts in robotics, to tailor robot behavior to their needs [3]. End-user robot programming can be much more complex than traditional EUP, as robot programs must refer to physical objects and locations, and the robot must move safely in the environment, especially when humans are in close proximity.

The survey by Ajaykumar et al. [3] distinguishes two main approaches to end-user robot programming: the former is based on programming by demonstration [116, 128], which, however, requires several trials to make the system learn and generalize a skill; the latter exploits the idea of simplifying program specification by end users.

Specification by end users of robot programs is primarily achieved in the literature through visual programming environments. Most of these environments employ computer science notations such as flowcharts [6, 58] or hierarchical trees [117, 76], thus making them difficult to manage by non-expert programmers. Recently, block-based notations provided by Google Blockly [62] and Scratch [125], with their puzzle-like blocks, proved to be suitable for novice programmers. Thus, the puzzle metaphor inspired the development of robot programming environments, like Robot Blockly [151] and EUD-MARS [4]. However,

[8]Robot Operating System (ROS) - https://www.ros.org
[9]ROS-Industrial - https://rosindustrial.org

also in these cases, programming concepts, such as variables, conditionals, and loops, must be known by end users to create a robot program from scratch. CodeFrame [130] is another system allowing users to drag blocks into a canvas to create a robot program. This system is designed to train novice cobot programmers, and requires learning low-level details of robot programming, including trajectory definition, collision avoidance, joint speed, and the like. A higher-level approach to robot programming is proposed by Schou et al. [131], where a skill-based visual language allows the user to define a sequence of high-level robot actions, like picking up an object, rotating an object, navigating to a given location, and so on. However, this approach is less flexible than the previous ones since the user can only create a sequence of actions, choosing them among those made available in the programming environment.

An intermediate approach is proposed in [126], where a hybrid visual programming environment for end users is introduced. The system combines a block-based notation, suitable for specifying imperative robot task sequences, with a data-flow graph representation, which is more appropriate for expressing complex and nested conditions. In a controlled study with 113 participants, this hybrid approach was compared with a purely block-based variant. Although participants expressed a subjective preference for the first one, quantitative results indicated higher performance in terms of correctness and usability with the purely block-based interface. The study also highlighted some limitations: the hybrid environment introduced higher cognitive complexity and reduced usability in the graph-based component, with participants achieving less accurate results and lower overall success rates.

Recently, reinforcement learning (RL) has been explored as a way to enable end users to program robots. Ayalew et al. [10] presented two RL-based programming paradigms: full Markov Decision Process (MDP) Programming and Goal-Only MDP Programming (Goal-MDP). In an online study with 409 participants, Goal-MDP allowed users to create robot programs that were comparable in terms of effectiveness and user experience to traditional approaches, such as sequential or trigger-action programming, while generating significantly shorter programs. RL-based paradigms have demonstrated the potential for reducing the cognitive burden of programming by focusing on high-level goals rather than low-level instructions. However, Full-MDP turned out to be too demanding for domain experts, and even Goal-MDP suffered from misunderstandings in goal specification, limited debugging support, and scalability concerns in richer environments.

Novel approaches to robot programming are being proposed, which exploit teleoperation through a 2D screen or immersive virtual reality (VR) [155, 24] or augmented reality (AR) [74] using head-mounted displays and special controllers. VR/AR is a promising human-robot interaction interface also for end users, offering a direct mapping of the robot's arm and gripper to VR/AR hand controllers. With VR/AR, end users can simply control robots, but they may struggle to create complex robot tasks, including repetitive action sequences, conditional behaviors, and the like.

To further lower the barriers for end users and complement interfaces such as VR and teleoperation, recent research has explored the use of natural language and multimodal programming techniques, enabling end users to specify robot behaviors more intuitively.

2.4 Natural Language Processing and Multimodal Programming Techniques

The widespread use of natural language in human interactions inspired the ideation of systems supporting natural language programming, exploiting natural language processing (NLP). This paradigm has been proposed to program navigation, assembly, and manipulation tasks for industrial robots [137, 106], or interaction tasks for social and service robots [18, 28]. Large Language Models (LLMs) are currently being used for natural language understanding [150], code generation [129], and several other tasks in robotics [85]. Critical limitations of these approaches are highlighted in the literature. Specifically, Wang et al. [150] identify critical challenges, including the lack of depth perception reasoning capabilities in LLMs, dependency on carefully crafted prompts requiring domain expertise, and constraints imposed by predefined action sets that limit robotic flexibility and real-world applicability. Schenkenfelder et al. [129] highlight difficulties in correctly interpreting technical abbreviations and variable names, as well as challenges in generating appropriate visual feedback or interface elements to support robot programming. Vemprala et al. [149] explored the use of OpenAI ChatGPT [111] for robot programming based on a high-level function library. ChatGPT is in charge of parsing end users' utterances and converting them to a sequence of function calls. However, end users are required to provide feedback on the generated code, possibly suggesting modifications; thus, they must possess some programming knowledge. A similar problem occurs with the work by Karli et al. [81], which exploits an LLM to directly generate and provide the user with the Python code corresponding to the requested robot task.

After revealing the difficulties users face when expressing their requests for programming a robot using natural language alone, a multimodal approach combining natural language with touch-screen sketching was proposed by Porfirio et al. [122]. The resulting system, called Tabula, allows users to use sketching to define program logic (e.g., loops), considered difficult to express verbally. However, as highlighted in [122], Tabula does not offer feedback to the user about the generated program and, thus, does not allow them to verify its correctness.

While multimodal and natural language programming approaches have contributed to lowering the barrier for end users, they remain constrained by issues such as limited feedback, difficulties in verifying correctness, and the need for prior technical expertise. These limitations motivate the adoption of an End-User Development (EUD) approach, which aims to overcome such challenges by enabling more accessible and reliable interaction with robotic systems.

2.5 End-User Development

EUD [94, 115, 12] extends the concept of EUP by not only enabling domain experts to directly program or configure digital artifacts, but also by allowing them to shape, adapt, and build tools that suit their practices over time. Following Fischer's concept of domain-oriented design [43], EUD environments are structured around the specific practices, goals, and constraints of the professional domain, enabling domain experts to develop and tailor tools that reflect their expertise without requiring programming skills. In healthcare, this distinction is particularly relevant: clinicians, therapists, and caregivers often lack technical expertise in programming, yet they hold critical domain knowledge that must be embedded in the behavior of assistive systems. Consequently, EUD approaches in healthcare seek to empower these professionals to design and customize robotic and software solutions without relying on technical developers [100, 38, 48].

A recent literature review by Vaiani and Paternò provides a comprehensive overview of EUD in the context of HRI [148]. The analysis highlights how EUD has become an emerging field, enabling end users to personalize robotic systems across a variety of domains. The review identifies core trends, including the shift toward natural language and multimodal interaction, the integration of participatory design approaches [108], and the increasing role of artificial intelligence in lowering the barriers to robot customization. Importantly, the review emphasizes that EUD in HRI contributes not only to usability but also to long-term appropriation of robotic technologies, ensuring that end users retain agency in adapting systems to evolving practices.

Building on earlier foundational work, recent research has demonstrated the potential of EUD in several healthcare domains. A seminal work on EUD in healthcare is provided by Carmien et al. [31], who proposed socio-technical environments that enable non-technical caregivers to configure and adapt assistive systems for people with cognitive disabilities. Rather than programming, caregivers personalize task scripts and support mechanisms over time, illustrating how EUD can foster long-term appropriation and shared responsibility in care contexts. An additional early contribution to EUD in healthcare is provided by Tetteroo et al. [139], who reported lessons learned from the long-term deployment of *TagTrainer*, an EUD platform for physical rehabilitation. The system enabled therapists to create personalized rehabilitation exercises by visually configuring an interactive board rather than programming its behaviors through a programming language. Based on in-the-wild studies conducted in clinical settings, the authors highlighted key challenges for EUD adoption in healthcare, including time constraints, organizational factors, and the need for gradual appropriation, and derived design guidelines for sustainable EUD systems in professional care contexts. More recently, Manavi et al. presented a long-term participatory design study in an elder-care facility where nurses co-created a smartphone interface for customizing the behavior of social robots [100]. By adopting a no-code trigger–action model, care professionals were able to define reminders, interactive games, and social interactions for residents, thereby tailoring robot assistance to the rhythms of daily care without programming skills. Similarly, De La Rosa Gutierrez et al. introduced PLATYPUS [38], a block-based environment designed for physical therapists to program rehabilitation robots. Using puzzle-like blocks, therapists could assemble and adapt robot-assisted exercises, bridging the gap between clinical expertise and robot programming. In the same study, they also described a participatory design study that proposed a textual domain-specific language (DSL) based on Behavior-Driven Development (BDD), allowing therapists to specify robot behaviors in near-natural language scenarios. This approach lowers barriers by enabling therapists to describe robot actions in terms of conditions and expected outcomes, which the system then translates into executable tasks. A complementary effort focused on natural language interfaces for humanoid robots was carried out by Gallo et al., who presented a speech-based programming system where end users define rules by giving trigger–action instructions to the robot [48]. The system then automatically generates and executes the corresponding behaviors, making personalization accessible to caregivers and patients.

These examples highlight a convergence of design strategies aimed at embedding health-related expertise directly into technological artefacts. They show that EUD can facilitate the appropriation of robotic systems in healthcare, ensuring adaptability and long-term usability.

While the studies discussed above demonstrate the potential of EUD to support personalization and long-term appropriation of systems, they also raise the question of how to assess the quality of EUD environments beyond traditional usability criteria. Classic usability metrics primarily focus on the effectiveness and efficiency of system use, but they are insufficient to capture the specific challenges of EUD, where end users are required not only to operate a system but also to create, modify, and evolve digital artifacts over time. To address this gap, Barricelli et al. introduced the concept of *EUDability*, defined as *"the degree of concreteness, modularity, structuredness, reusability, and testability fostered by an EUD environment designed for specified end-user developers, with a specified goal to be pursued in a specified context"* [15]. EUDability provides a conceptual framework to evaluate how well an EUD environment supports end users in performing development activities that are meaningful within their domain, taking into account both the characteristics of the users and the context of use. The five dimensions of EUDability jointly capture key qualities that are particularly relevant in professional and safety-critical domains such as healthcare. *Concreteness* and *Structuredness* support understandability and reduce cognitive load, *Modularity* and *Reusability* enable incremental development and adaptation, and *Testability* allows end users to validate and gain confidence in the behaviors they define. By focusing on these dimensions, EUDability shifts the evaluation of EUD systems from a purely interaction-centered perspective toward a development-centered one, emphasizing the empowerment of domain experts as active shapers of technology rather than passive users.

Within the healthcare context, medication management represents a relevant domain for EUD, given the strong requirements for reliability, personalization, and user trust. As a preliminary investigation, the systematic review reported in [53] analyzes pill and medication dispensers, examining existing technologies and workflows from a human-centered perspective. The work mapped current solutions, identifying gaps such as limited adaptability, insufficient support for caregivers in configuring devices, and a lack of mechanisms for personalization by end users. By clarifying the limitations of current systems, it motivated the exploration of novel EUD approaches that would enable pharmacists and caregivers to directly tailor robotic assistants to medication workflows such as preparing pill dispensers with customized dosing schedules, preparing galenic formulations with patient-specific concentrations and excipients, and filling and manipulating test tubes for pharmaceutical compounding and quality control procedures. These examples highlight the complexity and specificity of medication management tasks that require flexible, user-configurable automation systems.

Overall, EUD promises to overcome the limitations identified above by shifting development power from engineers to end users. By providing high-level interfaces that are tuned to a specific domain, EUD environments can hide complexity and allow on-the-fly task redefinition. In summary, EUD offers a path to make collaborative robots truly adaptable: it extends the concept of programming to non-technical users and thereby addresses the usability gaps of traditional and end-user programming methods.

The content and approach of this part of the thesis are built upon these foundational notions of EUD, leveraging its principles to design high-level interfaces that empower end users to customize and control robotic systems according to their domain expertise.

Chapter 3

The Starting Point: CAPIRCI

This chapter introduces the CAPIRCI[1] (Chat And Program Industrial Robots through Convenient Interaction) system, which represents the starting point of the research presented in this thesis. CAPIRCI was conceived as a multimodal programming environment for collaborative robots aimed at empowering non-technical users to program robot tasks. The underlying idea was initially presented in [21] through a first proof-of-concept. Building on that early concept, the work reported in *"A hybrid approach to user-oriented programming of collaborative robots"*, published in the *Robotics and Computer–Integrated Manufacturing* journal [47], further developed CAPIRCI into an operational prototype[2], expanding its functions and validating its usability with users on a real robot. The project is based on the Denso COBOTTA robot, described in Section 2.1, which is available at the University of Brescia.

At its core, CAPIRCI embodies three main standpoints: a stepwise programming approach, a multimodal interaction mode, and a cognitive match between the user's mental model and the system's conceptual model. This chapter also discusses the limitations of the developed prototype and how these limitations motivated further developments.

3.1 Scope of the Work

The CAPIRCI project emerged from the need to address a knowledge gap in collaborative robotics, specifically regarding how humans can effectively define and adapt robot behaviors in real-world domains without requiring specialized technical skills. Building on the practical experience gathered in our Robotics Laboratory at the University of Brescia, it became evident that the interaction between humans and collaborative robots presents recurring challenges related to communication modalities, abstraction levels, and translation of domain knowledge into executable robot programs.

In this context, humans interacting with robots are the end users, who are usually domain experts, namely professionals with deep knowledge of the operational processes and tasks to be performed, but who are not experts in computer science or robotics. The CAPIRCI system is designed to enable these users to transfer their task knowledge into robot-executable instructions through intuitive interaction mechanisms.

We identified six key knowledge sources required for robot programming: the operational environment, the specific task requirements, the user's abilities and limitations, the

[1] "capirci" in Italian means "to understand each other"
[2] CAPIRCI - https://github.com/luigigargioni/capirci_legacy

robot's capabilities and constraints, the principles of effective human-robot collaboration, and the technical aspects of robot programming languages. End users typically possess comprehensive knowledge of their own abilities and understanding of the environment and tasks, but have limited familiarity with robotic capabilities, human-robot collaboration principles, and programming formalisms. This asymmetry highlights the need for interaction frameworks that bridge the gap between domain expertise and robot control.

Starting from this premise, our approach is structured around three fundamental theoretical standpoints that guided the design and development of CAPIRCI.

The first standpoint, the *simple programming journey paradigm*, states that effective robot programming can be structured as a two-phase process. In the first phase, users express an initial outline of the task through informal interaction (e.g., natural language), enabling a rapid translation of their domain expertise into preliminary task definitions. The second phase focuses on reviewing and refining task definition using more structured tools that support a systematic approach to programming (e.g., a graphic interface). This paradigm emphasizes accessibility while maintaining methodological rigor, allowing end users to contribute effectively without requiring technical expertise.

The second standpoint introduces a *multimodal interaction mode* that integrates natural language dialogue with block-based programming. Natural language interaction supports the initial exploratory phase, where users can describe goals and constraints in intuitive terms. Block-based interaction is then employed for refinement, enabling users to reason about the logical and temporal structure of the task. This multimodal approach leverages the complementary strengths of intuitive communication and structured visual representation, fostering both expressiveness and precision.

The third standpoint focuses on achieving *robust cognitive matching* between user mental models and the programming interface. Since end users reason in terms of domain-relevant concepts rather than robotic parameters, the interface must abstract from low-level details, such as coordinate frames, forces, or joint parameters, and instead present operations in familiar, task-oriented terms. This alignment enables users to operate at higher levels of abstraction, while the system autonomously handles technical translation and execution.

CAPIRCI can be regarded as an experimental testbed for validating the feasibility of these theoretical standpoints, bridging the gap between conceptual hypotheses and practical implementation. The system was conceived to support the definition of tasks, objects, actions, and locations in a way that can be progressively extended, reused, and shared across different users and domains. In doing so, CAPIRCI enables a new form of incremental, domain-driven robot programming, where users can gradually build and refine their own libraries of task elements and interaction patterns.

The empirical validation through controlled user studies and workflow analysis aimed to assess whether this approach can effectively empower non-technical end users to program collaborative robots, thereby laying the groundwork for future paradigms of robot programming for domain experts.

3.2 Programming the Robot

Programming a collaborative robot means specifying logic structures (e.g., conditions and loops), objects, actions, and locations so that the robot can execute the desired behavior safely and correctly. CAPIRCI implements the simple programming journey paradigm:

users first outline the task in an informal way, then refine and finalize it with more structured tools. More specifically, three alternative workflows are supported:

1. Define a new task entirely through natural language, interacting with a chat.
2. Define a new task entirely through a graphical (block-based) interface.
3. Start interacting with the chat to create a task draft, then refine and complete it using the graphical interface.

This flexibility allows users to adopt the interaction style that best matches their skills, and it enables a smooth learning curve from simple to complex tasks.

3.2.1 Natural Language Interface

The natural language interface allows users to interact with CAPIRCI as if it were an instant messaging system, making the process familiar and intuitive. Through a system-driven dialogue, the user can specify the object to pick, the location where to place it, the number of repetitions, and any intermediate actions on the object.

The chat interface supports simple, direct statements and manages dialogue scripts such as single-turn task specification, question–answer interactions to collect missing parameters, and confirmation of interpreted requests. For example, a user can write *"put 10 flasks in the box"*, and CAPIRCI extracts object, location, and repetitions from that sentence. An example of multi-turn interaction is reported in Figure 3.1. In the third step of the figure, the user is asked whether they want to visualize the defined task in the graphic interface; if confirmed, they are directly redirected to it.

The natural language interface of CAPIRCI relies on widely used NLP tools to parse and interpret user input. Specifically, Stanford CoreNLP[3] and the Python Natural Language Toolkit (NLTK)[4] were adopted to implement tokenization, part-of-speech tagging, and shallow parsing. The system recognizes a finite set of noun phrases and imperative verb patterns corresponding to the domain of robot programming (e.g., "pick", "place"). This approach allows CAPIRCI to map simple, direct sentences into robot actions without requiring users to learn a formal syntax.

In this version of the system, natural language understanding is intentionally simple: it assumes direct speech, affirmative commands, and single-level references. Pronoun resolution, anaphora, and negation are not handled, although synonym lists provided by the user can enrich the vocabulary for object recognition.

3.2.2 Graphic Interface

The graphic interface of CAPIRCI complements the natural language interface by providing a full-featured, block-based programming environment specifically designed for collaborative robots. While inspired by well-known educational environments such as Blockly[5] or Scratch[6], CAPIRCI implements a custom block language tightly integrated with robot objects, locations, actions, and events. An example of a program created with the graphic interface, corresponding to the interaction in Figure 3.1, is reported in Figure 3.2.

[3] CoreNLP - https://stanfordnlp.github.io/CoreNLP
[4] NLTK - https://www.nltk.org
[5] Blockly - https://developers.google.com/blockly
[6] Scratch - https://scratch.mit.edu

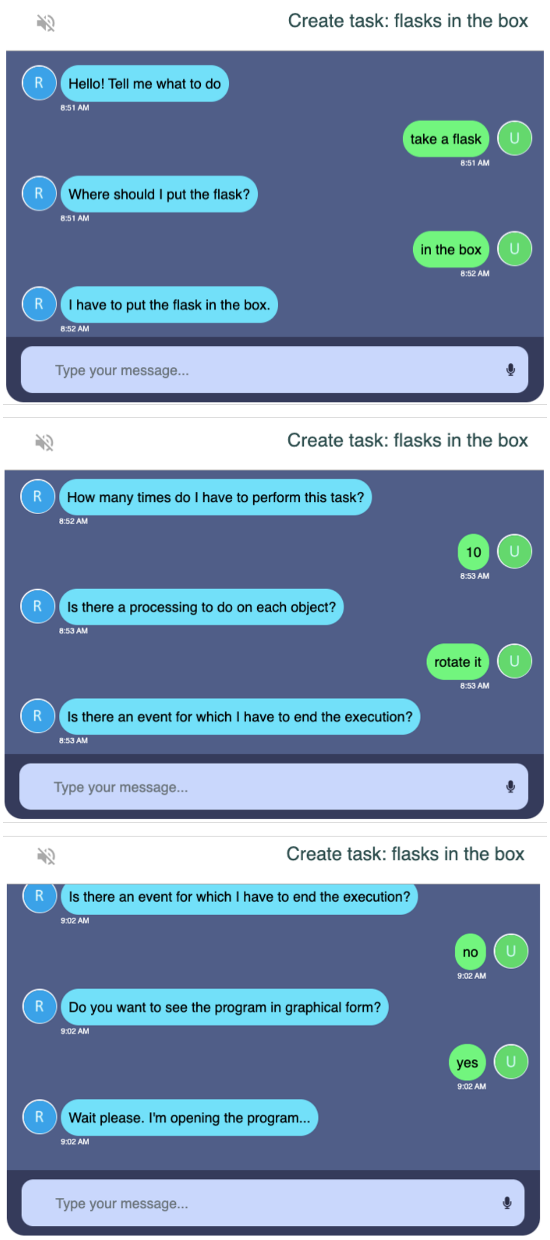


Figure 3.1: Creating robot tasks with CAPIRCI through the natural language interface

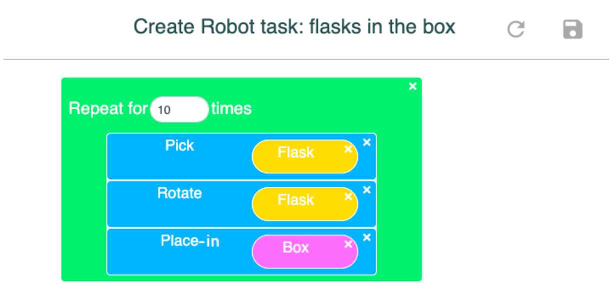


Figure 3.2: A program created through the chat of CAPIRCI visualized in the graphic interface

The whole interface (see Figure 3.3) consists of two main areas: the *Libraries* area, which contains all available blocks grouped into six categories (Tasks, Controls, Events, Actions, Objects, and Locations), and the *Project* area, where users assemble programs by dragging and dropping blocks. *Tasks* store previously created robot programs that can be reused and modified for new tasks. *Controls* provide programming constructs such as loops and conditionals, including "When," "Stop-when," and "Repeat" blocks that can be nested and require event conditions to be specified in designated drop zones. *Events* define the conditions necessary to complete control structures, enabling behaviors such as stopping an action sequence when a specific object is detected. *Actions* represent high-level robot operations including basic "pick" and "place" functions as well as user-defined actions like "rotate" or "shake", abstracting low-level implementation details. *Objects* contain domain-specific items that the robot can manipulate, with the capability for users to define new objects through image capture and parameter specification. *Locations* specify spatial positions where objects can be placed, with new locations defined by physically moving the robot arm to the desired coordinates. Each library is color-coded to help users distinguish between different elements at a glance. For example, action blocks are light blue, object blocks are yellow, and location blocks are pink.

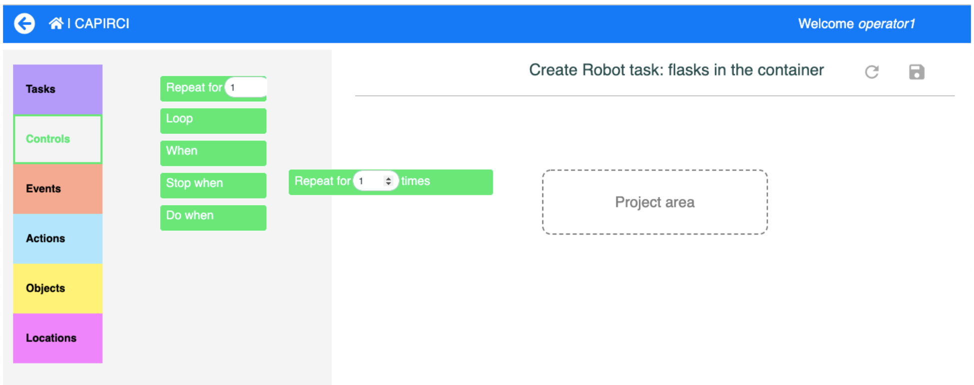


Figure 3.3: The graphic interface of CAPIRCI during program creation

Beyond simply displaying blocks, the interface enforces semantic constraints. If a user attempts to drop an object block where an action block is expected, the system prevents the operation and issues a warning. Tooltips explain the meaning of each block and provide immediate feedback. This design reduces the cognitive load for users and mirrors the incremental learning curve envisioned by the simple programming journey paradigm.

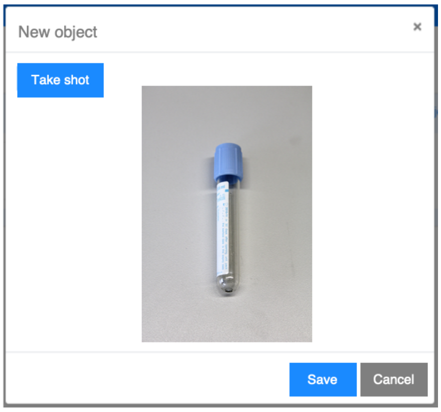
Every block corresponds to a robot instruction with specific parameters. Built-in actions like "pick" and "place" have predetermined settings, whereas custom objects, locations, and actions require users to define their own parameters such as coordinates, forces, or movement sequences. When a program is complete, the sequence of blocks is translated into an XML description that captures all the actions, controls, and conditions.

This architecture allows CAPIRCI to combine two ideas: on one hand, reusability of elements across tasks (through the *Tasks* library), and on the other, fine-grained control over robot actions while hiding low-level complexity from the user. Ultimately, the multimodal approach enables users to transition seamlessly from natural language interaction to visual task representation, thus bridging the gap between conversational programming and explicit program visualization. This synergy not only reinforces understanding of the underlying task logic but also enhances transparency, editability, and user trust in the collaborative process.

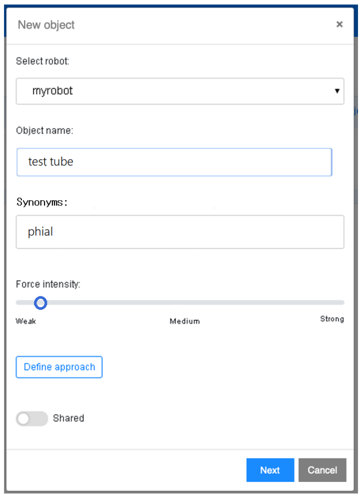
3.3 New Items

In line with the third standpoint (i.e., *robust cognitive matching*), CAPIRCI was designed to be domain-independent. Users can define new objects, locations, and actions to adapt the system to their own work environment.

When defining a new object, users provide a picture taken with the robot camera (Figure 3.4a) and supply identification and technical data such as the required gripping force and approach distance[7] (Figure 3.4b). Users can also assign alternative keywords (i.e., user-defined synonyms) that enable the system's natural language interface to correctly recognize and reference the same object even when different words or expressions are used. In addition, a dedicated *Shared* toggle allows users to make the defined object available to other users, a feature that is also provided for Locations and Actions.



(a) Taking a shot of the object



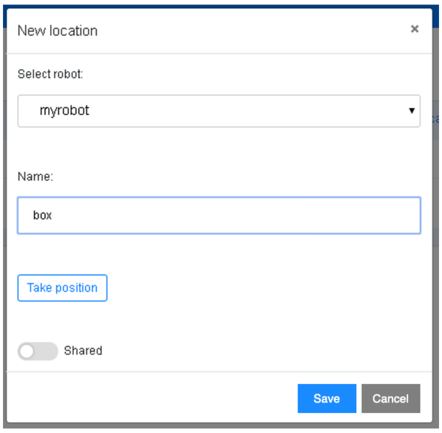
(b) Providing object data

Figure 3.4: Definition of a new object in CAPIRCI: (a) providing object data; (b) taking a shot of the object

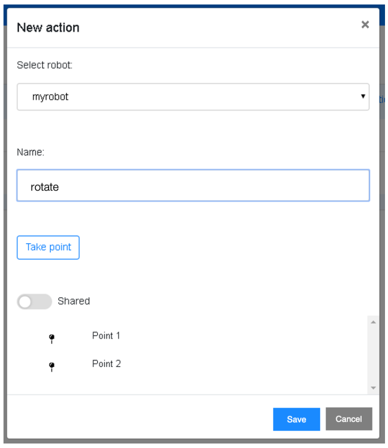
Locations are defined by moving the robot arm to the desired point, storing the position, and assigning a name to such a position. Examples of locations include trays, boxes, or table locations. The form used to define the new location "box" is shown in Figure 3.5a.

New actions represent robot trajectories or processing steps between pick and place operations. Actions are recorded by moving the robot arm along the desired path and saving key points. Figure 3.5b shows the form used to define the new action "rotate".

[7]In Robotics, "approach" is meant to be the safe or optimal distance a robot maintains from an object before initiating an action, ensuring accurate operation and safety.



(a) New location



(b) New action

Figure 3.5: Definition of a new location (a) and a new action (b) in CAPIRCI

This mechanism allows a collaborative, incremental collection of knowledge: objects, locations, and actions defined by one user can be shared with others. This approach makes CAPIRCI a flexible and continuously evolving system, where knowledge progressively accumulates through user contributions.

3.4 User Study

To validate the multimodal approach to robot programming proposed with CAPIRCI, two user studies were conducted in a realistic laboratory setting. The first study was a controlled experiment that involved 20 participants (8 women and 12 men, mean age approximately 35 years) with no prior experience in robot programming. A between-subject design was adopted: Group A used CAPIRCI in the multimodal mode (natural language chat plus graphical interface), whereas Group B used only the graphical interface. Participants were asked to perform four tasks of increasing complexity inspired by laboratory scenarios (e.g., pick-and-place of tubes and samples) under a fixed time limit. The second study was a usability study where 10 participants completed eight progressive tasks ("register the robot," "define an object," "define an action," "define a location," "create a task via chat," "modify it via graphical interface," "recreate it via graphical interface," and "execute it on the robot") following only a brief 10–15 minute introduction. During both studies, observations and spontaneous comments were collected, and at the end, participants completed the System Usability Scale (SUS) questionnaire [27] and a short semi-structured interview.

Results showed that all participants were able to complete the assigned tasks, including creating complex programs (with loops and user-defined objects) and executing them on the robot. Quantitative analysis indicated shorter execution times and higher success rates in the multimodal condition compared with the graphical-only condition. SUS scores were very high (all over 80), exceeding the scores typically reported in the literature for comparable tools (see [72, 152]). Participants described the experience as pleasant and satisfying and expressed increased willingness to use collaborative robots.

3.5 Limitations

Based on the user studies reported in [47], CAPIRCI demonstrated the feasibility of a multimodal approach to collaborative robot programming. However, it also revealed several limitations.

One of the most evident constraints concerned the natural language interface. The parser could handle only straightforward imperative sentences and a limited set of noun phrases. Pronouns, indirect speech, or more complex syntactic structures were beyond the scope of the system, and this occasionally produced misinterpretations. Nevertheless, these limitations provided useful insights into the kind of linguistic phenomena that should be addressed in future versions.

A second area of limitation was related to the block-based environment. While the custom block language enabled loops, conditionals, and events, some constructs still required users to understand how to manipulate them correctly. For example, tasks involving nested "Repeat" blocks or event-triggered behaviors occasionally generated confusion during the user studies. This highlighted the need for clearer visual cues, improved onboarding, and possibly adaptive guidance mechanisms.

The interaction between the virtual and physical components also emerged as a source of friction. Users had to move the robot arm manually to define approach distances or capture object snapshots at the right time, and sometimes misunderstood the order of these operations. This interplay between virtual programming and physical calibration is intrinsic in robot programming, but CAPIRCI's early prototype showed where usability improvements were most urgent.

Overall, these limitations were not only expected but also instrumental: they guided the design of subsequent iterations, informed the research questions of this thesis, and clarified where multimodal interaction can be most beneficial.

Chapter 4

A Domain-Specific EUD Environment: PRAISE

This chapter presents PRAISE (Pharmaceutical Robotic and AI System for End Users)[1], a domain-specific EUD environment designed to support pharmacists in programming collaborative robots for the preparation of personalized medicines. Whereas CAPIRCI demonstrated the feasibility of combining natural language and block-based programming for generic pick-and-place tasks, PRAISE extends these concepts to a highly regulated and safety-critical domain where precision, traceability, and user control are essential, while explicitly aiming to overcome the limitations identified in CAPIRCI.

Enabling pharmacists, domain experts without programming expertise, to define and manage robotic tasks raises a broader question about the role of EUD in specialized domains. Addressing this challenge motivates the formulation of RQ1: *To what extent can domain experts define domain-specific tasks and items through dedicated user interfaces designed for non-programmers?*

4.1 Case Study

According to Eurostat [41], European people over 65 years old will reach 129,8 million by 2050, up from 90,5 million in 2019, with a median age that is projected to reach 48.2 years, thus increasing by 4.5 years in about thirty years.

Within this context, as a preliminary step, I conducted a systematic review of pill and medication dispensers presented in the literature to analyze existing technologies and workflows from a human-centered perspective [53] (reported in *"A systematic review on pill and medication dispensers from a human-centered perspective"*, published in the *Journal of Healthcare Informatics Research*). The review mapped current solutions, identifying gaps such as limited adaptability, insufficient support for caregivers in configuring devices, and a lack of mechanisms for personalization by end users. These findings provided important guidance for designing EUD systems that are intuitive, flexible, and responsive to users' needs, serving as a foundation for the work presented in this chapter.

A further aspect related to aging is that, not only is the pharmaceutical sector constantly growing [75], producing an increasing number of types of medicines, but there is also a growing need for medicines tailored to patients' characteristics (e.g., gender, weight, allergies, etc.). Such personalized medicines include capsules, tablets, creams, and oint-

[1]PRAISE - https://github.com/luigigargioni/galenic

ments produced at pharmacies, starting from the so-called *galenic formulations* that are prepared by pharmacists by mixing and blending different ingredients. These medicines can be produced at pharmacies in small batches, taking into account the exact quantity required for the specific patient, thus contributing to limiting material waste. In addition, the industrial production of limited quantities of medicines is often not profitable enough for the industry, and thus, some medicines may be very expensive or not produced at all [147]. This phenomenon is evident in medicines devoted to rare diseases. In this way, the production of galenic medicines contributes to responsible health.

The preparation of galenic formulations is traditionally carried out manually by measuring, weighing, and mixing various raw materials according to specific recipes. The process often occurs in a cleanroom environment, in order to minimize the risk of contamination and ensure product quality and safety [40]. However, some steps of such a manual process are highly repetitive and require a lot of time and physical effort; therefore, they can be perceived as tedious by human workers or can cause them strain and injury, especially if they must be performed for prolonged periods. Other steps, instead, require a high level of precision and attention to detail [153], thus becoming prone to human errors, such as incorrect measurements and cross-contamination.

Current information technologies can be adopted to address the above problems. In this project, in line with recent advances (e.g., [102]), the use of collaborative robots is proposed to assist pharmacists in performing the most repetitive, tiresome, and error-prone steps of the galenic formulation preparation. Indeed, cobots can perform tasks that require a high level of precision and consistency, such as mixing and blending, to obtain more homogeneous and uniform products [20]. Also, the placing of prepared capsules into given containers can be assigned to cobots to increase speed and efficiency [102].

In order to address these challenges and enable pharmacists to customize and control collaborative robots according to the specific requirements of galenic formulation preparation, it is essential to provide an EUD environment that supports the creation of robot tasks. Such an environment would empower pharmacists, who are domain experts but not necessarily programming experts, to program and adapt cobots to their workflows safely and efficiently. The following section outlines the research methodology adopted to design such an EUD environment.

4.1.1 Research Methodology

The methodology adopted to design an environment that allows the programming of collaborative robots that support pharmacists in galenic formulation preparation follows a Human-Centered Artificial Intelligence (HCAI) approach [135]. The approach is *human-centered* since system design started from pharmacists' practice, needs, and abilities; AI techniques have been exploited to facilitate user-system interaction, by means of natural language interfaces and image processing algorithms. The approach is articulated in three stages that were iteratively executed until a complete and functioning system was obtained:

1. *User Research*: this stage includes the analysis of the domain and of pharmacists' current practice; the definition of users' profile is also part of this stage; different methods are used to carry out this stage, such as direct observation, structured and unstructured interviews, focus groups, definition of personas and scenarios.

2. *Interaction Design and System Development*: in this stage, the software system

supporting the EUD activities of pharmacists aimed to define the domain concepts and tasks of collaborative robots is developed; this stage started from low-fidelity and high-fidelity prototypes, and developed until system implementation.

3. *Evaluation*: each prototype or interactive system created along the development cycle was evaluated in terms of usability and user experience.

To limit the scope of the analysis, it was decided to focus the study on the preparation of galenic solutions in granular form, i.e., galenic formulations involving the dissolution of some ingredient during the process and the final production of capsules containing the medicine.

4.2 Interviews with Domain Experts

As reported in *"Exploring the Adoption of Collaborative Robots for the Preparation of Galenic Formulations"* published in the *Information* journal [55], three domain experts were interviewed in the user research stage to gather information about the work practice, the terminology used, the environmental setting, and possible issues affecting the galenic preparation process. The profiles of the three domain experts are reported in Table 4.1, where technical proficiency is a self-assessment on a 1-5 scale of proficiency in computer use, with 1 indicating low proficiency and 5 indicating very high proficiency.

Table 4.1: Domain experts' profiles

ID	**Age**	**Gender**	**Profession**	**Work experience *(years)***	**Technical proficiency *(1-5)***
E1	32	Male	Pharmacist	6	3
E2	33	Female	Nurse assistant in pharmacy	10	4
E3	29	Female	PhD student in pharmacology	2	3

I conducted two to three semi-structured interviews with each expert, carefully taking notes and later organizing them into a single, well-structured document. Table 4.2 lists the open-ended questions used during the interviews.

Table 4.2: Structure of the semi-structured interviews

Part 1: Questions about the current work process.	
1	Please describe the procedure for preparing galenic formulations.
2	Is this process currently automated in any way, even partially?
3	What difficulties may arise during this process?
4	Can this process be partially or fully automated?
5	Which steps do you believe can be automated?
Part 2: Questions about the role of a collaborative robot in the preparation of galenic formulations (before asking these questions, a brief description of a collaborative robot in terms of a robot arm that can safely work close to the human was provided).	
1	Can a collaborative robot aid in automating this process?
2	How would you envision collaborating with a robot?
3	Are there regulatory restrictions on the use of robots for galenic preparations?
4	Is the use of robots suitable for working with galenic preparations?

Pharmacists explained the most critical steps of their work process, underlining in which steps automatic support could be useful and where human judgment must be kept in the process. They also provided me with technical documentation and videos that explained some important details of the galenic preparation process, such as the mandatory steps to ensure accuracy and efficiency [36] and the tools used during the process.

A thematic analysis [25] was then performed by one of my supervisors and me on the resulting document, following a deductive and semantic approach. Codes were extracted by each researcher and then reconciled in a meeting session, which also served to identify the main themes.

The themes that emerged from the interviews, which will be described in Section 4.2.1, are the following:

- Current procedure for the preparation of galenic formulations;
- Problems and challenges affecting the current work practice;
- Opportunities and threats related to the introduction of collaborative robots.

The knowledge acquired through the interviews, along with additional analysis of documents describing the galenic preparation process and research on users' profiles, allowed i) identifying where, in the work process, collaborative robots could be of help (see Section 4.2.2); and ii) outlining the user requirements for the EUD environment (see Section 4.2.3).

4.2.1 Themes Emerged from the Interviews

Table 4.3 summarizes the themes that emerged from the thematic analysis with their corresponding codes.

Table 4.3: Themes and codes emerged from the thematic analysis of the interviews

Theme	Codes
Current procedure for the preparation of galenic formulations	• Galenic formulation procedure • Automatic mixer • Operculator • Manual tools
Problems and challenges in current work practice	• Mainly manual process • Time-consuming steps • Low-value tasks • Compound homogeneity • Consistency of capsule filling • Precision required • Toxic or hazardous substances
Opportunities and threats in introducing collaborative robots	• Only certain steps can be performed by a robot • Support the worker, not replace them • Suitability of robots to different users • Regulatory considerations

Current procedure for the preparation of galenic formulations

Preparing a galenic formulation must ensure accuracy and efficiency [36] and should be conducted following the Good Manufacturing Practices (GMP) [140] in compliance with regulatory guidelines to ensure the safety and quality of the final product. For this purpose, the pharmacists must carry out a set of activities in a sequence. The interviews with experts allowed us to acquire knowledge about the sequence of steps that compose the galenic preparation process and the tools used in some of these steps. In this case, the sequence of steps is the following:

1. *Formulation Design*: The first step is to determine the composition of the galenic solution, including the active pharmaceutical ingredients (shortly, ingredients in the following) and excipients.

2. *Solubility Assessment*: This step helps in selecting appropriate solvents and co-solvents to achieve the desired solubility and stability of the solution.

3. *Ingredient Weighting*: Accurate weighting of ingredients and excipients is crucial to ensure the formulation meets the desired specifications.

4. *Mixing and Dissolution*: The weighed ingredients are mixed and dissolved in the chosen solvent(s). In technologically advanced pharmacies, an automatic mixer is used in this step, but most pharmacists still use manual tools.

5. *pH Adjustment*: Depending on the formulation requirements, the pH of the solution may need to be adjusted.

6. *Filtration and Clarification*: To remove any particulate matter or undissolved solids, the solution may undergo filtration using filters of appropriate pore sizes.

7. *Quality Control and Analysis*: The prepared galenic solution is subjected to rigorous quality control measures to ensure its safety, efficacy, and compliance with regulatory standards.

8. *Packaging*: This phase involves transferring the solution into the capsules, ensuring uniformity of dosage, and avoiding cross-contamination.

9. *Storage*: The storage step consists of transferring the capsules into suitable containers, such as glass or plastic bottles.

Problems and challenges in current work practice

From the interviews, it emerged that the current preparation of galenic formulations is mainly manual, thus yielding problems and challenges that human workers must address.

Time-consuming activities must be performed, especially when a large number of capsules are produced. This problem was underlined for Step 4, which has the goal of obtaining a uniform mixing and efficient dissolution of ingredients to ensure consistent drug potency and therapeutic efficacy, and for Step 9, where transferring each capsule at a time into a container is considered a low-value and repetitive task for the pharmacist.

In addition, several activities require high precision. For instance, in Step 4, the mixing activity must lead to compound homogeneity. This requires precise mixing ratios, consistent particle size reduction, and optimal dissolution rates; any error may lead to batch-to-batch variability and quality issues. Another issue related to the *Mixing and Dissolution* step (Step 4) is that ingredients and excipients may be toxic or hazardous substances, thus their manipulation poses challenges to human workers. Precision is also fundamental in Step 8, which consists of transferring the galenic preparation compound into capsules made available in a special machine. In this step, consistent quantities of preparation must be transferred into the capsules to guarantee uniformity and reliability. To ensure consistency, the manual process usually leads to compound waste. Furthermore, during this operation, it is essential to prevent contamination, degradation, or other undesirable changes to ensure that the medicines remain effective and safe throughout their life.

Opportunities and threats in introducing collaborative robots

A third theme emerging from the interviews concerned the potential for automation and the role a collaborative robot could play in the galenic preparation process, as well as related threats envisaged by domain experts.

As described in Table 4.2, questions bringing about this theme were posed after a brief description of the characteristics of collaborative robots. The interviewees were not aware of the existence and potential of this kind of robot. Based on the description of their work practice and our knowledge of the literature about collaborative robots in the pharmaceutical sector, we identified the adoption of these robots as an innovation opportunity to be discussed with the experts. This opportunity could not have emerged without the knowledge sharing between domain experts and computer scientists enabled by the human-centered methodology adopted. The domain experts observed that some steps of the galenic preparation process cannot be delegated to a machine yet, since they require decision-making and human control; labor-intensive, low-added value, and repetitive tasks have been considered by the domain experts to be the most suitable for automation.

Domain experts also highlighted that robots must collaborate with pharmacists rather than replace them, thus leaving decision-making in the hands of healthcare professionals. They also underlined that pharmacists possess different expertise in technology, and thus, some users, especially the older ones, might encounter difficulties in dealing with robots. Considering that pharmacists have no computer programming experience, a main challenge that emerged was instructing the robot in doing the desired activities.

Finally, it emerged that regulations could be easily satisfied thanks to sanitized robotic assistants, ensuring the utmost cleanliness.

4.2.2 Human-Robot Collaboration in Galenic Formulation Preparation

The themes that emerged from the interviews, combined with our competencies in collaborative robots, allowed us to identify which steps could be delegated to a robot and which ones can be performed at the moment only by a human. Figure 4.1 illustrates the sequence of steps and highlights which ones must be carried out by human workers before the use of a collaborative robot (steps 1, 2, and 3), and which ones can become the core activities of the human-robot collaboration, distinguishing the activities delegated to the robot (steps 4, 8 and 9) from those that should remain in the hands of the humans, possibly with the help of further machines (steps 5, 6 and 7).

This hypothesis was discussed with one of the three domain experts (E1), who considered it a valid solution.

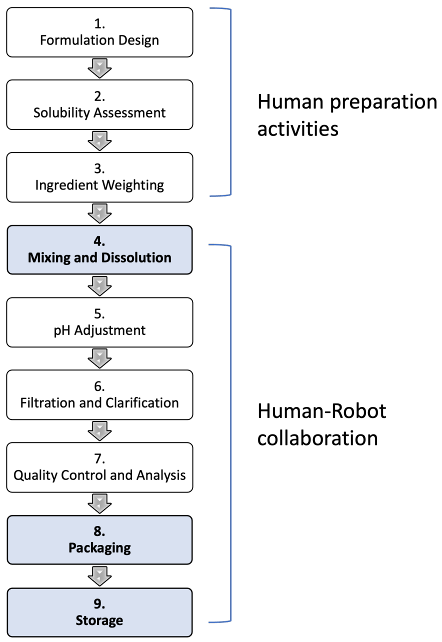


Figure 4.1: The steps of galenic solution preparation: from step 4 to step 9, a human-robot collaboration can be put in place

The goal of the *Mixing and Dissolution* step (Step 4) is to achieve a uniform mixing and efficient dissolution of the preparation to ensure consistent drug potency and therapeutic efficacy. Different mixing techniques, such as stirring, shaking, or vortexing, are employed to ensure uniform distribution and dissolution of the ingredients. If necessary, heating or sonification may be used to aid the dissolution process. This activity is a manual process that may become very labor-intensive and time-consuming, especially when dealing with the production of a large number of capsules or complex formulations. Human operators must pay attention to precise mixing ratios, consistent particle size reduction, and optimal dissolution rates. Thus, human errors may occur, leading to batch-to-batch variability and quality issues. In this step, robots can work continuously without fatigue, performing mixing and dissolution processes on a large scale, thus reducing processing time and increasing productivity. Furthermore, robots can perform tasks with high precision and accuracy, since they can precisely control the speed, duration, and force applied during the mixing activity. This yields more reliable and reproducible results, ensuring batch-to-batch consistency and reducing variability in the resulting product. Finally, when dealing with dangerous substances, robots can minimize health and safety issues, protecting workers from potential harm.

The goal of the *Packaging* step (Step 8) is to transfer the obtained galenic preparation into capsules. This requires the proper handling of the galenic preparation to maintain stability, potency, and integrity over time. To carry out this activity, pharmacists use a special machine, called *operculator* or *operculating machine* or *sealing machine*. This is used to place half capsules in the cavities of a grid and then fill the bottom half of each capsule with the galenic preparation. The operculator is equipped with a capsule loading mechanism to place the bottom halves in the correct orientation for filling and sealing. Different types of grids may be used according to the shape, size, and quantity of the capsules to be produced. When the bottom halves of the capsules have been placed in the chosen grid, the human operator will pour over the bowl containing the preparation to fill the capsules. Then, the operculator is used to align the top halves with the bottom halves of the capsules, and eventually seal the capsules securely with controlled compression. The sealed capsules can then be collected from the operculator for further processing, inspection, or storage. In this step, a robot can accurately measure and dispense the appropriate quantities of galenic preparation into capsules by performing precise and consistent movements, minimizing the risk of spillage, maintaining proper hygiene and sterility, and avoiding human errors during the process. In addition, a robot can be programmed to comply with GMP and to adhere to safety protocols and regulatory requirements.

The *Storage* step (Step 9) consists of transferring the sealed capsules from the operculator into suitable containers (e.g., plastic or glass bottles) for delivery to the customer. This is a trivial and repetitive task for the pharmacist, which requires a huge amount of time with very low added value. It can easily be performed by a cobot through the execution of a pick-and-place activity. This would improve the overall efficiency of the process and enhance the ergonomics of the work.

4.2.3 User Requirements

After the identification and comprehension of the steps that can be carried out collaboratively between a pharmacist and a robot, I created two user personas (see Figure 4.2 and Figure 4.3) to empathize with the target user population of the system we intended to develop to support pharmacists in the programming of robot tasks.

Personas also allowed us to delineate the main characteristics of the users: they are experts in the pharmaceutical domain, thus they have at least a bachelor degree and adopt a specific domain language; they are careful and precise workers, eager to help

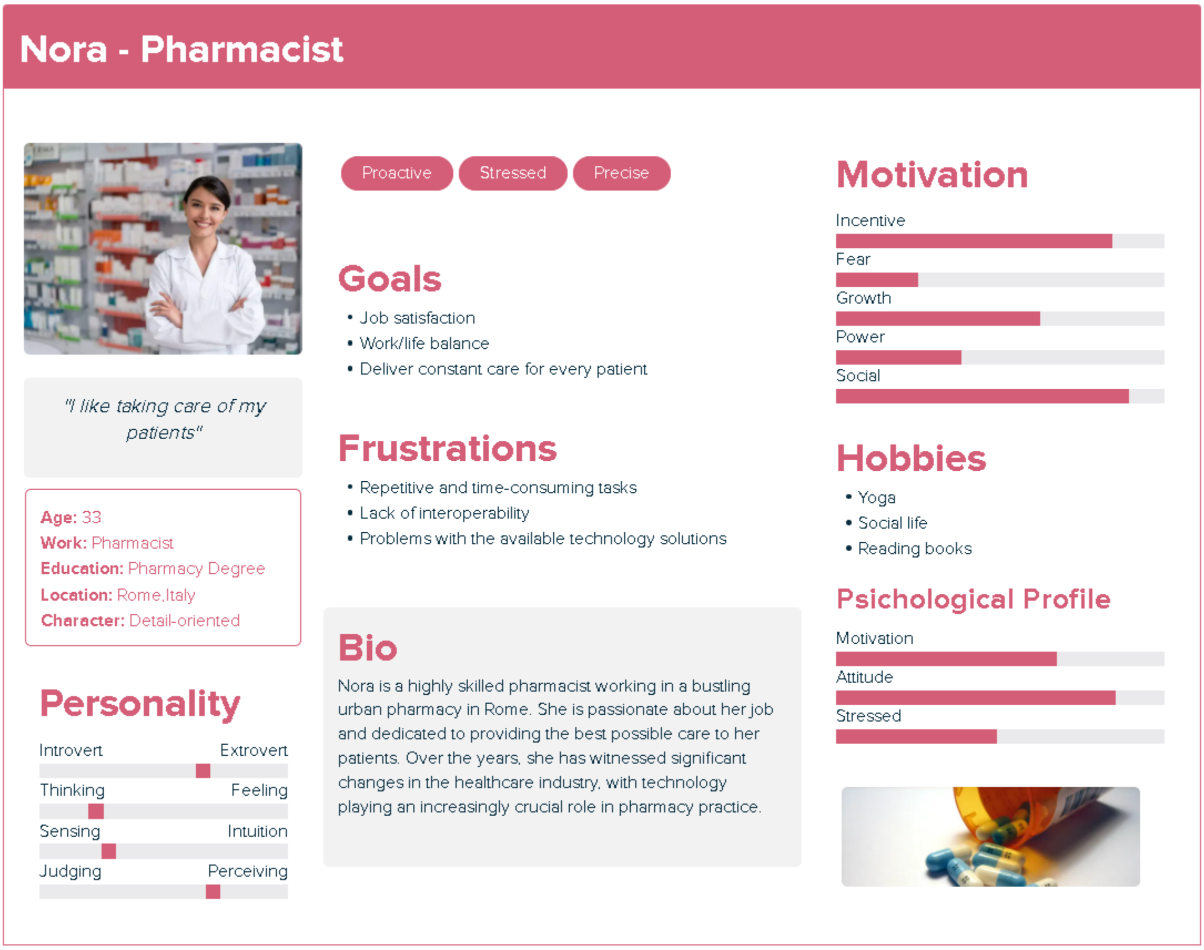


Figure 4.2: A persona of an early-career pharmacist representing a target user of PRAISE

their patients, but often overwhelmed by their work, due to the different tasks to be performed in a pharmacy; they may have a short or long experience in the preparation of galenic formulations, and, according to their age, they may be more or less willing to accept the intervention of automation in a complex activity that has been carried out manually for centuries; some may find instructing a robot and collaborating with it exciting, but others may be afraid to make mistakes and create dangerous situations.

On the basis of these considerations, we decided to design an EUD environment that could guide the users step-by-step during the definition of robot tasks and always provide a suitable and comprehensible explanation of the activities carried out with the system.

4.3 Interaction Design

With CAPIRCI (see Chapter 3), we obtained very positive results proposing a multimodal approach to robot programming, which adopts a combination of EUD techniques to accommodate different users' backgrounds and programming skills. In that case, only pick-and-place tasks could be defined by the end user interacting with two integrated features: i) a chat-based interface where a simple guided natural language dialogue led to defining robot tasks, and ii) a graphic interface where robot tasks can be expressed through the composition of specific types of blocks. A comparative experiment demonstrated that users preferred multimodal interaction to graphic interaction alone [47].

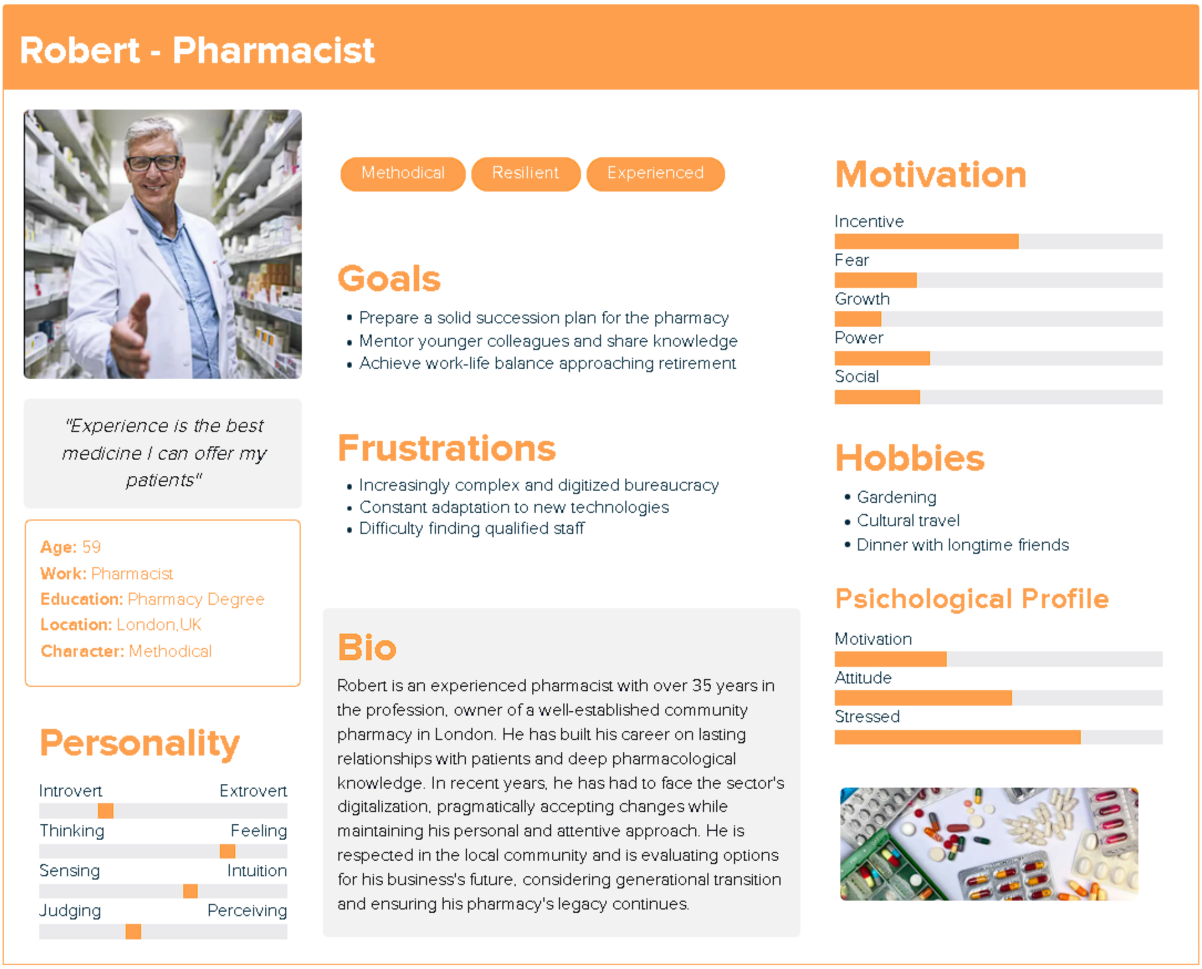


Figure 4.3: A persona of an experienced pharmacist representing a target user of PRAISE

In the case of galenic formulation preparations, the robot program to be generated must be structured along a sequence of steps that encompass different types of robot actions (e.g., mixing ingredients, filling capsules with the galenic preparation, picking capsules, and placing them into containers), interleaved with human actions. Thus, robot programming must be more flexible than in our previous EUD environment, and, at the same time, it must be suitable for a well-identified user profile, i.e., a pharmacist.

We decided to maintain the multimodal approach based on a chat interface and a graphic interface, but we completely redesigned both of them to cope with the problem at hand, trying to enhance the naturalness, flexibility, and verifiability of the programming activity. To this end, we decided to integrate the chat-based interface with a generative AI approach, exploiting LLMs. However, we had to make such an integration as deterministic as possible, given that LLMs are based on probabilistic algorithms that often provide different answers to the same questions, and, above all, may generate imprecise or wrong answers. Programming a collaborative robot must ensure safe program execution in the workplace, and the galenic medicines resulting from robot program execution must be compliant with specified prescriptions and quality standards. Following an HCAI methodology, we considered all these issues during the design and development of PRAISE, in order to provide the user with complete control of the generated program and prevent wrong task execution. This led to design and develop the EUD features, starting from digital low-fidelity mock-ups to be discussed with one of our domain experts (E3), until creating a high-fidelity interactive prototype.

4.3.1 Interaction Requirements

The interaction between pharmacists and the proposed system may occur at different times and with different requirements:

1. *At design time*: here PRAISE supports end users in defining domain items, namely the objects (e.g., operculator grids, containers, etc.) that the robot must recognize and be able to manipulate, and the mixing actions that the robot must be able to carry out during the mixing and dissolution phase. In this stage, the EUD environment exploits image recognition algorithms and graphic interfaces to allow the pharmacists to "teach" the robot the domain concepts. A basic setting of the system can be prepared by software developers, but pharmacists must be able to extend the domain when needed, e.g., new grid types must be recognized and new mixing actions must be performed by the robot.

2. *At programming time*: the EUD environment supports the definition of tasks for the preparation of galenic formulations using a multimodal approach based on a chat-based interface integrating an LLM and a graphic interface; the latter permits the verification of the programmed task and is crucial to obtain adequate system explainability and trustworthiness. In fact, the natural language dialogue with the chat-based interface leads to the definition of a robot program, whose correctness must be verified by the user before its deployment. Since the user is not an expert in robot programming, representing the program as a sequence of blocks in the graphic interface allows the user to evaluate program correctness and directly modify the blocks and/or their parameters if needed.

3. *At execution time*: the execution of the programmed tasks requires collaboration between the pharmacist and the robot. For example, chemicals can be put by the pharmacist in a specific bowl, and the robot can subsequently perform a blending activity; to ensure effective collaboration, proper human-robot coordination must be managed through the system. The collaboration described in this execution phase has not yet been implemented and represents a potential direction for future development. This is mainly due to technical limitations related to the available materials and robotic equipment, which would have prevented the real execution of collaborative tasks on the robot. Therefore, the work focused on the more exploratory and cognitively engaging phases of user interaction, namely points 1 and 2 above.

4.4 System Mockups

The following section illustrates preliminary solutions, developed with the aid of low-fidelity prototypes created using Balsamiq[2], designed to support specific interaction goals for a pharmacist. The outcomes of this design process are reported in *"Designing Human–Robot Collaboration for the Preparation of Personalized Medicines"*, published in

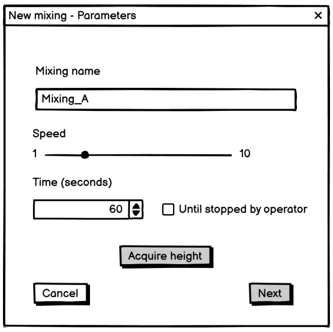
[2]Balsamiq - https://balsamiq.com

the proceedings of the *1st International Conference on Information Technology for Social Good (GoodIT) 2023 (Lisbon, Portugal)* [51].

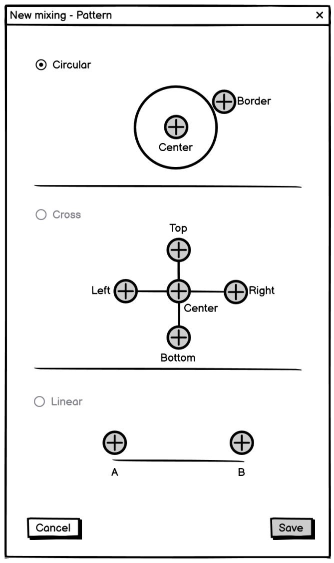
The supported interaction goals are the following:

1. *Domain definition*: the system should support the definition of mixing actions that the robot must be able to carry out in the *Mixing and Dissolution* phase, the definition of operculator grids that the robot must recognize during the *Packaging and Storage* phase, and the definition of containers that the robot must fill to complete the *Packaging and Storage* phase. Mixing actions are defined by specifying basic parameters such as name, execution speed, optional duration, and height (Figure 4.4a), and by setting the motion pattern through teach points[3] (Figure 4.4b), where the robot's positions are manually acquired. Operculator grids, used in packaging and storage, can be defined either manually, by setting the corner cells and the number of rows and columns (Figure 4.5a), or automatically through image recognition, with user confirmation (Figure 4.5b). Containers, representing the final destination of capsules, can be defined manually via teach for static or complex locations (Figure 4.6a), or automatically by capturing their shape and setting the unload height through image recognition (Figure 4.6b).

[3]The points can be acquired through the *teach* action, a common operation when using a collaborative robot. It consists of manually moving the robot arm to the correct position and acquiring the corresponding coordinates by selecting the point in the graphic interface.

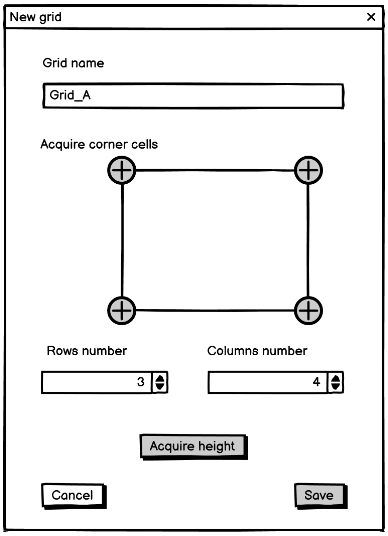


(a) Step 1: Definition of the parameters for identification and execution of a mixing action

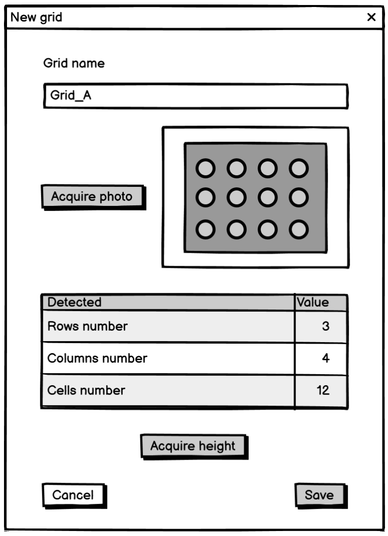


(b) Step 2: Definition of pivotal points to define the pattern of a mixing action

Figure 4.4: Mixing action definition mockup

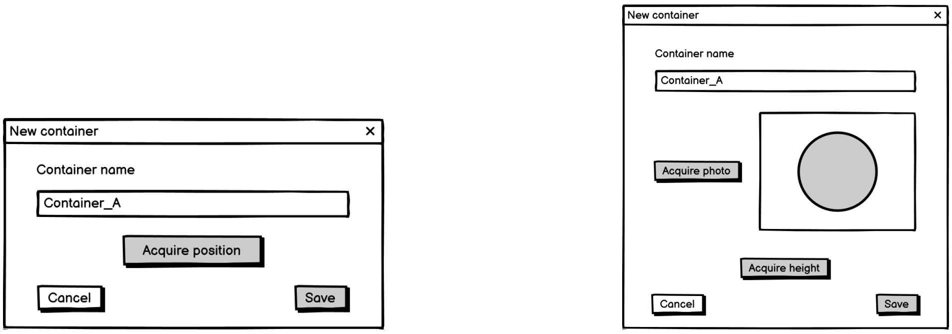


(a) Manual



(b) Automatic

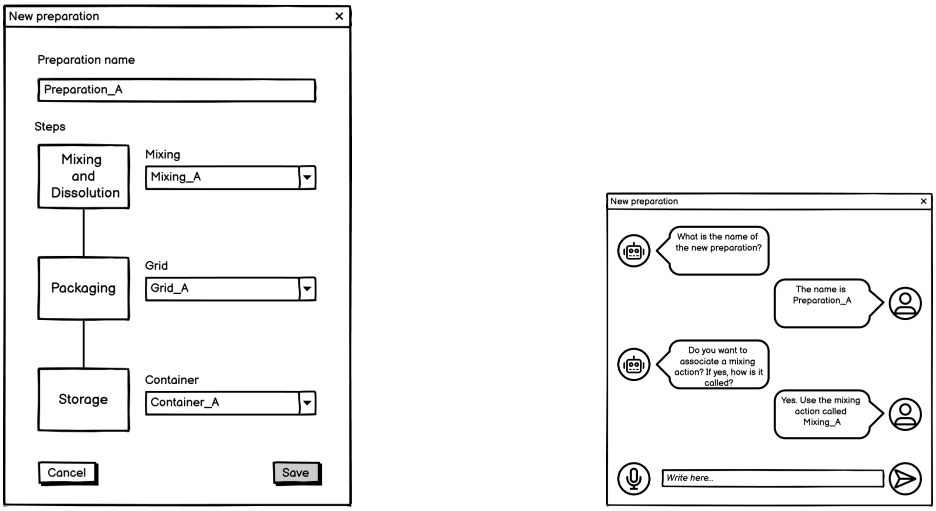
Figure 4.5: Grid definition mockup



(a) Manual (b) Automatic

Figure 4.6: Container definition mockup

2. *Preparation definition*: the system should support the pharmacist in the definition of preparations of galenic formulations using the library components and interacting with a graphic or a chat interface according to their preferences. Figure 4.7a shows the graphic interface, while Figure 4.7b shows the chat-based interface.



(a) Graphic interface (b) Chat interface

Figure 4.7: Preparation definition mockup

3. *Preparation execution*: the execution of the programmed preparations requires the collaboration between the pharmacists and the robot. The former will do some of the steps identified in the process of galenic solution preparation (see Figure 4.1), for example, ingredient weighting, and the latter will execute the steps related to mixing, packaging, and storage, always under the control of the human operator. The preparation execution will become feasible via an appropriate visual interface, facilitating the comprehensive monitoring of each step (Figure 4.8). A mockup of this interface was created; however, as mentioned above, it was not implemented due to the technical limitations related to materials and robotic equipment.

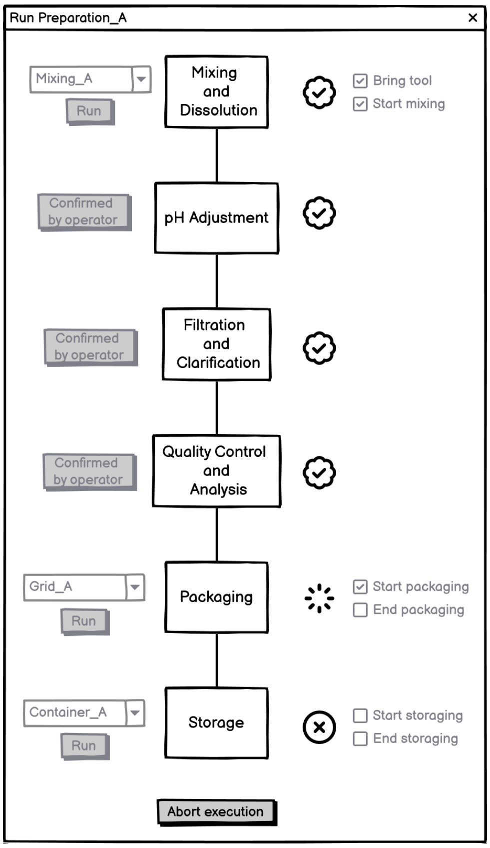


Figure 4.8: Run preparation mockup

4.5 The Developed Prototype

The mock-ups described in Section 4.4 provided the blueprint for the first working prototype of PRAISE. Developed iteratively with continuous input from one of the domain experts (E3), the prototype translated the low-fidelity wireframes into a functional web application that pharmacists could use to define collaborative robot tasks for galenic formulation preparation. The design of the prototype and the subsequent user study are reported in *"Preparation of Personalized Medicines through Collaborative Robots: A Hybrid Approach to the End-User Development of Robot Programs"* published in the *Journal on Responsible Computing* [57].

4.5.1 Architecture

The developed prototype is a web-based application leveraging the client-server paradigm. It comprises a front-end in JavaScript/Typescript React, a back-end in Python Django, and a SQLite database. The architecture and the system workflow are illustrated in Figure 4.9.

The creation of a new robot program, specifically a galenic formulation preparation, stems from the initial interaction with the Chat *(Step 1)*. The user's requests made through the chat are directed to an Adapter that utilizes an LLM API to interpret the user's requests *(Step 2)*. This module provides the LLM with the user's request alongside instructions and constraints to ensure correct interpretation of the task at hand *(Step 3)*. Upon defining the program and concluding the interaction with the Chat, a JSON

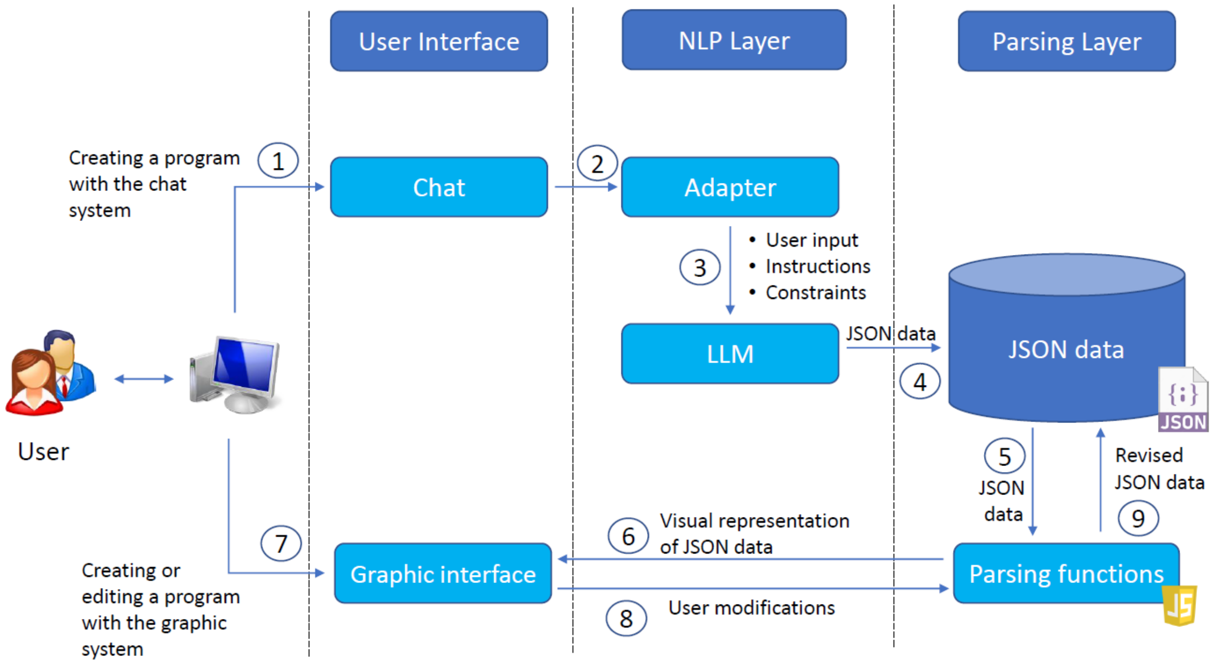


Figure 4.9: The prototype architecture

representation of the program is generated *(Step 4)*. Then, using parsing functions *(Step 5)*, the program is visualized in a graphic format *(Step 6)*. A review of the program can then be carried out *(Step 7)*, interacting with the Graphic interface, followed by confirmation or modification of the proposed solution *(Step 8)*. After completing this step, the program will be saved again as a JSON file (*Step 9*). Advanced users of the application can start defining a new preparation by using the graphic interface (*Step 7*). This guarantees a faster execution, yet it requires more cognitive effort than using the Chat.

4.5.2 LLM Selection and Setting

After the analysis of different LLM-based tools, like LLM Llama2 [105] [141] by Meta, Bard by Google [63], and OpenAI ChatGPT (Chat Generative Pre-trained Transformer), we decided to exploit the third one in PRAISE, since at the moment, it was considered the most powerful tool for text comprehension [95]. In addition, the APIs offered by ChatGPT facilitate easy and tailored integration in custom applications. We used the *gpt-3.5-turbo* model, which was available free of charge at the time of PRAISE'd development. This might imply certain limitations in terms of latency speeds and call frequency. Nonetheless, these constraints did not impede our testing as the free functionalities were adequate for our project's requirements.

The message exchange with ChatGPT involves three roles: the *User*, the *Assistant*, and the *System* [112]. The *User*, representing the individual or entity interacting with the model, carries on the conversation by providing input prompts. These prompts can encompass queries, requests, or any form of textual input. The *Assistant*, embodying the ChatGPT model, interprets the *User*'s input, comprehends the conversation context, and generates coherent and contextually relevant responses. The *System* includes the infrastructure, API endpoints, and additional components provided by the OpenAI service. Developers are responsible for managing the *System* role, overseeing tasks such as

API integration, error handling, and conversation context management through *prompt engineering*.

Prompt engineering is currently regarded as a relevant topic, but some aspects are still not well understood. Efforts have been made to develop specific patterns that enhance the models' comprehension of prompts [133]. However, the final result still heavily relies on empirical experience. Some of these patterns include starting with an explanation of the context, to clarify the application domain the LLM should belong to. Then, instructions for goal achievement must be described using short and specific sentences to facilitate model understanding.

In the frame of the project, the chat-based interface integrating ChatGPT serves the purpose of guiding the user towards programming the robot tasks for galenic formulation preparation. To tailor the behavior of ChatGPT to our specific requirements, instructions are provided before the actual beginning of the conversation with the user. In particular, the Adapter we developed is responsible for handling the calls to ChatGPT's APIs and instructing, via the *System* role, the model on the context and goal. Information about the context is given to the model through the sentence in natural language: *"The user is a pharmacist and he/she needs to create a task for a cobot to help him/her prepare galenic formulations"*. As to the goal, the Adapter provides instructions to specify that the cobot task must be composed of three steps: mixing, packaging, and storage (e.g., *"To define a task, the user has to specify three steps: mixing, packaging, and storage"*). Then, for each step, a list of required parameters and their possible values is described using a sequence of brief sentences to enable the model to recognize and interpret the details of each step. We found that clear punctuation was also crucial in achieving the desired results, as well as the order and position of sentences within the entire prompt. Several attempts have been made to determine the right syntax and level of detail required to reach a correct understanding of our instructions.

In addition to the instructions concerning the context and goal of the model, the JSON data format expected as the output of the model is included in the request. This specification is described both in natural language and in a prescribed format to obtain a correct and complete JSON document. The desired output is formally defined by passing to the *Chat completions API* of ChatGPT a function that expects as a parameter a representation of our JSON format. In this way, the API will reply with a JSON object compliant with the expected format.

Finally, the *temperature* parameter of the model must be set. This parameter ranges from 0 to 1, and is crucial to control the level of randomness of the generated text. Indeed, temperature has an impact on the probability of the potential tokens at each stage of the generation process. Increasing the temperature generates an output that is more diverse and creative, while lowering it produces a more predictable and focused output. Setting the temperature to 0 would mean that the model would always select the most probable token, resulting in its complete determinism. A temperature of 0.2 was selected in our case to obtain a suitable balance between a more deterministic reasoning and human-like behavior.

The ChatGPT's API receives the parameters from the Adapter module at every user interaction, including model, temperature, output function, and chat log. The chat log is, in turn, composed of the user message and all the instructions previously described; it is important for retaining memory of all the information already exchanged during the conversation.

4.5.3 User Interface

The Homepage of PRAISE presents itself as composed of two main areas (see Figure 4.10): on the left side, there is a collapsible menu with links to all the pages of the application that allow the user to manage existing preparations, and defining the domain concepts; whilst, in the central stage, the textual content invites the user to start defining a galenic preparation and two boxes describe the two possible interaction modalities to perform such a task, namely Chat or Graphic.

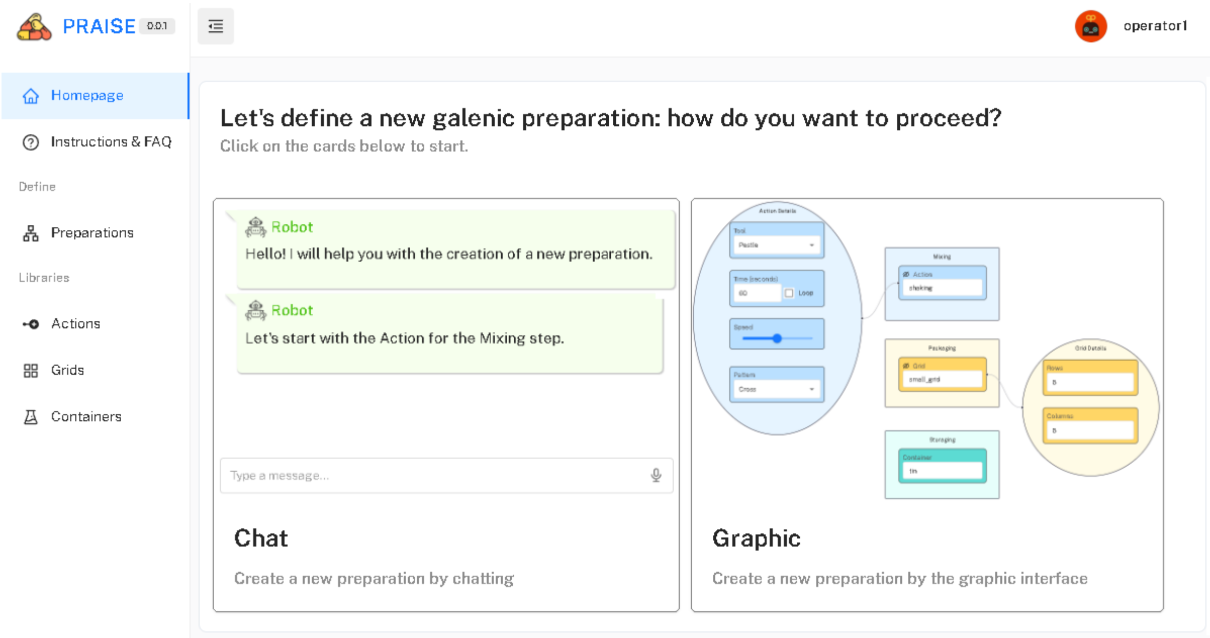


Figure 4.10: The homepage of PRAISE

The user can thus select one of these boxes to access the chat and graphic interfaces supporting the creation of robot tasks. At this time, the user can also define new domain concepts, but a more linear user journey foresees the definition of the domain concepts before preparation definition. This phase can also be carried out by other users, who may set the domain concepts as shared in their community, or by an administrator or superuser.

As an instance of the robot task definition workflow, useful as a running example for presenting the user interface in detail, let us introduce an interaction scenario:

A pharmacist wants to set up a new galenic preparation to prepare a blood pressure medication without a particular excipient present in similar medications on the market to which the patient is allergic. Before defining this preparation, the pharmacist decides to define three new domain items that will be used in the preparation: a new shaking action to mix the ingredients, a new operculator grid having 10 rows and 10 columns to place the void capsules, and a new container (a tin) where to place the filled capsules for delivery to the patient. Then, during the interaction with the chat for the preparation definition, the pharmacist changes their mind and decides to use a new, smaller grid having 6 rows and 6 columns instead of the previously defined and available in the relevant library. Similarly, when the pharmacist visualizes the defined preparation in the graphic interface, they decide to change the selected container and use a new one (a pulvis), adding it to the corresponding library.

Domain definition

To define the domain concepts before preparation definition, the user must select one of the links *Actions*, *Grids*, and *Containers* in the left side menu. In fact, the links allow the user to reach the pages that list the already created mixing actions, operculator grids, and containers, respectively, and permit them to define a new action, grid, or container.

Figure 4.11 shows the details of the *Action* that the user is defining for the *Mixing* step. A mixing action is characterized by its *Name*, the *Shared* attribute, to give other users the possibility to use the action in their preparation, and some optional *Keywords*. The latest will be used as synonyms to refer to the same object while interacting with the chat; this feature aims to make the application even more user-centered by giving the possibility to customize the individual user dictionary and thus provide a tailored interaction. Then, technical data must be provided including the action duration expressed as a number of seconds (*Time* box) or as a loop until the arrival of the user's signal (*Loop* checkbox), the action speed (set through the *Speed* slider), the movement definition (to be selected as a pattern among those available in the *Pattern* drop-down list), and finally the tool the robot must use to perform the action (to be selected with the *Tool* drop-down list). Other technical data to be provided for a mixing action are strictly related to the robot operation, like the robot itself (to be selected with the *Robot* drop-down list), and the *Height* at which it must perform the action; the latter can be acquired using the teaching feature of the robot, which consists of manually moving the arm until the desired position and register it (*Get height* feature). In a similar way, the user can define a customized pattern for the mixing action by acquiring a list of *Points*.

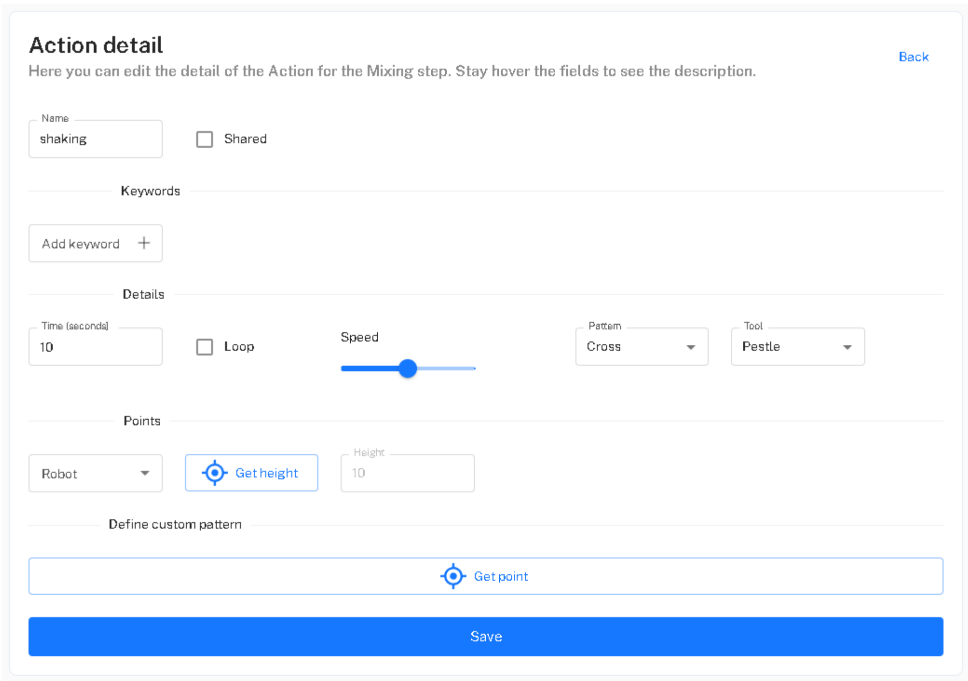


Figure 4.11: The action detail

Figure 4.12 shows the details of the operculator *Grid* to be defined for the *Packaging* step. In this case, the user must define the number of *Columns* and *Rows* of the grid, and capture a photo through the robot camera (*Get photo* feature), which will be processed by image recognition algorithms to automatically identify the external shape of the grid and the slots where the capsules will be inserted during the packaging step.

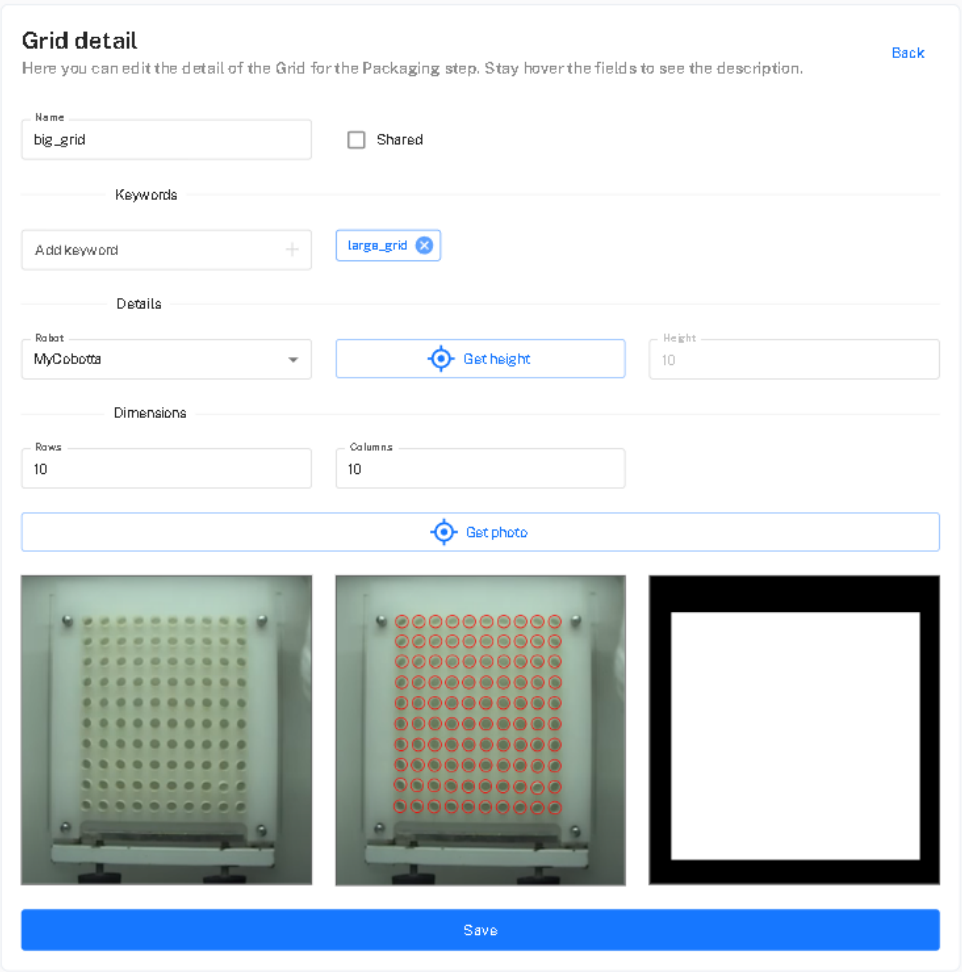


Figure 4.12: The grid detail

The last type of domain item to be defined, useful for the *Storage* step, is the *Container*. In this case, the characterizing parameter is the *Position* that the robot must reach to fill the container with capsules. It can be a fixed point acquired through robot teaching or a shape obtained after processing the container photos captured through the robot camera. Figure 4.13 shows the page used to describe the data characterizing a container: they just include the *Name*, the *Shared* attribute, the *Keywords*, and the definition of how the robot must reach the container to fill it with capsules. The latter can be a fixed *Position* acquired through robot teaching (*Get position* feature), or a shape obtained after processing the container photos captured through the robot camera (*Get photo* feature). Also in this case, the height and photos have been acquired through the robot *MyCobotta*, and the container has been called *tin*.

All the actions, grids, and containers defined by a logged user (or shared by other users) can be accessed from the left-side menu available on each page of the application and are presented as a list in the central stage of the page. Similarly, when the user selects *Preparations* in the menu, the already defined preparations are presented in a list (Figure 4.14). Here, the user can select one of them for preparation execution or for creating a new preparation by simply reusing the data of an existing preparation, possibly changing its parameters.

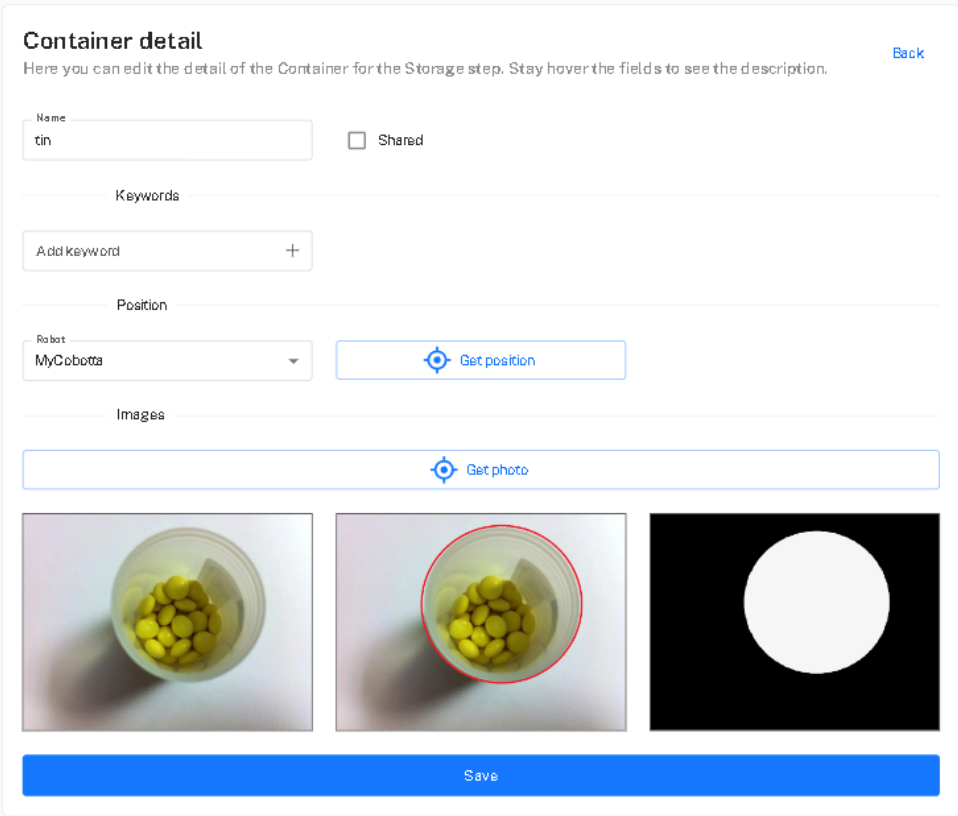


Figure 4.13: The container detail

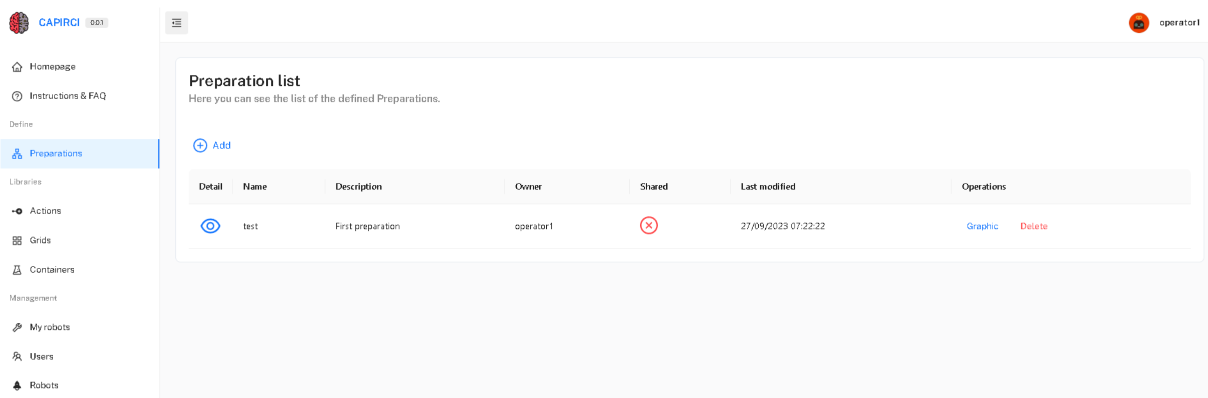


Figure 4.14: The list of the defined preparations

Defining preparations

As previously explained, the user can define a galenic formulation preparation by interacting in natural language with a chat-based interface. As one may observe in Figure 4.15, the system starts the conversation by suggesting what the user must say to instantiate the preparation steps. Textual messages are exchanged with the *Adapter* module, which in turn interacts with the NLP engine. The conversation may also occur by talking and listening to the answers, selecting the red button at the top right of the interface.

In the example shown in Figure 4.16, the pharmacist has decided to use the action *shaking* previously defined. For the grid, instead of the *big_grid* item, the user has decided to use a smaller grid with 6 columns and 6 rows. For the latter, the user has indicated the

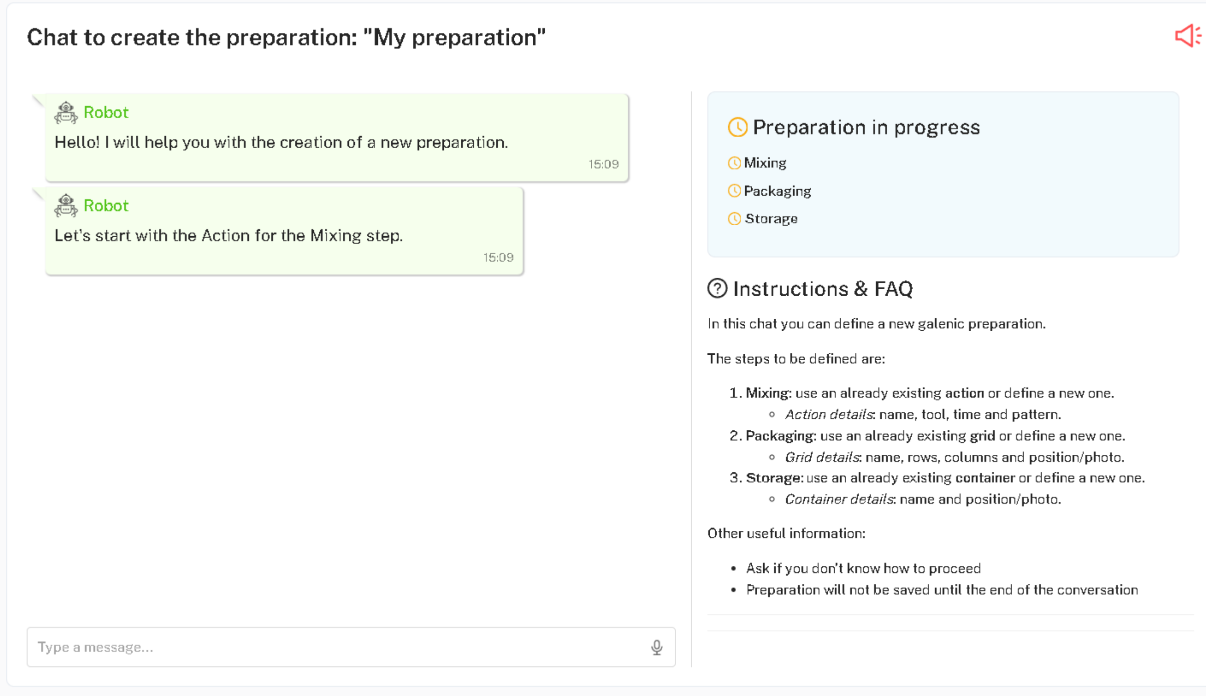


Figure 4.15: The chat interface

name *medium_grid*, the number of rows and columns, and has given a command to take a photo of the grid. On the right of the chat, a live report on the preparation in progress and some useful information are provided.

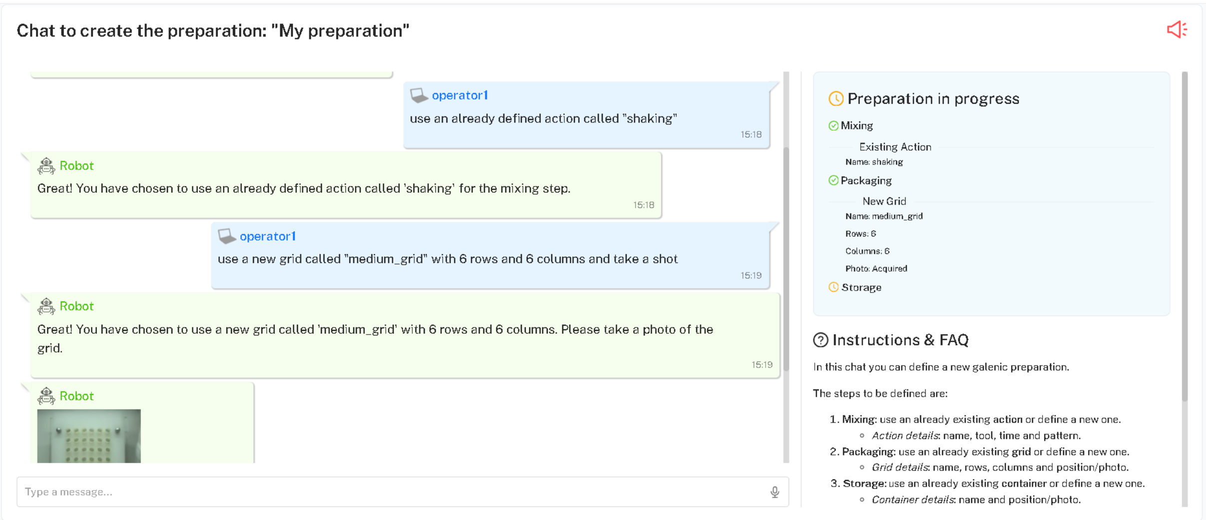


Figure 4.16: The preparation in progress

When the preparation definition is completed with the specification of the container for the Storage step (a tin, in our scenario), the chat provides a summary of the defined preparation. This gives a first opportunity to the user to control the correctness of the LLM's output and possibly require a modification. Then, the system redirects the user to the graphic interface (Figure 4.17) where the user can check the correctness of the preparation and possibly modify it through a different interaction style. In this way, potential biases or errors in the LLM's output are subject to a double-check, and the user can have full control of the robot task definition.

In the graphic interface, a preparation is represented by three main central blocks that correspond to the *Mixing*, *Packaging*, and *Storage* steps. The action, grid, and container appearing in the blocks can be modified by selecting from a drop-down list another element available in the respective libraries. For the *Action* and the *Grid*, a panel with details can also be opened on user selection as shown in Figure 4.17. These details can be modified for the current preparation without changing the underlying library data. At this stage, the user can also create a new action, grid, or container by writing a new name and clicking on the *Add* button that will appear when the name is not found (see Figure 4.18). According to our scenario, the pharmacist decides at this point to change the container previously selected for the Storage step and define a new container called *Pulvis*.

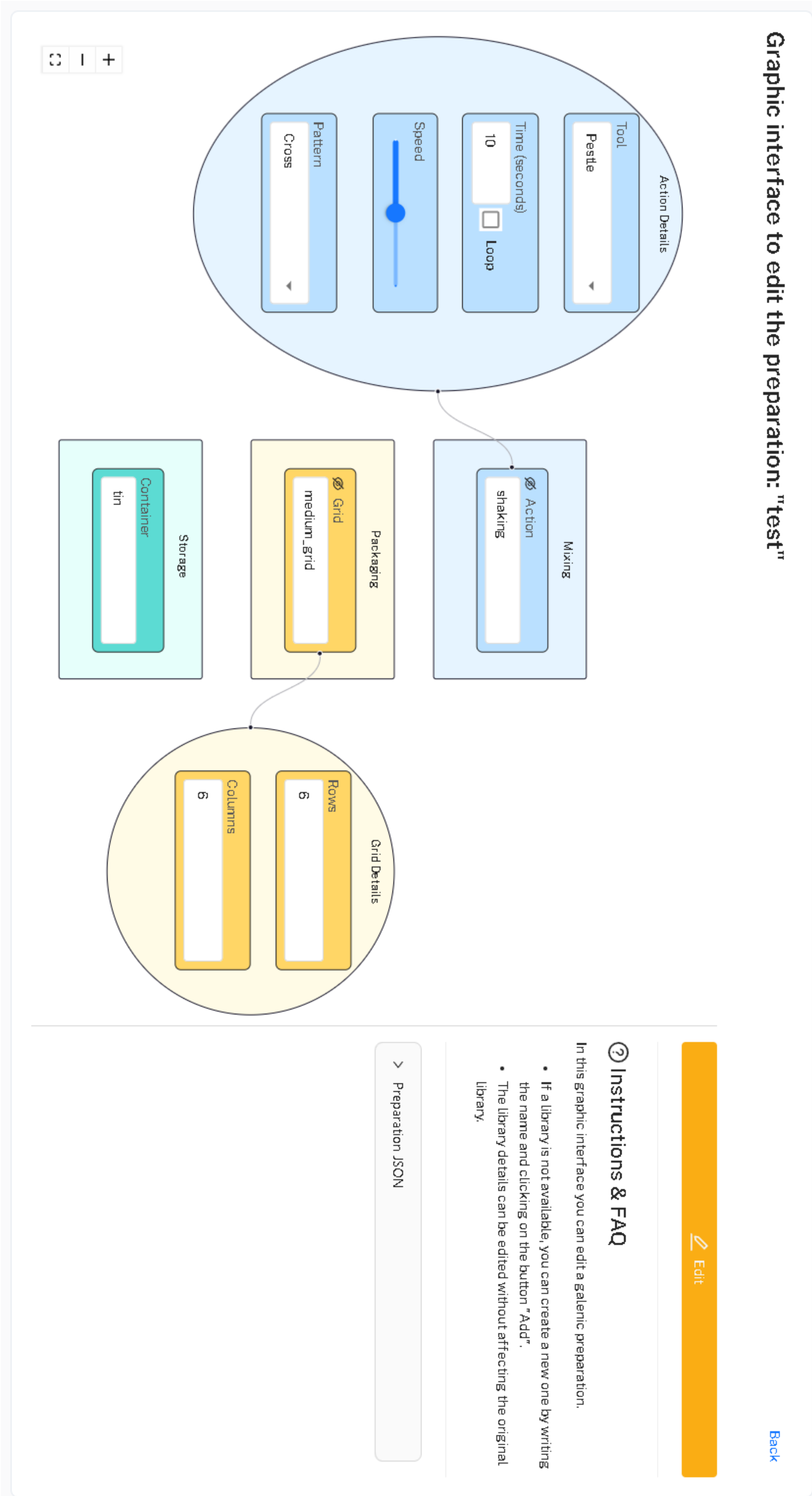


Figure 4.17: The graphic interface

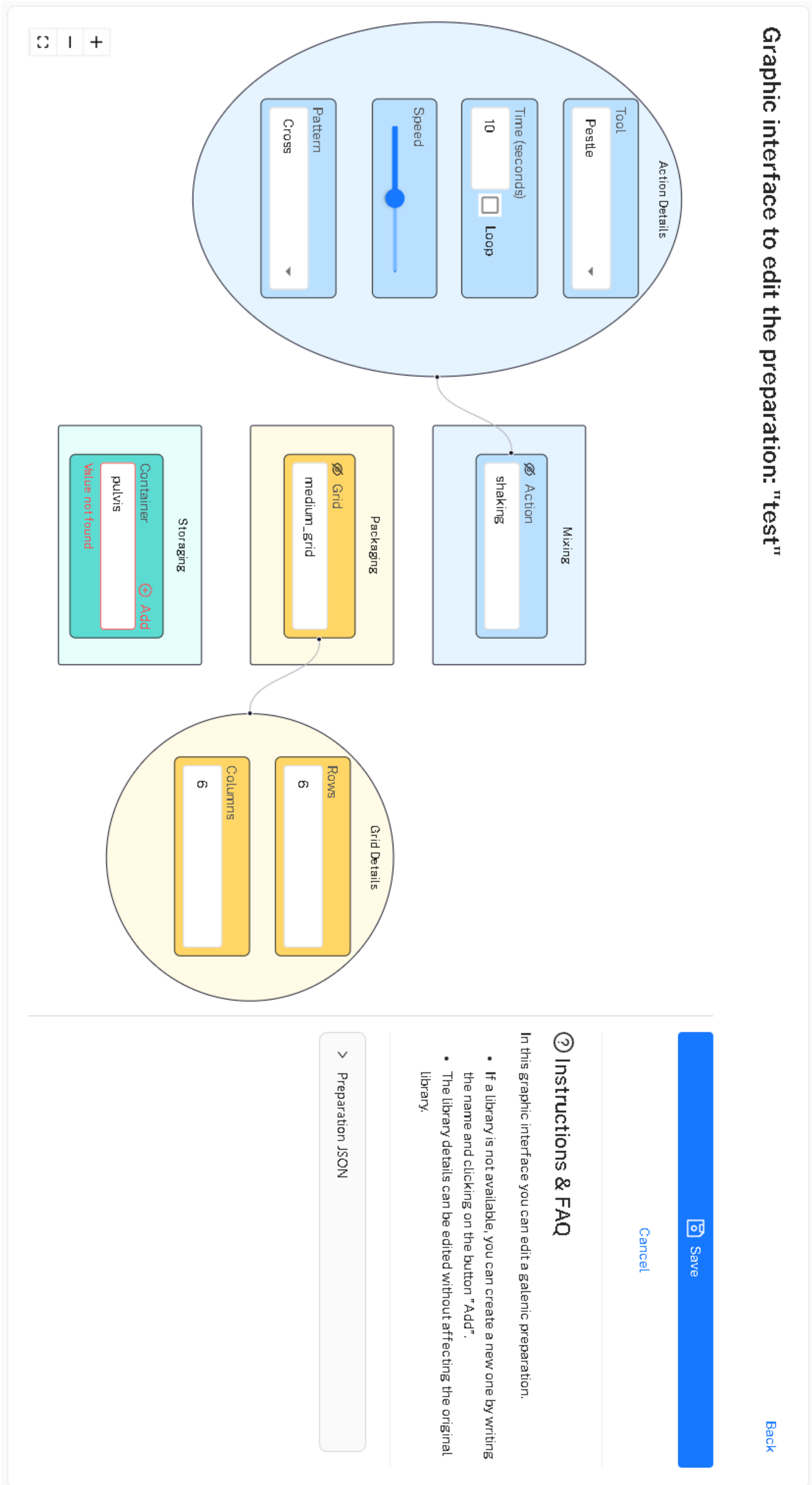


Figure 4.18: The graphic interface while adding the new container

4.5.4 Robot Program Generation

Once the preparation is defined, either through the chat or the graphic interface, it is saved as a JSON object in the database. The JSON object describing the task reported in Figure 4.17 is shown in Listing 4.1. This object includes all the necessary information for the execution of the task by the robot. Particularly, we created a program template that, based on the JSON object, is instantiated at runtime to deal with the case at hand; this instance includes the elementary robot commands needed to accomplish the task, like moving to a specific position or searching for an object in the working area. Using the JSON object as a high-level specification of the robot task enables the generation of elementary robot actions at runtime based on the specific robot characteristics. This allows the task definition to be applicable to different robots, provided that suitable program templates are developed for the different contexts.

In practical situations, the correct execution of a robot program requires taking into account possible issues related to robot calibration, collision avoidance, and singularities[4]. In the case at hand, the user does not need to explicitly deal with the problem of collisions, since avoidance of dangerous collisions is a native feature of collaborative robots. Furthermore, we assume that, for each task to be carried out, the robot operates in a predefined workspace, where objects are placed in fixed positions. If the setup of the workspace is correct, the robot program will not lead to collisions at runtime. If the setup is not correct or the user interferes with the robot movements, the native collision avoidance feature will stop robot operation. Singularities might, in principle, occur during action definition; however, in case the user chooses an available movement from the *Pattern* drop-down list, this is assumed to be free of singularities since it has been defined by the developer of the EUD environment, who is assumed to be technically competent. If, instead, the user decides to define a customized movement through direct robot teaching, the collaborative robot will allow the user to impart only feasible movements. Finally, we assume that robot calibration is performed by experts in robotics; thus, this aspect is out of the scope of the end users' activity.

[4]Singularity points in Robotics are robot positions where movement becomes unpredictable or restricted, making control less effective or unstable. This happens because at these positions the robot's joints and links align in a way that reduces its ability to move or exert forces in certain directions.

Listing 4.1: JSON structure of a defined task

```
{
  "mixing": {
    "tool": "P",
    "time": "10",
    "pattern": "C",
    "speed": 2,
    "actionName": "shaking",
    "actionId": 6,
    "hideDetails": false
  },
  "packaging": {
    "rows": 6,
    "columns": 6,
    "gridName": "medium_grid",
    "gridId": 22,
    "hideDetails": false
  },
  "storaging": {
    "containerName": "tin",
    "containerId": 7
  },
}
```

4.6 User Study

A user study has been carried out to collect feedback from users and assess the feasibility of the proposed robot programming approach to prepare galenic formulation preparations.

4.6.1 Methodology

We decided to collect mainly qualitative data in order to evaluate the user experience during the interaction with PRAISE and to analyze the challenges and problems encountered by the users. To this purpose, three different analysis tools have been applied:

- direct observation of participants' behavior during task execution and participants' comments; the notes I took while managing the user study were subsequently examined by one of my supervisors and me to identify relevant themes according to an inductive and latent thematic analysis [25]. Each theme led to identifying recommendations for system improvement;
- the UEQ (User Experience Questionnaire) questionnaire [89] (see Appendix A.1) to assess the user experience, to be filled in by participants immediately after the conclusion of the test session;
- a semi-structured individual interview with each participant, carried out at the end of the test session. The interview, organized around 6 questions aimed at gathering additional opinions concerning the effectiveness, efficiency, and overall

user experience of the application. Furthermore, the feasibility of using a chat-based interface for robot programming in relation to safety and security was explored. We were also interested in gathering information about further activities in the healthcare domain that collaborative robots might support. Lastly, participants were asked to provide additional comments, if any, on the proposed solution. Also, in this case, the answers collected were examined by two researchers, applying an inductive and latent thematic analysis to identify further and more general themes to be considered for the design of similar systems and future work.

Nine participants (5 females and 4 males) have been involved in the user study. All the participants work in the pharmaceutical field, with different backgrounds, specializations, and experiences. Table 4.4 reports the demographic data of participants, their profession, the number of years in the current role, their knowledge of galenic preparations (on a scale from 1 - very low, to 5 - very high), and their knowledge of computer technologies (on a scale from 1 to 5).

ID	**Age**	**Gender**	**Profession**	**Work experience *(years)***	**Galenic experience *(1-5)***	**Computer proficiency *(1-5)***
P1	27	Male	PhD student in virology	3	4	4
P2	24	Female	Pharmacist	1	4	3
P3	30	Female	Pharmacist	5	3	4
P4	27	Male	PhD student in chemistry	2	3	3
P5	30	Male	Pharmacist	4	5	4
P6	33	Male	Pharmacist	7	4	4
P7	33	Female	Pharmacist	5	5	3
P8	33	Female	Nurse assistant in pharmacy	10	1	4
P9	45	Female	Pharmacist	18	3	3

Table 4.4: Partecipants data

Participants in the user study were asked to execute five tasks of increasing complexity related to the definition of domain concepts and robot programming through the chat and the graphic interfaces (Figure 4.19 reports the details of the assigned tasks).

Before starting the experiment, participants were invited to an introductory session of about 5 minutes. In this introduction, I described the goal of the study, carried out a brief presentation of the prototype, and illustrated the assumptions about the preparation of galenic formulations. The think-aloud protocol was used during the tests, thus participants were encouraged to talk during task execution, making explicit their thoughts.

During task execution, I observed each participant and annotated their comments and significant behaviors, such as difficulties while interacting with the prototype and errors made in the task execution. Upon completion of the test session, I asked the participant to fill in the UEQ questionnaire and conducted the interview with the participant.

The user study was conducted through remote moderated tests using Google Meet, Google Drive, and Google Forms. Google Meet was used for video calling and screen

1. Define a new Action for the Mixing step with these details:
   - Name as desired (try to remember it).
   - Shared: False.
   - Don't define any keywords.
   - Time: 60 seconds.
   - Loop: False.
   - Speed: Medium.
   - Pattern: Cross.
   - Tool: Spatula.
   - No points acquired from the robot.
2. Define a new Grid for the Packaging step with these details:
   - Name as desired (try to remember it).
   - Shared: False.
   - Select the first robot 'MyCobotta' and acquire the height of the operculator grid.
   - Don't define any keywords.
   - Rows: 5.
   - Columns: 5.
   - Acquire a photo of the grid.
3. Create a new preparation through the chat with these details:
   - Name as desired.
   - Description: 'My first preparation'.
   - Shared: False.
   - Follow the chat requests and use the Action you defined for the Mixing step and the Grid you defined for the Packaging step. For the Storage step, define a new Container with a name as desired directly in the chat and acquire a photo of it.
   - At the end, confirm the summary if everything has been understood correctly by the chat.
4. The preparation defined through the chat will automatically be opened in the graphic interface. Please:
   - Check whether all details have been correctly defined.
   - Change the time of the Action from 60 to 120 seconds and the tool from Spatula to Pestle. For the Grid, change the number of rows and columns from 5 to 6.
5. Define a new preparation through the graphic interface:
   - Use the Action you defined for the Mixing step.
   - As a Grid for the Packaging step, try to use a not already defined grid and define it through this interface.
   - Use the Container you defined for the Storage step through the chat.

Figure 4.19: Tasks performed during the user study

sharing, Google Drive was used to share the document containing the list of tasks to be performed, and Google Forms was used to collect user data and UEQ results. PRAISE was uploaded on an Amazon AWS cloud machine, and its address was made public and reachable to everyone. Robot manipulations for point and photo acquisition were only simulated (i.e., when clicking the related buttons, already stored data were retrieved). This test modality allowed us to recruit a good number of representative users of the target user population. Each test session lasted from 30 to 40 minutes, including the introductory session.

A pilot study with one female participant, a 28-year-old PhD student in pharmacology, was performed before the user study to validate the test protocol and refine the description of the submitted tasks. This pilot test was carried out in person in order to have a more comprehensive discussion and collection of feedback.

4.6.2 Ethical Considerations

The study involved only user interviews and non-sensitive interaction data. All data were collected anonymously, handled with strict confidentiality, and informed consent was collected, in line with ethical standards and with no risk of harm to participants. At the time of the study (September 2023), the ethical committee of our university had not yet been established, so no formal approval was required.

4.7 Results

The results obtained from direct observation, the UEQ questionnaire, and final interviews are detailed below.

4.7.1 Findings from Direct Observation and Participants' Comments

The following themes and related recommendations for system improvement emerged from the thematic analysis of data gathered during task execution. Table 4.5 summarizes the themes that emerged from the thematic analysis with their corresponding codes.

Discoverability - Task 3 required using the chat to create a new preparation. Three participants were uncertain about how to initiate the conversation. They asked *"What should I tell it?"* or *"How should I tell it?'*. This challenge is connected to the familiar discoverability issue [110], which historically impacted command-line interfaces and can now affect conversational interfaces as well [35]. This issue arose despite the existence of initial guidance provided in the chat upon opening the page (see Figure 4.15). In particular, the messages *"Hello! I will help you with the creation of a new preparation"* and *"Let's start with the Action for the Mixing step"* was manually added by the developer to the chat history and was not generated by the LLM system, with the intention of facilitating the beginning of the conversation. These messages were, however, interpreted as part of the graphical user interface, as they were already present as soon as the chat opened, without any animations, like a fixed element. Consequently, users ignored these messages and did not consider them part of the conversation. Still referring to this problem, another user suggested adding a section devoted to Frequently Asked Questions (FAQs).

Table 4.5: Themes and codes emerged from the thematic analysis of the direct observation and participants' comments

Theme	Codes
Discoverability	• Uncertain initial interaction • Initial guidance • Messages as graphic elements • Visibility of graphic elements
Gap between user's mental model and system	• Assumptions made • Connection with the real world • Language of the user
Robustness of natural language understanding	• Mother-tongue language preferred • Ambiguous names • Chat flexibility
Non-deterministic behavior	• NLP misinterpretations • Lack of guidance • Lack of task summary
Interaction with the graphic interface	• Easy to use • Efficiency • Interaction steps • System feedback

As to the interaction with the graphic interface, a discoverability problem occurred when the users had to change the name of an item, defining a new one. The *Add* button only appeared after a new (still unused) name of the item had been typed. Until there was no text in the field, the button remained invisible.

Recommendations: Instructions on how to start a conversation with the chat must be made more visible in the interface, as well as a different mechanism to add new items through the graphic interface must be studied. The development of FAQs should be considered as well.

Gap between user's mental model and system conceptual model - Notwithstanding the initial introduction presenting the application and its underlying assumptions, two users encountered difficulty, particularly in Task 3, in establishing a clear connection between the operations carried out in the application and the actual laboratory workflow. One of these users said *"Sorry, maybe I didn't understand how it is supposed to work"*. However, after a further quick explanation of the application and its aims, there were no more issues in proceeding with the tasks. Also concerning this topic, one user reported that the term *Library* used to name a category of items in the domain was unclear and far from the terminology used for galenic preparations.

Recommendations: The problem related to the term *Library* can be easily solved, even though only one user underlined that it was unclear. The difficulty in creating a connection between the application and the real world requires further investigation when the actual integration of a collaborative robot in the laboratory workflow is complete.

Robustness of natural language understanding - The NLP engine demonstrated robustness in many areas. Six participants opted for Italian as their preferred language during the interaction with the chat, rather than the originally proposed English language. These users always first asked *"Does it understand if I speak to it in Italian?"* and were pleasantly surprised at the affirmative answer. Two participants even changed their language preference mid-way through the interaction, yet the system swiftly adapted to the new language without any issues. A user tried to ask, out of curiosity, what material the container was made of, and the chat reasonably replied *"glass"*. The NLP engine also exhibited flexibility in terms of the arrangement and presentation of information. For instance, during the execution of Task 3, a user began with the *Storage* step, rather than following the suggested sequence, but this did not hinder the preparation definition. Furthermore, participants defined the different steps either by providing one detail at a time or by providing all the required details together, and this did not bring about any problems. The only issues that emerged were due to ambiguous names used for the definition of domain items. For example, three users named *Mixing* the action for the *Mixing* phase, thus overloading the term and giving rise to possible ambiguity. The NLP engine remained consistent in its responses, as was evident when it confirmed: *"Ok, I will use the Mixing action for the Mixing step"*, using the overloaded term for both meanings correctly. In front of this message, users recognized the ambiguity that they had created and looked for an explanation, checking the side panel of the current preparation. One user acknowledged: *"Ah ok, it is actually right. I messed up. It is as if you give a person the name 'Name' "*.

Recommendations: The NLP engine used in our system and its adaptation to the preparation of galenic formulations with a robot result in being sufficiently robust; however, additional scenario-based tests would be useful to investigate how users explore the interface and converse with the chat in their daily work routine.

Non-deterministic behavior - We observed that the LLM-based approach sometimes led to non-deterministic behaviors, but without hindering task completion. Examples of these events include ambiguous names and an occasional lack of guidance or task summary. For instance, the LLM was instructed to guide the user step-by-step during the conversation, providing the user with details on what and how to define. However, it occasionally failed to take the initiative, leading users to seek clarification on what action to take next. Before asking the chat, three users requested guidance from the researcher, asking *"What should I do now?"*. In these cases, it was suggested to ask in the chat, and users obtained, in this way, clear instructions on how to proceed. This situation also occurred occasionally when the system confirmed the definition of the last step without providing the user with further instructions to proceed with the conversation. However, if the user inquired about the next step, the LLM provided a task summary as it had been instructed. Alternatively, if the user indicated completion with a statement like *"OK, I'm done"*, the LLM correctly reached the end of the conversation without any further action.

Recommendations: Avoiding non-deterministic behaviors is currently a critical issue for LLMs. In our case, we may reduce the temperature parameter and provide users with instructions to ask the system when they are uncertain about the actions to take or the system's answers.

Interaction with the graphic interface - Overall, there were no difficulties in understanding the application features and in using the graphic interface to view and revise the preparation. One participant reported that *"Even a pharmacist with no experience in galenic preparations could do it after an initial introduction"*. Another one said that *"The graphic elements are very intuitive and quick to use"*. A user suggested reducing the interaction steps by showing the preparation in the edit mode as soon as it is opened. Three users did not immediately understand the behavior of the interface for item creation due to the lack of system feedback related to the robot's height acquisition.

Recommendations: Since efficiency must be balanced with robustness, further tests are necessary to evaluate whether the suggestion to speed up the process could be suitable for all users; then, it is important to add system feedback to confirm the robot's data acquisition.

4.7.2 Findings from UEQ Questionnaire

Considering the UEQ benchmark ranges, which classify results as *Bad*, *Below average*, *Above average*, *Good*, and *Excellent* [132], the obtained results proved that interacting with our prototype application was very satisfying for the participants in the study (see Figure 4.20). Indeed, Attractiveness, Dependability, and Stimulation are in the *Excellent* range of the benchmark, Efficiency, and Novelty are in the Good range, while only Perspicuity is in the *Above average* range of the benchmark. The most appreciated aspects were Attractiveness (M=2.04, SD=0.45) and Dependability (M=2.11, SD=0.38).

As to Perspicuity having a lower evaluation than the other UX dimensions, given that the prototype is intended for professional use, it is reasonable to expect that something is not immediately clear, and that a training period can be necessary. Nonetheless, the general positive feedback indicates that there are no significant issues about comprehending and carrying out tasks after the initial use.

UEQ results can also be interpreted using Attractiveness as a pure valence dimension, Pragmatic Quality (that includes Perspicuity, Efficiency, and Dependability and covers all the goal-directed aspects of the user experience), and Hedonic Quality (that includes Stimulation and Novelty and concerns not goal-directed aspects of user experience). Also in this case, the analysis shows positive results, with Attractiveness equal to 2.04, Pragmatic Quality equal to 1.78, and Hedonic Quality equal to 1.68, thus all above 0.8, the considered threshold for this scale structure.

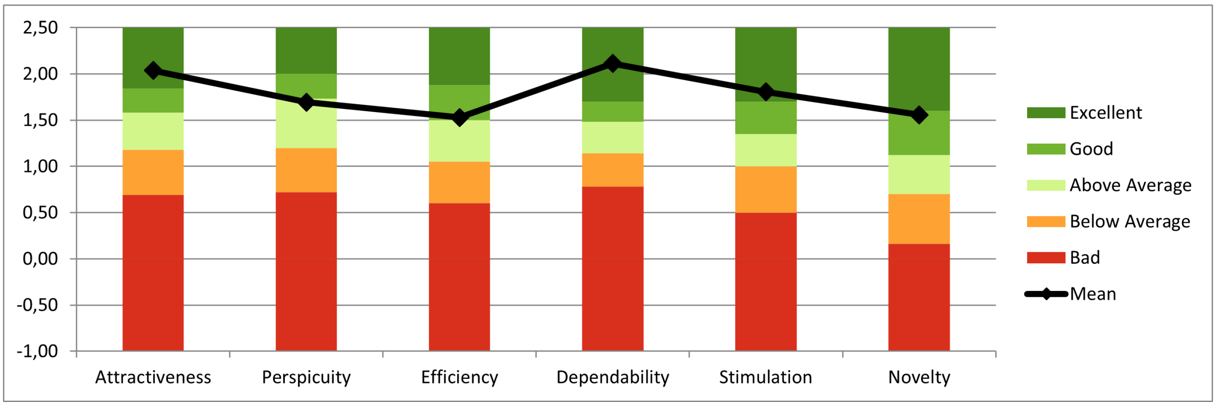


Figure 4.20: UEQ results compared with the UEQ benchmark

4.7.3 Findings from the Final Interviews

The interview questions presented to participants are reported in Table 4.6.

Further themes emerged from the thematic analysis of the data gathered through the interviews. Table 4.7 summarizes the themes that emerged from the thematic analysis with their corresponding codes.

Table 4.6: Questions of the final semi-structured interviews

1	Was it easy or difficult to understand and remember how to navigate and use the application?
2	How do you evaluate the effectiveness of the application in achieving your purpose? Did you really succeed in achieving your goal? If not, how close were you?
3	How do you evaluate the efficiency of the application in achieving your purpose? With how much effort did you manage to achieve your goal?
4	Do you think that using the chat to program the robot is robust enough, or does it leave room for error?
5	Would you see this approach to robot exploitation and programming suitable for other medical and/or pharmaceutical problems as well?
6	Do you have any other comments regarding the application, approach, technology, or anything else?

Table 4.7: Themes and codes emerged from the thematic analysis of the final interviews

Theme	**Codes**
Learnability	• Easy to use • Familiarity with technology • Older pharmacy staff
Usefulness and innovation	• Engaging interaction based on natural language • Effective approach to galenic preparation • Robot suitable for repetitive tasks
Graphic vs chat-based interface	• User preference for graphic interface • Mimicking work practice • Efficiency of the graphic interface • Correct message understanding • Human-like interaction • Rigid step-by-step interaction
User's trust in the proposed technology	• Combining chat and graphic interfaces has potential • Correct understanding of ambiguous messages • Chat reliability
Potential future applications	• Potential applications • Additional features • Additional steps in the process

Learnability - All participants reported that PRAISE was easy to use, albeit after a learning activity at the beginning. Six users stated that they quickly grasped how it worked despite encountering common challenges associated with trying out novel software applications. In this respect, one user commented: *"I generally encounter some difficulties initially when dealing with novel software applications"*. Two users expressed concerns about implementing this approach with older users who might not be familiar with software technologies. In this regard, one participant reported: *"In my lab, I am by far the youngest; everyone else is almost 50 years old, and I don't know how they would react to such a system"*.

Recommendations: some users may encounter difficulties due to the way natural language processing is implemented in chat interfaces and how it interprets their requests. Users might struggle to phrase queries in a way that the system understands, or face unexpected behavior with more complex inputs. Providing brief guidance on effective interaction strategies and ensuring the system can handle flexible or imprecise input can help improve usability and learnability.

Usefulness and innovation - Users were pleasantly surprised by the innovative approach based on natural language interaction. They also appreciated the idea of addressing the entire production process related to galenic formulation preparation, instead of considering only a single step, as often happens with other technologies. In this regard, one participant reported *"This prototype about galenic formulations is useful and innovative, right not to leave this procedure entirely manual"*. The support of collaborative robots generated significant interest in the participants, who confirmed the presence of some repetitive and tedious steps during the production process of galenic medicines.

Recommendations: to guarantee a thorough comprehension of the requirements, it is crucial to consult with domain experts. They can provide valuable insights that may not be immediately apparent to the developers.

Graphic vs chat-based interface - Two participants stated a preference for the graphic interface due to its efficiency, practicality, and resemblance to the worksheet used for galenic formulation preparation in the laboratory. For example, a participant commented: *"The chat is comfortable, but I prefer the graphic part because it is more schematic and clear and it is easier to translate it into the real context than the chat"*. The graphic elements were highly valued for their effectiveness by all the participants. The other seven participants were intrigued by the chat function, which demonstrated great computational power when compared with other technologies based on natural language interaction they had previously experienced. As one participant reported, *"It really understands even if I misspell something"*. One user suggested making the interaction with the chat faster. Indeed, during the task execution, he used a step-by-step approach, but during the interview, he was informed that there was also the possibility of defining everything at once and just with voice (not only with written text). The participant expressed enthusiasm when informed of this possibility.

Recommendations: the use of natural language interaction is becoming increasingly common. However, in some cases, a graphical interface is necessary to provide a clear and concise representation of data. It is important to avoid relying solely on the latest and most powerful technologies, as this may lead to issues with efficacy (see *Einstellung effect*[5]

[5] *Einstellung* [98] is the development of a mechanized state of mind. Often called a problem solving set, *Einstellung* refers to a person's predisposition to solve a given problem in a specific manner even though better or more appropriate methods of solving the problem exist.

or *Maslow's Hammer*[6]). It is also crucial to leave users the possibility to choose their preferred interaction mode. Regarding the chat interface, it aims to simulate human-human communication as closely as possible. To achieve this, it is important to consider the various scenarios that can arise during human interactions, such as consecutive messages, interruptions, and the possibility of recovering from misunderstandings.

User's trust in the proposed technology - All participants appreciated the effectiveness, robustness, and flexibility of the proposed solution. One participant commented that she trusted the application and the chat once she observed a correct understanding of the input. The coexistence of both the natural language and graphic interaction methods was also valued for the security it provided, as well as for the flexibility to select the preferred approach based on individual needs.

Recommendations: providing users with a sense of security from the beginning when using an application is crucial. Otherwise, they may begin to doubt even when the application responds correctly, resulting in additional mental effort and failure to support the user.

Potential future applications - The participants suggested other potential applications of this approach within the healthcare sector. One participant outlined that using a collaborative robot endowed with an intuitive programming interface may be beneficial in accurately dispensing the correct dosage of antibiotics for a predetermined treatment. This would lead to patients avoiding the overuse of antibiotics, thus reducing the development of antibiotic resistance. A participant proposed the implementation of a multipurpose robot that could perform various operations, including diluting the SARS-CoV-2 vaccine and other generic preparations, such as the preparation of drips in a hospital. Additionally, three participants recommended the integration of the robot-based solution with a digital prescription system exploiting an optical character recognition (OCR) component. One participant advocated the use of collaborative robots to help order medicines by expiration date on the shelves of the pharmacy; indeed, this is a very tedious activity currently performed manually by pharmacists or their assistants. Finally, a participant highlighted that cannabis-based products involve more steps in the production process than galenic preparations. Accordingly, he proposed that in such cases, a cobot could be even more advantageous.

Recommendations: when designing a novel technology, it is important not to simply tailor it to the specific case, but also to imagine from the beginning whether it is applicable to other contexts and make it as transversal as possible.

4.7.4 User Study Results Summary

The user study allowed us to demonstrate the validity of the proposed multimodal approach to robot task definition by domain experts (pharmacists in our case). Natural language interaction attracted and engaged the users, especially when they discovered that the system was able to understand mixed-language sentences and to answer reasonably off-topic questions. In addition, the possibility of using the LLM as an assistant that helps users formulate their requests made the interaction goal-effective and supported a smooth learning of the system.

When using an AI-based system, there is always the risk that the user feels controlled

[6]The law of the instrument, law of the hammer [26], Maslow's hammer, or golden hammer is a cognitive bias that involves an over-reliance on a familiar tool. Abraham Maslow wrote in 1966, It is tempting, if the only tool you have is a hammer, to treat everything as if it were a nail." [101]

by the system rather than vice versa. PRAISE did not exhibit this behavior, but allowed the users to express their requests according to their preferences and habits. This was ensured by the prompt engineering activity performed by the developer: prompts were properly tuned to balance human-like creative conversation with deterministic behavior.

Furthermore, beginning a conversation with a natural language interface is a widely recognized issue in the literature [110] [35] due to the discoverability problem. Our study suggests that its impact can be mitigated by providing users with a simple explanation of the system functionality and how requests must be formulated. For instance, an example of chat messages that can be exchanged could be provided at the beginning of the conversation to help users understand the appropriate level of abstraction and freedom in sentence formulation. In addition, the LLM can be instructed to act as an assistant if the user is uncertain about how to proceed with the conversation, providing instructions to guide the conversation proactively. This turns out to be beneficial when structured tasks like programming must be carried out. Further research is needed to specify which initial instructions should be given to the LLM to always proactively guide the user during the conversation, which is aimed, in our case, to define robot tasks, but which can have different goals in other contexts.

The study also allowed us to acquire information about the difficulties encountered by end users, which have been reformulated in terms of recommendations for system improvement. For instance, the results of our study outlined once again the importance of Nielsen's usability principle *Match between the system and the real world* [109], which becomes crucial when interaction exploits natural language in relation to a specific application domain. In the context of human-AI interaction, not only must the user be able to comprehend the terminology used by the AI-based interface, but it is also mandatory that the AI counterpart "understands" the users' requests. Otherwise, misunderstandings may occur, thus hindering the effectiveness and usefulness of the AI technique.

As to the non-determinism inherent to LLMs, which may lead to nonsensical or wrong answers (also known as *hallucinations*), we proved that proper fine-tuning through the *Adapter* module permitted us to mitigate or totally avoid (as in our case) such behavior. While additional tests would be useful to confirm this result, we also remark that the proposed multimodal approach allows the user to discover and solve possible hallucinations using the graphic interface, thus always keeping control over the system output.

4.8 EUDability Evaluation

To complement the analysis described in Section 4.7, PRAISE was evaluated in terms of EUDability, a concept originally proposed by Barricelli et al. [15] to assess the extent to which a system supports end-user development activities. The concept of EUDability is discussed in detail in Section 2.5. The evaluation methodology proposed in [16] integrates two complementary assessments: an inspective evaluation of the system's EUDability dimensions conducted by HCI experts, and an empirical assessment of users' Computational Thinking (CT) skills through the Computational Thinking Scale (CTS) [143] properly adapted to EUD activities. This dual approach enables a comprehensive evaluation of the fit between system capabilities and user competencies. The EUDability inspection was conducted through a predictive inspection-based method, following the 15-item checklist proposed in [14] (Table 4.8).

Three HCI experts familiar with EUD principles independently assessed PRAISE across all five EUDability dimensions. The experts were provided with a comprehensive

Table 4.8: Checklist for EUDability predictive inspection-based evaluation [14]

Dimensions	**Items to check *(Yes/No)***
Concreteness	The EUD environment: **1.** represents the domain concepts as elements for problem solving **2.** does not require knowing the low-level details of each element **3.** uses the language that is more familiar to the end user
Modularity	The EUD environment allows: **4.** the identification of elements that may compose a problem solution **5.** organizing the elements in meaningful categories **6.** freely exploring each category of elements
Structuredness	The EUD environment suggests: **7.** the steps to be performed for problem solving **8.** the order of the steps to be performed for problem solving **9.** the connectedness between different steps of the problem solution
Reusability	The outcome resulting from an EUD activity can be: **10.** saved in a library **11.** included in another EUD project **12.** modified to be adapted to another EUD project
Testability	The outcome resulting from an EUD activity can be: **13.** executed within the EUD environment **14.** inspected by the user **15.** represented in alternative ways

description of the system's goals, supported tasks, and target users (pharmacists) before conducting their individual evaluations. Following the individual assessments, a debriefing session was held where the experts discussed their findings and reached consensus on the final ratings for each dimension as follows:

- The **Concreteness** dimension has been assessed as *High*, since the main elements refer to domain concepts, the user does not have to know the low-level details related to robot configuration and robot movements, and the terms used to name the elements belong to the pharmacists' language.

- The **Modularity** dimension has been assessed as *High*, since the elements to create robot tasks for a galenic formulation preparation can be easily identified, retrieved, and freely explored.

- The **Structuredness** dimension has been assessed as *Mid*, since the chat and graphical interfaces are organized in three steps necessary to create a robot task, and these are shown in the interface in a given order, and progressively updated; however, it is unclear if these steps are connected.

- The **Reusability** dimension has been assessed as *High*, since the robot tasks (i.e., the "preparations") can be saved, reused, and modified to create different robot tasks.

- The **Testability** dimension has been assessed as *Mid*, since the execution of the robot task cannot be simulated within the EUD environment, but the graphic interface can be used to inspect the defined task and possibly modify it; furthermore, the graphical representation is an alternative way to the chat to describe the robot tasks.

These assessments indicate the minimum required CT skill levels that users must possess to effectively utilize PRAISE. When an EUDability dimension is rated as High, users need only Low-level CT skills in the corresponding area, as the system provides strong support. Conversely, Mid-level EUDability ratings require Mid-level CT skills from users. Table 4.9 summarizes the mapping between EUDability dimensions and their corresponding CT skill requirements for PRAISE.

Table 4.9: Required CT levels for the EUD environment devoted to pharmacists

EUDability	**Level *(Low/Mid/High)***	**Min. CT level *(Low/Mid/High)***	**CT skill**
Concreteness	*High*	*Low*	Abstraction
Modularity	*High*	*Low*	Decomposition
Structuredness	*Mid*	*Mid*	Algorithm Design
Reusability	*High*	*Low*	Generalization
Testability	*Mid*	*Mid*	Evaluation

To complete the evaluation of PRAISE EUDability, the CTS questionnaire was administered to 20 healthcare professionals: 12 participants were pharmacists, 7 were enrolled in a Nursing B.Sc. course, and 1 was a medical doctor and also a university professor. The CTS assesses five fundamental CT skills through a 5-point Likert scale questionnaire, with items adapted from the educational context discussed in [143] to workplace problem-solving scenarios that foster EUD activities. The five CT dimensions (Abstraction, Decomposition, Algorithm Design, Generalization, and Evaluation) evaluated correspond directly to the EUDability dimensions. The results of the CTS administration showed that all five CT skills reached a *High* level. Statistical analysis using χ^2 tests on the pooled responses from all healthcare professionals in the study confirmed that the distribution was significantly skewed toward high competency levels across all dimensions ($p < .001$ for all subscales). This finding indicates that healthcare professionals possess strong computational thinking capabilities across the spectrum of skills required for end-user development activities.

The unification of the EUDability assessment with the CT skills evaluation demonstrates a favorable fit between PRAISE and its target users. Since the system requires only Low to Mid levels of CT competencies (as indicated in Table 4.9), while users possess High levels across all five CT dimensions, there is a clear alignment between system demands and user capabilities. This match suggests that pharmacists should be able to effectively utilize PRAISE for programming collaborative robots in galenic preparation tasks without requiring extensive training or experiencing significant cognitive barriers. In particular, the High rating in Concreteness, Modularity, and Reusability dimensions indicates that PRAISE successfully abstracts robot programming complexity and presents it through domain-familiar concepts, while the Mid ratings in Structuredness and Testability are adequately compensated by pharmacists' strong Algorithm Design and Evaluation skills.

The complete results of this evaluation are reported in *"Towards the Unification of Computational Thinking and EUDability: Two Cases from Healthcare"* published in the proceedings of the *17th International Conference on Advanced Visual Interfaces (AVI) 2024 (Arenzano, Genoa, Italy)* [16].

4.9 Discussion

This section discusses the findings, open issues, and future directions derived from this research, with a focus on RQ1, which investigates the extent to which domain experts can define domain-specific tasks and items through dedicated user interfaces designed for non-programmers.

Prompt engineering is presently a topic of great interest, with several approaches being presented. For instance, in [156], a tool for supporting the development and systematic evaluation of prompting strategies is presented. In some cases, issues may emerge due to the complexity of initial instructions to be given to the LLM and/or to its non-deterministic behaviors. For these situations, alternative approaches can be explored. One of these solutions is to break the LLM instance into multiple LLM instances, thus creating a chain, as suggested in [154]. This solution enables requests to be distributed across multiple instances, rather than relying on a single one, to enhance comprehension by the LLM through the use of shorter prompts. For future development, if, in our case, a robot task for a specific context becomes overly complex, we may consider breaking it into multiple LLM instances or following other approaches to manage such complexity.

Another point of interest is related to managing the dialogue with the user when the user prompt is misinterpreted, possibly due to non-determinism. In that case, it would be useful if the system provided the user with an explanation of why it interpreted the prompt in a certain way, thus allowing the user to refine the prompt and enforce the desired interpretation. In this sense, another essential aspect is error recovery, wherein both the user and system can offer explanations to understand each other and fix errors that have occurred. Developers typically pay attention only to the explanations provided by the system as a reaction to user input identified as incorrect. With our approach, the user can provide additional information to the system to enhance mutual understanding. In fact, LLMs undergo training on human language and conversation structure so they can improve the output by following users' directions; this could facilitate collaboration and help users make better decisions.

Furthermore, interaction with written text composition may also bring inefficiencies. A potential solution to reduce this issue would be to express the messages through voice commands, facilitating hands-free utilization of the application, thus improving productivity even while participating in other tasks. This will require assessing the comprehension level of the speech recognition engine, even when long messages are uttered.

The effectiveness of a conversation greatly affects the level of users' trust and confidence in the used technology. The multimodal approach proposed here allows users to trust the system as they can view the interaction results and fix any errors if needed. This approach has general applicability, particularly when non-deterministic algorithms are involved, as it provides users with complete control over the final output. Therefore, while it is beneficial to employ AI-based technologies, it is important to retain control in the users' hands and give the possibility to verify the system output, especially when non-deterministic behaviors may lead to errors.

Further considerations can be made about the implementation of AI technologies with older users, even within domains primarily involving expert professionals such as pharmacists. In fact, domain expertise does not necessarily correlate with high levels of digital literacy or familiarity with AI-based interaction paradigms. In many real-world settings, especially in healthcare-related professions, experienced practitioners may belong to older age groups and may have had limited exposure to recent advances in interactive AI systems. Limited digital literacy and unfamiliarity with technology may hinder interaction with LLM-powered interfaces. Another challenge arises from the diverse linguistic expressions and accents prevalent among older users. LLMs, while advanced, may struggle to accurately interpret regional dialects or speech impairments, leading to misunderstandings and frustrations. Further experimentation is needed to explore this issue and investigate how to cope with it.

An additional consideration emerges from the EUDability evaluation of PRAISE, which provided further insight into how well the system supports end users in carrying out complex task definition activities. The evaluation revealed that PRAISE offers a balanced level of abstraction suited to domain experts with limited technical knowledge. The combination of graphic and natural language interfaces facilitates the decomposition of tasks into concrete and modular steps, supporting users' procedural reasoning and reducing the need for advanced computational abstraction. Beyond the specific case of PRAISE, this outcome suggests that the EUDability analysis can serve as a general diagnostic framework for assessing whether an EUD environment truly aligns with the cognitive and problem-solving abilities of its intended users. In other words, evaluating the interplay between the system's structural affordances and users' CT skills can inform design decisions for other AI-based and domain-oriented EUD systems.

Privacy and ethics are significant concerns for AI applications. With respect to privacy, it is noteworthy that no sensitive patient data are used; moreover, the ingredients involved in the preparation of a specific galenic formulation are not disclosed to the LLM, while the sequence of preparation steps does not represent sensitive information. With regard to the ethical issue, we underline once again that PRAISE is intended to serve as an aid, rather than as a replacement for the practice of the pharmacist, who remains always in charge of the final confirmation of the operations.

4.9.1 Design Implications

The findings of this study suggest some design implications that can be applied to other systems combining natural language and graphic interfaces for end users:

- **Support for multiple interaction modalities:** providing both graphical and natural language interaction allows users to choose the interaction modality that best fits their experience, context, and task complexity. This enhances flexibility, fosters trust, and accommodates users with varying technical backgrounds.

- **Guided discoverability and onboarding:** novel systems, especially those leveraging natural language interfaces, should offer clear, easily accessible guidance at the beginning of the interaction. Examples, prompts, or FAQs can mitigate discoverability issues and support users in understanding how to interact with the system effectively.

- **Management of non-determinism:** in systems relying on AI models prone to non-deterministic behavior, interfaces should provide users with feedback, task summaries, and opportunities to verify or correct outputs. This preserves user control and confidence, even in complex or unpredictable scenarios.

- **Error recovery and mutual understanding:** both users and the system should be able to clarify and correct misunderstandings. Designing mechanisms for iterative clarification, feedback, and explanation enhances collaboration and reduces the risk of errors.

- **Learnability and inclusivity:** training and adaptive guidance are crucial, especially for users with limited experience with technology. Systems should be designed to be intuitive for diverse user groups, accommodating varying levels of digital literacy and linguistic expression.

- **Matching between CT skills and system cognitive demands:** systems should incorporate adaptive mechanisms, such as progressive disclosure of features, contextual examples, or guided templates, to accommodate users with diverse CT profiles. By matching the system's representation and interaction style to the user's reasoning patterns, developers can enhance usability, foster autonomy, and reduce the risk of cognitive overload.

4.9.2 Limitations of the Work

The research presented in this chapter represents only an initial step of the design and development of an AI-based EUD environment for programming collaborative robots to be used as pharmacists' companions. The following limitations affect the work.

First of all, as described in section 4.5.2, the *gpt-3.5-turbo* model was selected for ChatGPT API as it was the most powerful one available for free use at the time of development (August 2023). At present, the *gpt-5* model is available with a paid subscription. Testing our approach with this new model, or other more powerful models, would be interesting to overcome the problematic situations that (rarely) happen during the interaction with PRAISE related to imprecise output generation, prompt interpretation failures, and non-deterministic behaviors.

Another limitation concerns the user study organization. Due to the specific and stringent selection criteria of domain experts, recruiting volunteers who were willing to reach our robotics laboratory to use PRAISE with a real connected robot posed a challenge. Therefore, to maximize user participation, the study was carried out remotely, thus limiting observation possibilities and time for the test session. The users were, however, able to complete the assigned tasks without significant problems, and the tool for videoconference allowed collecting sufficient data to distill significant findings. Moreover, the average age of participants was relatively low, and this could have influenced the outcome.

Then, tasks carried out during the tests were performed only on the PRAISE application without any physical interaction with the robot. This disconnection from the real context might have altered the participants' perception of the usefulness and effectiveness of the proposed system. Additionally, we studied the automation of only three steps of the galenic preparation process and did not address the overall process execution in human-robot collaboration. To mitigate these limitations, the final interviews were designed to gather qualitative feedback from participants, providing insights into the potential application of collaborative robots and the associated programming environment beyond the tested scenario.

Finally, another limitation emerging from the EUDability evaluation concerns the system's dependency on users' CT skills. While PRAISE effectively supports pharmacists with moderate abstraction and decomposition capabilities, users with lower computational thinking skills may still encounter difficulties in defining complete task sequences or managing more complex compositions. The current design offers limited adaptive feedback in these situations. It lacks mechanisms for dynamically simplifying or restructuring the interaction flow according to the user's evolving proficiency, which would otherwise help ensure consistent accessibility and learning progression over time.

Chapter 5

A Domain-Independent Environment for Robot Task Definition: ADMIRABLE

Defining robotic tasks in a domain-independent way represents a crucial step toward making collaborative robots usable across multiple contexts without requiring reprogramming by technical experts. Among the most common and representative activities in collaborative robotics are *pick-and-place* operations, tasks involving the identification, grasping, and positioning of objects within a workspace. These tasks, though conceptually simple, entail the precise coordination of perception and control, and therefore provide an ideal testbed for exploring intuitive and accessible methods of robot programming.

This chapter presents the design and evaluation of ADMIRABLE (A Domain-independent Multimodal Interface for Robot Adaptable Behavior Leveraging End-user development), a domain-independent environment supporting the definition of robotic tasks through EUD methodologies. Building on the insights gained from the design and development of the PRAISE system, a new domain-independent EUD environment was designed. ADMIRABLE allows non-technical users to define items, actions (i.e., processing operations between a pick and a place), and locations, and to compose pick-and-place tasks by combining natural language interaction, graphical block-based programming, and virtual simulation through a digital twin of the robot. The integration of these modalities aims to lower the technical barrier to robot programming while ensuring task accuracy, transparency, and safety before execution.

In line with RQ2 (*"What are the capabilities and limitations of natural language interfaces in supporting users to accurately specify, validate, and refine structured robotic tasks, and how can multimodal interaction enhance this process during task definition?"*), this work investigates how multimodal interaction can enhance user understanding and control during task specification. In particular, this chapter presents the two prototypes that were developed to explore and assess the proposed solutions.

5.1 Aims and Assumptions

The developed approach merges three distinct interaction paradigms: i) natural language interaction via a chat-based interface, ii) visual interaction using a block-based graphic interface, and iii) desktop virtual reality interaction through a digital twin of the robot.

The primary objective of this multimodal interaction style is to facilitate users in defining simple robot tasks easily and quickly through the chat interface. Furthermore, users can use the block-based interface to verify the developed task and refine it by possibly increasing its complexity through direct manipulation of blocks. Through the final interaction modality, it is possible to run the defined task on the digital twin to anticipate any problems that may occur during execution on the real robot.

To progressively explore the feasibility of ADMIRABLE, two prototypes were developed. The first prototype[1], presented in *"Integrating ChatGPT with Blockly for End-User Development of Robot Tasks"* and published in the proceedings of the *19th International Conference on Human-Robot Interaction (HRI) 2024 (Boulder, Colorado, USA)* [52], focused on defining the basic multimodal interaction between a chat-based natural language interface and a visual block-based programming interface. The second prototype[2], whose related paper is currently under review in *ACM Transactions on Human-Robot Interaction*, extended this foundation by integrating a digital twin component, enabling users to simulate robot tasks before real-world execution and providing a stronger coupling with the robot's middleware.

For the project, the COBOTTA robot developed by DENSO WAVE Ltd. was chosen, one of the robots shown in Figure 2.1. The robot programming approach described in this chapter is expected to significantly improve the intuitiveness of COBOTTA amongst non-technical users. This approach strives to simplify the programming process for a wide range of tasks that can be executed by COBOTTA or any connected collaborative robot. It is worth noting that the expressive capabilities of the proposed prototype are limited to a subset of the basic functions of COBOTTA. In particular, the version of COBOTTA used for experimentation does not support remote control and operations that involve equipment beyond that provided in the basic edition, with a camera and an electric gripper.

5.2 The First Prototype

In the following, I will describe the two main components that constitute ADMIRABLE from the perspective of the end user: the chat interface and the graphic interface.

5.2.1 The Chat Interface

The objective of the chat interface is to facilitate the definition of a new pick-and-place task. The user can interact with the application using natural language via text-based or voice-based communication. The chat interface leverages an LLM to extract and parse the user's intents from their natural language requests. The model is provided with a specific and detailed initial prompt, which guides the model in fulfilling its designated role. The prompt describes the structure of the tasks the user can define, as well as the role of the model, which assists the user in defining all the required details. Additionally, the initial prompt defines the custom JSON structure that is requested as output. It is of significant importance to have a parsable data structure as output, as this allows for the construction of the logic upon it. It should be noted that the pick-and-place task, which can be defined by the user, may also include a processing action in between the pick step and the place step. Furthermore, tasks may contain different pick-processing-place structures, use loops

[1] BlocklyGPT - `https://github.com/luigigargioni/blocklyGPT`
[2] DTblocklyGPT - `https://github.com/luigigargioni/DTblocklyGPT`

and conditional statements, and include different levels of nesting. However, as reported in [88], the limitation regarding task complexity is inherent to the natural language itself. Indeed, even in a human-to-human conversation, it is challenging to formulate a request or a statement that includes multiple levels of nested logic. Therefore, it was decided to focus only on two nested levels of requests to build a task through the chat interface (e.g., a condition inside another condition). This assumption is fundamental to restricting the intent analysis and the focus of the LLM. Once the user has defined their task, the chat will provide a summary of the information provided for initial verification. The user is then allowed to modify the task, if they wish, and to explain the changes they have made. Alternatively, the user may choose to confirm the task and view it in the graphic interface, which offers a more intuitive representation of the robot program. Figure 5.1 shows the LLM-based chat interface.

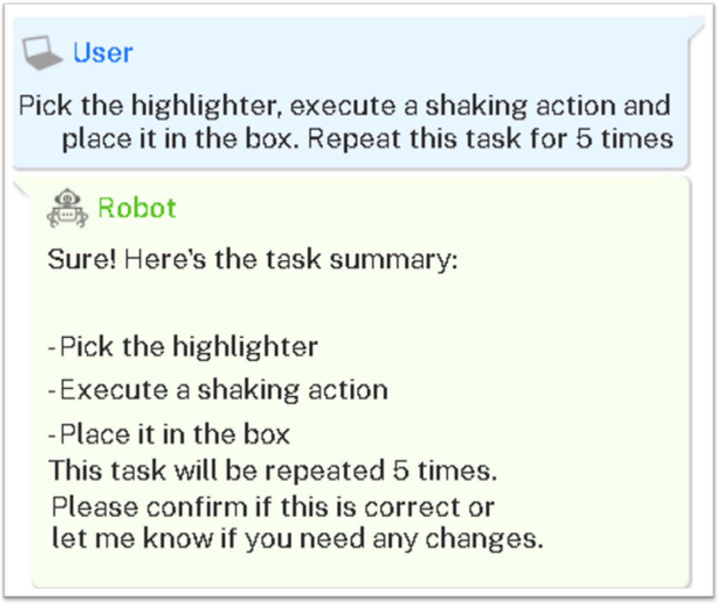


Figure 5.1: The LLM-based chat interface

5.2.2 The Graphic Interface

The drag-and-drop graphic interface is a supporting tool for easily and efficiently reviewing the task defined through the chat. Any discrepancies or misunderstandings that occur during natural language interaction can be addressed by editing the task in the graphic interface, which involves rearranging, inserting, and deleting blocks. The graphic interface also allows the creation of new tasks from scratch with an infinite potential for nesting. It leverages the Google Blockly library[3], capitalizing on its adaptability, specifically the ability to create custom blocks and define the conditions for their interconnection, thereby tailoring them more closely to the application domain. The creation of these custom blocks and conditions is undertaken by the developers of the application, with the objective of making them available to the end user. In this case, six custom block categories were designed. The *Logic* category contains the loop and conditional blocks used to create the task logic, such as the "When-Do" or "Repeat" blocks. The *Events*

[3]Google Blockly - https://developers.google.com/blockly

category includes the blocks used as conditions in the logic blocks, for example, the "Sensor signal" and the "Different object detected" blocks. The *Steps* category encompasses the primary blocks used to construct a task, including "Pick", "Processing", and "Place". The *Objects*, *Actions*, and *Locations* categories contain the items necessary for the *Steps* blocks. These are, respectively, objects for the "Pick" step, actions for the "Processing" step, and locations for the "Place" step. The items included in the categories of objects, actions, and locations can be defined directly by the user through dedicated features of the EUD environment. Figure 5.2 shows the Blockly-based graphic interface.

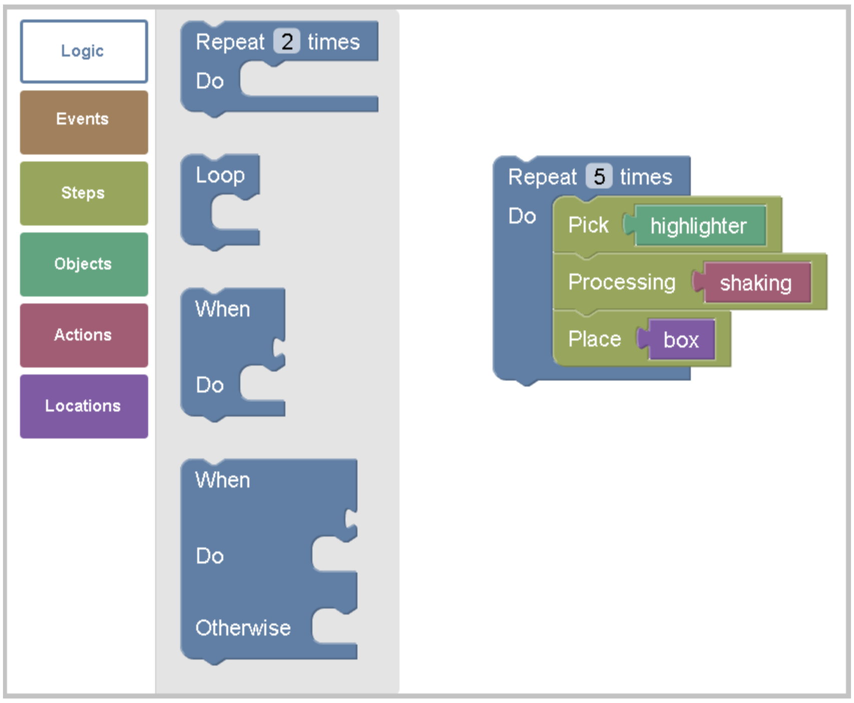


Figure 5.2: The Blockly-based graphic interface: in this Figure, the "Logic" category is expanded, and it shows the conditional and loop blocks

To create a new object, a tangible entity that a robot can manipulate, the user should capture a picture of it using the robot camera and set all relevant data required for proper manipulation. For identification purposes, users must provide an object name (e.g., "highlighter") and, optionally, a set of synonyms to be used by the natural language processing function of the chat. Technical data must also be provided, pertaining to the required force for object grasping and the approach to the object, i.e., the Z-axis distance the robotic arm needs to attain in order to pick up the object. After having acquired the photo of the object, the image is processed through segmentation, binarization, filtering, and contour searching. At the time of execution, the robot can efficiently search for items in the designated area, retrieve their position and alignment, and successfully grab them. Additionally, it is possible to save a fixed position for the object, allowing the robot to use that predefined location during execution instead of performing a search each time. This feature is particularly useful for static workspaces or repetitive tasks where the object's position remains unchanged.

A location refers to a point in space that serves as the target for a place operation (e.g., "box"). Therefore, its definition remains straightforward: upon providing the identification information for the location, the robot arm must be manoeuvred to the intended point before acquiring the specific position.

An action enables the robot to move along a specified path between a pick and a place step. For instance, the user may define a “shaking” action to be performed on the object before its placement. An action consists of a sequence of points (i.e., a cloud of points) that the robot arm must reach to follow a defined path. Recording a cloud of points is achieved by guiding the robot arm along the desired path and repeatedly pressing a specific button in the application to capture significant points of the trajectory. As for objects and locations, identification data related to the action must also be provided.

Regarding the different block connections in the drag-and-drop interface, a set of rules has been established with the objective of regulating the manner in which blocks may or may not be connected, thereby assisting users in constructing syntactically correct tasks. The structure and meaning of the blocks follow the typical programming constructs; however, some modifications have been made to align with the user’s language. For instance, conditional blocks are labeled with “When-Do”, rather than with the conventional “If-Then”. Similarly, loop blocks have a “Repeat” label, rather than the traditional “For” name. The possibility of defining elements of one’s own domain and using non-technical language is meant to facilitate user comprehension of the applications and the development process.

In summary, to create a task for the robot, the first step is to define the items, namely the object to pick *(e.g., highlighter)*, the location where to place the object *(e.g., box)*, and the optional action *(e.g., shaking)* to be executed between the pick and place steps. These items will then be saved in the corresponding libraries for reuse in future tasks. The corresponding steps (i.e., pick, optional action, and place) can then be organized into a sequence and incorporated within a repeat block (as illustrated in Figure 5.2) or within a conditional block.

The graphic interface enables the unwitting creation of a JSON data structure that represents a robot task. Such a data structure is then employed by the ROS-based architecture for the purposes of simulating the task on the digital twin and/or executing it on the real robot.

5.2.3 The System Architecture

ADMIRABLE is a web application implemented in accordance with the client-server paradigm. The core of the application comprises a front-end developed in Typescript with the React library and a back-end developed in Python with the Django framework. The data management system is based on a SQLite database.

The first prototype of this EUD environment served as a proof of concept for validating the multimodal workflow and the automatic generation of robot programs from user-defined goals. However, it did not yet support task simulation or safety verification through the digital twin, enabling only the task execution on the real robot. The prototype architecture is shown in Figure 5.3. It encompasses four layers:

1. The *User Interface Layer*: through this layer, the user can interact with either the chat or the graphic interface to define pick-and-place tasks using natural language interaction or block-based interaction.

2. The *Task Definition Layer*: this layer is the application logic of the EUD environment, where ChatGPT or Blockly are exploited to process users’ requests coming from the User Interface layer.

3. The *Data Layer*: in this layer, robot programs are stored in the database using a JSON format compatible with Blockly; parsing functions transform the output of ChatGPT into JSON data that the Blockly engine can recognize.

4. The *Execution Layer*: here, the task execution by the robot is managed by means of a program template.

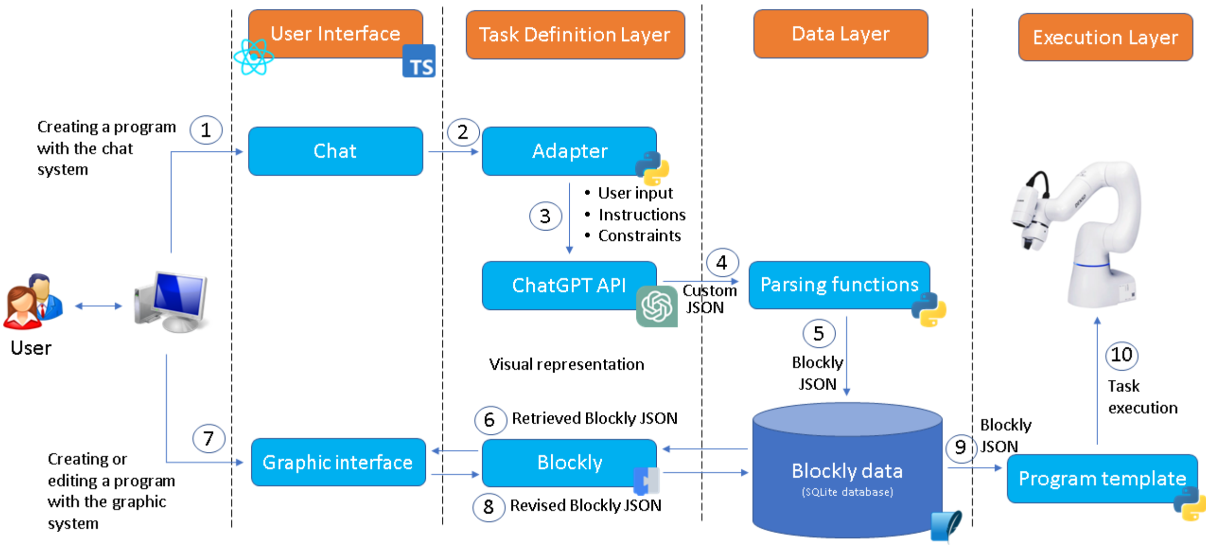


Figure 5.3: The first prototype architecture

Going more into detail in the user workflow, let us assume that the definition of a new task for the robot begins with the interaction with the Chat interface (*Step 1* in Figure 5.3). User messages, conveyed via the chat, are directed to an Adapter, which exploits the ChatGPT API to interpret the user's request *(Step 2)*. This module provides ChatGPT not only with the user's request, but also with specific constraints and instructions, to ensure precise and non-deterministic interpretation of the robot task *(Step 3)*. After the task definition is complete and the interaction with the Chat ends, a custom JSON representation of the robot program is generated *(Step 4)*. Following this, parsing functions are employed to convert the custom JSON format used by the Chat into a JSON format that can be interpreted by Blockly *(Step 5)*, enabling the program to be visualised in a graphical format *(Step 6)*. At this stage, the user can check the correctness of the program in the Graphic interface (*Step 7*), and can either confirm or modify it (*Step 8*). Finally, the Blockly program representation is used by a program template written in Python *(Step 9)*; this program will execute the elementary robot actions needed to accomplish the task *(Step 10)*, like moving the arm to a specific position or searching for an object in the working area.

Advanced users of the application can directly define a new robot task through the Graphic interface starting from *Step 7*. While this strategy guarantees faster execution, it necessitates more cognitive effort than using the Chat interface and then simply checking and correcting what AI has generated.

To tailor the behavior of ChatGPT, it must be provided with clear instructions about its goals and context. As mentioned above, the *Adapter* is responsible for handling the calls to the ChatGPT APIs and instructing the model about its goals. All instructions are given as prompts at the beginning of the conversation and are transparent to the user. This specification allows the model to recognize the details required to define a complete robot task. Furthermore, the defined *items* are also included in the context, allowing the model to accurately interpret what the user is referring to when formulating a request. In addition to these instructions, the request from the *Adapter* includes the JSON data format expected as the output of the model. The desired output is defined by passing a function to ChatGPT *Chat completions API*, which expects as parameter a representation of the custom JSON format. In this way, the API will respond with a JSON object compliant with the expected format. As to the *temperature* parameter of the model, a value of 0.2 was chosen to obtain a relatively deterministic reasoning and prevent creative interpretation of the user's intentions. However, a small degree of creativity was retained to allow the chat to better adapt to user's requests and exhibit a human-like behavior.

After the dialogue is concluded, ChatGPT produces a task description in a custom JSON format. This format was introduced because ChatGPT was unable to always provide an output compatible with Blockly. The custom JSON data is passed to Python functions, which parse the data and transform them into a Blockly-compatible JSON format.

After the generation of the Blockly JSON data, the parsing functions query the database to retrieve the identifiers of objects, locations, and actions involved in the task defined through the chat, and integrate them into the data structure as meta-data. Item identifiers are necessary for the subsequent task execution to obtain other technical data (e.g., object photo, location position, etc.). Item searching in the database is also performed using synonyms specified during the item definition. Once the retrieval of the item details is finished, a data format that includes all the required information for execution is available and compatible with Blockly. Subsequently, the task is displayed in the graphic interface.

The task can now be executed on the COBOTTA robot. To execute the task, a Python program template is activated, which analyzes the Blockly JSON code and identifies the essential elements required to generate an executable sequence of robot instructions. Beyond including calls to proprietary DENSO APIs, these ones also include the objects to pick up, the actions to execute, the locations where the objects should be placed, and the conditions and cycles to apply. During execution, the robot picks up objects either by locating them with the camera or by using a previously saved fixed position; the objects are then manipulated by the robot arm and finally placed in the desired location.

5.3 The Second Prototype

Building upon the first prototype, a second, extended version of ADMIRABLE was implemented. This version introduced a digital twin module tightly integrated with the robot middleware, allowing users to preview and validate robot behavior before deployment. The visualization through desktop virtual reality enables users to verify the accuracy of their definitions and assess the consistency of the objects, actions, locations, conditions, and loops.

Figure 5.4 shows the digital twin environment. It has been developed using Gazebo and comprises both the robot model and the surrounding environment. Gazebo[4] is a simulation environment used to create and test robotic systems in a virtual 3D world.

[4]Gazebo - https://gazebosim.org/home

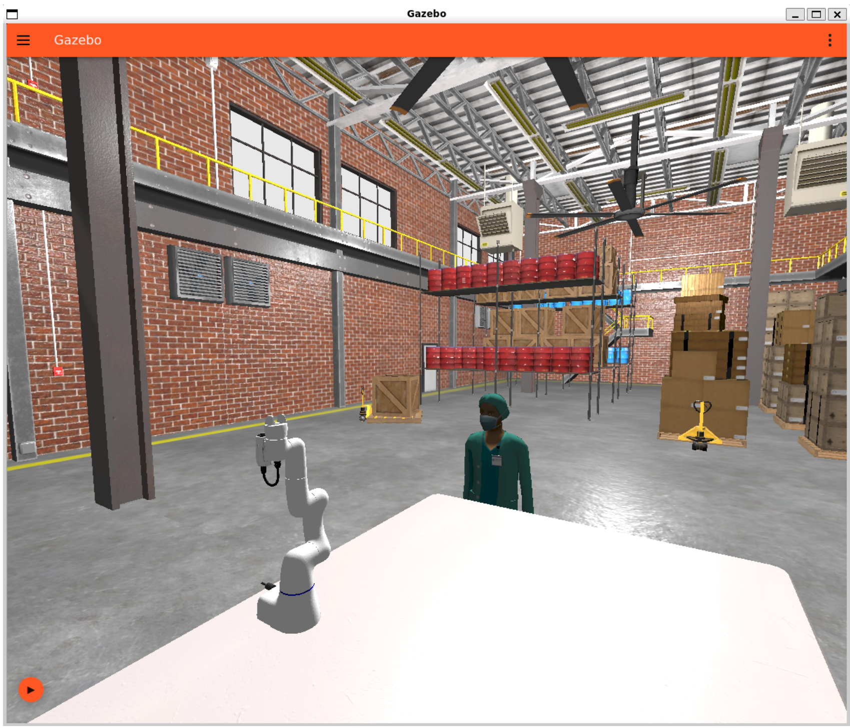


Figure 5.4: The digital twin environment

The integration of the digital twin required the development of additional software components and the adaptation of the existing architecture to ensure seamless communication between the virtual and real environments through the ROS infrastructure. Additionally, to integrate a digital twin into the development workflow, it was necessary to implement a ROS-based architecture that manages different nodes[5], topics[6], and messages[7], and that includes the Gazebo node[8].

This section describes the main stages of the COBOTTA modeling process, the integration of the digital twin into the ROS-based architecture, and its connection with the EUD environment.

5.3.1 The COBOTTA Model

A comprehensive description is necessary regarding the construction and integration of the COBOTTA digital twin in ADMIRABLE. At the time of the application development, a complete model of the robot in question was not available for integration into the Gazebo environment. Therefore, the robot model had to be constructed using the components provided by the manufacturer DENSO WAVE Ltd. The initial stage of the process entailed downloading, from the manufacturer's website, the exploded version of all the components of the robot (Figure 5.5) and technical datasheets, which included information on angles of rotation, inertia, and other physical characteristics. The components have been assembled in accordance with the specifications described in the technical datasheets, employing a sequence of steps and using different 3D modeling software, in-

[5]ROS Nodes - https://wiki.ros.org/Nodes
[6]ROS Topics - https://wiki.ros.org/Topics
[7]ROS Messages - https://wiki.ros.org/Messages
[8]Gazebo node: in ROS, a Gazebo node enables communication between the simulator and other ROS nodes, allowing the exchange of sensor data, control commands, and state information.

cluding RViz2[9] (Figure 5.6), Blender[10] (Figure 5.7), and Phobos add-on[11]. A 3D model has been constructed, incorporating all the geometric details, collision data, and inertial effects. Subsequently, the model has been exported in SDF format[12], an XML format that describes objects and environments for simulation in the robotic field, which is compatible with the Gazebo environment. Upon completion of the COBOTTA model, the simulated environment surrounding the robot has been developed to reproduce the real world. Particularly, the simulated environment comprises the supporting structure on which the robot is placed, such as a table, as well as all other objects that constitute the scene in the simulation interface. The final result of these processes is the generation of two SDF files: one for the robot and another that represents the simulated environment.

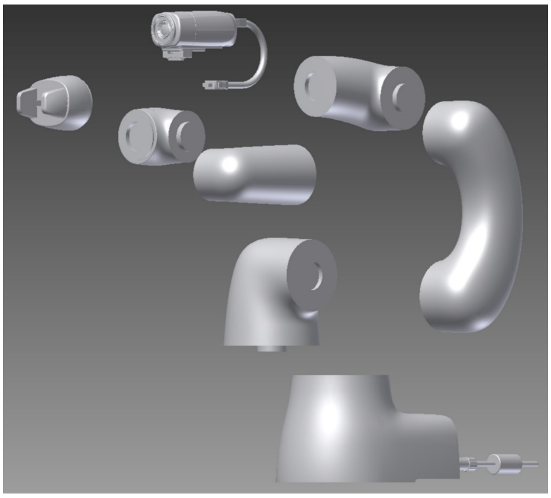


Figure 5.5: The exploded COBOTTA model

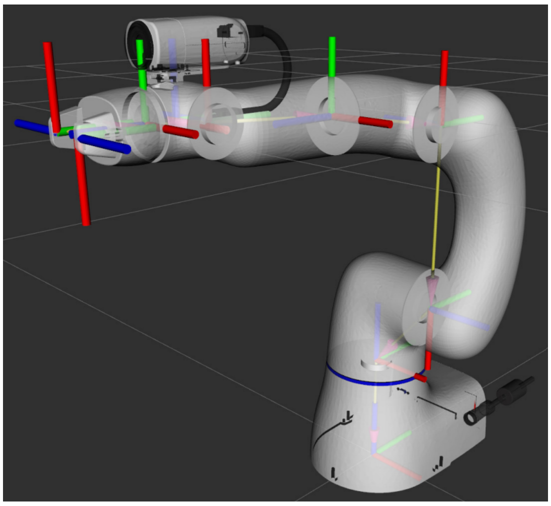


Figure 5.6: The COBOTTA model in RViz2

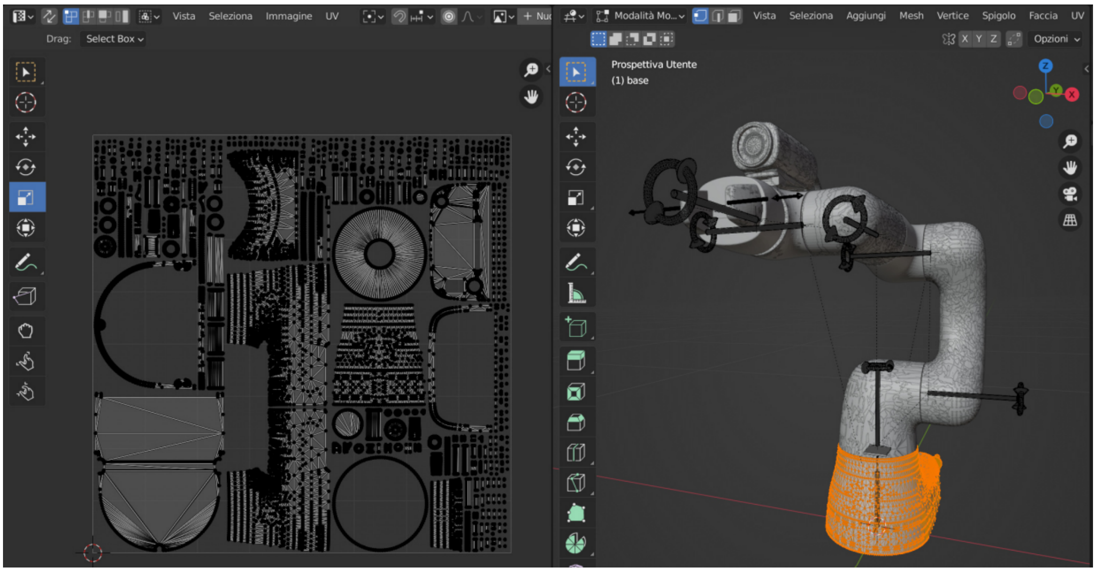


Figure 5.7: The COBOTTA model in Blender

[9]Rviz2 - https://github.com/ros2/rviz
[10]Blender - https://www.blender.org
[11]Phobos - https://github.com/dfki-ric/phobos
[12]Simulation Description Format (SDF) - http://sdformat.org

5.3.2 Digital Twin Integration into a ROS-based Architecture

As previously stated, the main application is a web-based application that employs a client-server paradigm, and it includes the chat and the graphic interface. To integrate the functionalities offered by ROS and Gazebo, it was necessary to create an architecture that manages these tools, as well as a middleware to connect the main application and the digital twin.

The robot middleware plays a pivotal role in receiving instructions from the main application and/or data from the real robot, as well as instantiating topics and creating messages for the digital twin. It is also important to note that this data flow also operates from the digital twin, whereby data from the digital twin is required for action on the real robot. The robot middleware is a Flask[13] application, a Python micro-framework that facilitates the creation of lightweight servers. It exposes a range of APIs that facilitate communication with the main application. Furthermore, the *rclpy*[14] Python library is employed for the management of nodes, topics, and messages that are integral to the ROS architectural framework.

Figure 5.8 illustrates the details of the digital twin based on the ROS architecture. The *Flask node* serves as the interface with the main application. Other significant nodes include the *Cobotta node*, which oversees the transmission and reception of commands to and from the physical robot; the *Ros-Gazebo Bridge*, which translates Gazebo topics into ROS topics; and the *Gazebo node*, which handles messages from the Gazebo virtual environment thanks to the *Ros-Gazebo bridge*.

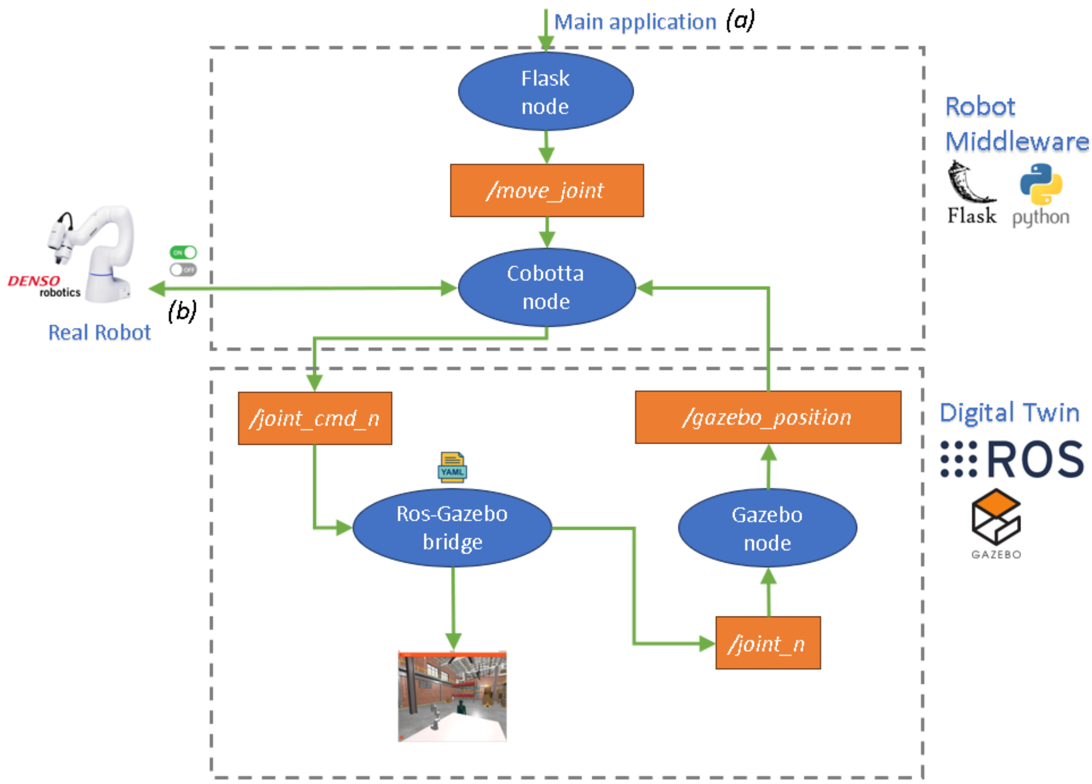


Figure 5.8: The digital twin architecture in detail and its interaction with the robot middleware

[13]Flask - https://flask.palletsprojects.com/en/stable
[14]ROS Client Library for the Python language (rclpy) - https://github.com/ros2/rclpy

The *Ros-Gazebo Bridge* is a YAML[15] file that maps each Gazebo node name to a ROS node name, thereby establishing a one-to-one mapping. This was a necessary step, as the conventions governing node nomenclature differ between the two systems. Following the ROS framework workflow, various topics and messages are exchanged between the different nodes. The ROS topics to which nodes transmit and receive messages are described in Table 5.1.

Topic	**Description**
/move_joint	Used to set a specific pose, which consists of a specific position for every joint.
/joint_cmd_n	Used to move every single joint in the Gazebo graphic interface.
/joint_n	Used to acquire the new position of the digital twin from Gazebo.
/gazebo_position	Used to merge all the joints' positions and return a specific robot pose.

Table 5.1: Description of topics used in the ROS-based architecture

Two different workflows are possible with this architecture. As one may observe in Figure 5.8, it is possible to send commands from the main application *(a)* to the Gazebo interface to execute a specific task. If the movement synchronization is enabled, the task will also be executed on the real robot *(b)*. Therefore, it is crucial to ensure that synchronization is enabled only when the selected task or movements have already been verified. A second method of interaction involves moving the real robot by teaching actions or tools; in this case, the positions and movements will be reflected in the digital twin.

5.3.3 Digital Twin Integration into the EUD Environment

Figure 5.9 shows how the described components are integrated in ADMIRABLE. The interaction starts with the chat interface (i.e., the green box in the figure), where the user can express their request in natural language. The objective of the LLM-based chat interface is to generate a structured intent based on the user's request. Accordingly, once the dialogue ends, the LLM generates a task description in a custom JSON representation.

Python functions convert the custom JSON into a Blockly-compatible format, integrating identifiers from the database that are essential for robot task execution; item synonyms improve the accuracy of database searches. The Blockly graphic interface (i.e., the blue box in the figure) allows users to review and modify tasks.

Finally, a Python program template analyzes the Blockly JSON code to identify the key elements required for generating an executable sequence of robot instructions to execute on the real robot or in the Gazebo virtual environment (i.e., the orange box in the figure).

[15]Yet Another Markup Language (YAML) - https://yaml.org

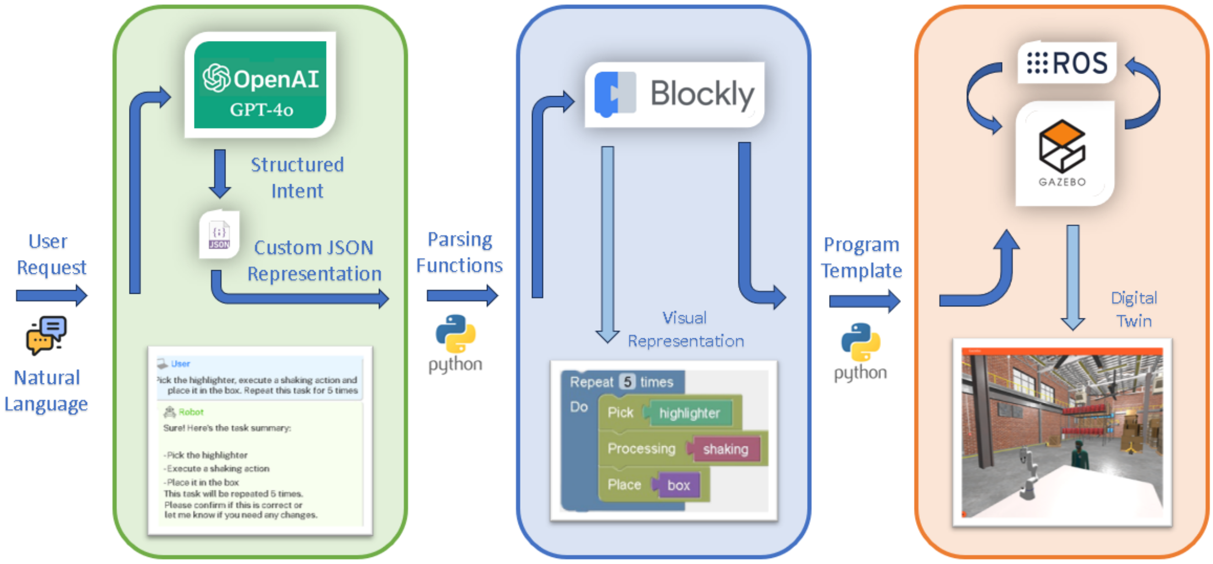


Figure 5.9: The components in the EUD environment

5.4 An Example of the EUD Environment at Work

In this section, I use a realistic interaction scenario to provide a detailed account of the user journey necessary to define and verify a robot task through the EUD environment:

A laboratory technician is required to undertake a series of chemical reactions within test tubes, subsequently transferring the test tubes to different containers, dependent on the nature of the reaction. To evaluate the type of reaction, the robot must position the test tube in a specific location in front of a camera capable of identifying the type of reaction that has occurred. To fulfill this task, the technician must specify the object to be used (test tube), the specific position of the robot in front of the camera, and the locations where the test tubes should be placed based on the reaction type (a divider in case of a successful reaction, a collector otherwise).

We assume that, before starting the task definition, the user has defined the *test tube* object, the *reaction-checking* action, and the two locations, *collector* and *divider*, as final positions for the test tubes. Once the required items for the task have been defined, the laboratory technician can use the chat interface to define the task. In the chat, the laboratory technician could ask: *"Take 20 test tubes, place them in front of the camera for the reaction-checking phase. For each test tube, if the camera sensor detects that the reaction occurred correctly, place the test tube in the divider; if not, put the test tube in the collector as it is to be discarded"*.

The task definition may also be provided in a step-by-step format. In that case, the chat interface will assist the user in defining the different steps and details. In a step-by-step interaction, the LLM-based chat interface must analyze a reduced amount of information each time. This type of interaction helps the model to better comprehend the details specified by the user to define the task. In this scenario, we assume that the entire task definition is provided in a single message, which represents the more challenging case. Indeed, the LLM must elaborate the full user message and parse the entire structure of the sentence, or even multiple sentences connected in a unique message, searching for the specific details that comprise the task.

Once all the details of the task are defined and the definition process is finished, a summary of the robot task is reported as the final message of the chat. The user could still modify it, for example, by changing the object to pick, through the request *"Use the flask object for my task and not the test tube"* and keeping the rest of the task structure. In this case, we assume that the user identifies no issues and confirms their intention to proceed with using the graphic interface for additional verification.

The user can now view the defined task in the graphic interface (Figure 5.10). As one may observe, the block-based representation of the robot task does not contain any technical terminology unfamiliar to the user. Even here, the user could easily change, for example, one of the two final containers by simply substituting the specific block and keeping the rest of the structure (Figure 5.11).

As a final step, the user wants to simulate the defined task. Before starting the simulation, the user can decide whether the specified condition (in this scenario, the camera sensor signal) will be regarded as true or false. This condition can be specified in the main application by using a checkbox in the proper form to fill in before starting the simulation. This is due to the fact that, within the context of a simulated environment, conditions based on real objects must be simulated. According to the user's decision regarding the camera sensor condition, the *collector* will appear in the simulation for the purpose of collecting discarded test tubes or, alternatively, the *divider* will appear for validated test tubes.

Within the interaction scenario, the first step of the task is the picking action, which involves the robot arm approaching the object (i.e., the test tube) to identify and subsequently grasp it. Subsequently, the processing action consists of moving the test tube to a designated position where a camera analyzes the reaction that has occurred within it. In the final step, the place action is executed. In this case, illustrated in Figure 5.12, the user decides to simulate the condition in which the camera sensor detects that the reaction occurred correctly, prompting the robot to place the test tube in the divider.

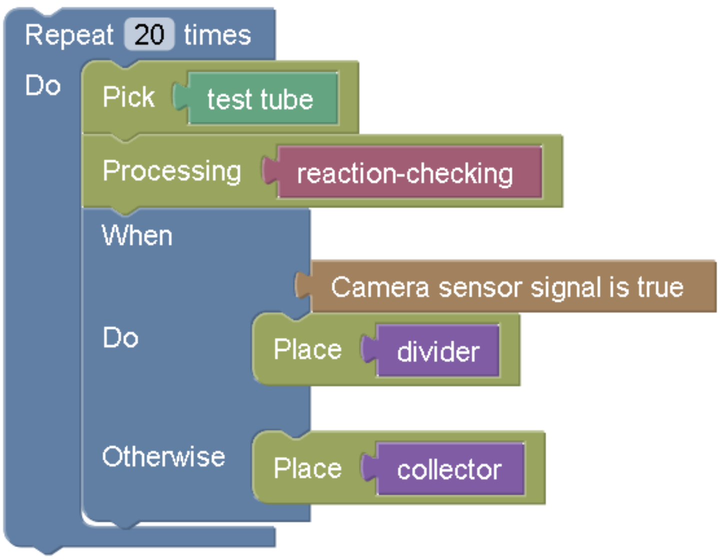


Figure 5.10: The task defined in the interaction scenario shown in the graphic interface

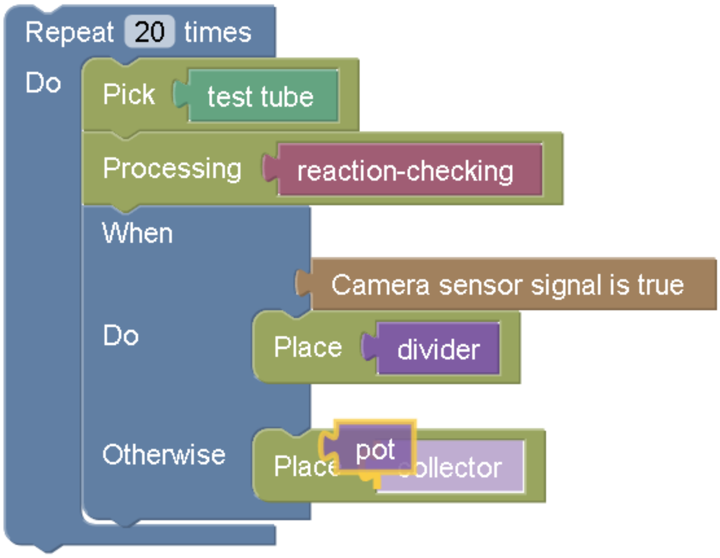


Figure 5.11: Editing the defined task through the graphic interface

Figure 5.12: Placing the test tube in the divider

5.5 User Study

A user study was conducted to gather feedback from users and evaluate the feasibility of the proposed robot programming approach.

5.5.1 Methodology

To evaluate the user experience during the interaction with the prototype and identify the main challenges and problems encountered by participants, we adopted a mixed-method approach combining qualitative and quantitative data. This approach allowed us to obtain a comprehensive understanding of both the subjective perceptions and the objective performance of users.

To this purpose, a set of complementary analysis tools was employed:

- direct observation of participants' behavior during task execution and collection of participants' comments; notes on the sessions were jointly collected by two researchers (including me), and subsequently analyzed by both of us to identify relevant themes according to an inductive and latent thematic analysis [25]. Each theme led to identifying recommendations for system improvement;

- quantitative data about task completion percentage and number of errors/retries were also collected to provide objective indicators of performance. These data allowed us to complement the subjective evaluations with measurable evidence of user efficiency and task success rate;

- the SUS (System Usability Scale) questionnaire [27] (see Appendix A.2) to assess the perceived usability of the system, to be filled in by participants immediately after the conclusion of the test session;

- the NASA-TLX (NASA Task Load Index) questionnaire [66] (see Appendix A.3) to assess the subjective workload experienced by participants during the interaction, to be filled in by participants immediately after the SUS questionnaire. The full weighted version of the NASA-TLX was used, including the pairwise comparisons to compute the weights of the different dimensions;

- a semi-structured individual interview with each participant, carried out at the end of the test session. The interview aimed to gather additional feedback on participants' overall experience with the application, focusing on the ease and naturalness of interaction, the usefulness of the block-based task representation, the effectiveness of the digital twin simulation, and participants' perceptions of safety and security when using a chat-based interface for robot programming. Participants were also invited to provide suggestions for improvements or additional features. As before, the collected answers were examined by two researchers (including me), who applied an inductive and latent thematic analysis to identify further and more general themes to be considered for the design of similar systems and future work.

Twelve participants (5 females and 7 males) have been involved in the user study. Participants work or study in various fields and have different backgrounds, specializations, and experiences. Table 5.2 reports the demographic data of participants, their sector, their knowledge of programming (on a scale from 1 - very low, to 5 - very high), and their knowledge of robotics (on a scale from 1 to 5).

None of the participants had an educational or professional background in computer science, robotics, or artificial intelligence. The reported levels of experience in programming (M=1.58, SD=0.90) and robotics (M=1.17, SD=0.39) confirm this. This was an intentional choice to ensure that users did not have prior biases or specific knowledge about the technologies used, thus providing more genuine and unbiased feedback.

Additionally, participants P5 and P9 reported having dyslexia. This condition might have influenced certain aspects of their interaction with the system or their responses during the study. Nevertheless, their inclusion is valuable, as it reflects the natural diversity of the general population and contributes to a more realistic and inclusive evaluation of the proposed approach.

ID	**Age**	**Gender**	**Sector**	**Programming experience *(1-5)***	**Robotics experience *(1-5)***
P1	25	Male	Cultural heritage	1	1
P2	26	Male	Economics	3	1
P3	26	Male	Astrophysics	3	1
P4	37	Male	Agriculture	1	1
P5	25	Male	Surveyor	2	1
P6	19	Male	High School Student	1	1
P7	35	Female	Elementary School Teacher	1	1
P8	24	Male	Manufacturing	1	2
P9	19	Female	High School Student	1	1
P10	24	Female	Human Nutrition	1	1
P11	26	Female	Biomedicine	3	2
P12	26	Female	Chemical Laboratory Quality Control	1	1

Table 5.2: Partecipants data

Participants in the user study were asked to execute five tasks of increasing complexity related to defining robot programs through the chat, the graphic interface, and the digital twin. The last task also involved executing the task on the real robot. Figure 5.13 reports the details of the assigned tasks.

Before starting the experiment, participants were invited to an introductory session of about 5 minutes. In this introduction, we described the goal of the study and provided a brief overview of the prototype. The think-aloud protocol was used during the tests, encouraging participants to verbalize their thoughts as they executed the tasks.

Meanwhile, we observed each participant and annotated their comments and significant behaviors, including difficulties they encountered while interacting with the prototype and errors they made during task execution. Upon completion of the test session, we asked the participant to fill in the SUS and NASA-TLX questionnaires and conducted an interview with them.

The user study was conducted in person in the Robotics Laboratory of the University of Brescia. Google Forms was used to collect user data and SUS results. An online tool[16] was used to collect data about the NASA-TLX. The prototype was run on the laptop of the research group. The setup included two monitors, one dedicated to the EUD environment and the other to the digital twin visualization window. Each test session lasted about 45 minutes, including the introductory session.

A pilot study with two participants was conducted prior to the user study to validate the test protocol, refine the description of the submitted tasks, and calibrate the time estimation required to complete them.

5.5.2 Ethical Considerations

The study was reviewed and approved by the Ethics Committee of the University of Brescia, Italy (approval code: 23/2025). All participants were informed by written informed

[16] NASA-TLX Calculator - https://nasa-tlx-calculator.vercel.app

1. **Scenario 1 – Define a basic robotic task:**
   Imagine that you want to use the robot to pick up an object, perform a short action, and then place it in a specific location. Specifically, you will need to have the robot pick up the blue flask, perform a shaking action, and then place the flask in a box.

   Use the chat to specify what the robot is to do. The application will guide you in defining the task, and you can then view it in the graphic interface to check its correctness.

2. **Scenario 2 – Add a condition:**
   In this scenario, the robot must pick up an object and decide where to place it based on a condition (e.g., a sensor result). In this case, the object, once picked up, will be placed in front of a camera for a scan to verify that its contents are correct.

   Specifically, if a signal arrives from the camera sensor, you should have the robot pick up the blue flask and put it in the collector; otherwise, if there is no signal, it should pick up the yellow flask and put it in the box.

   After defining the behavior with the chat and controlling it with the graphic interface, you can verify the operation using the digital twin to observe how the robot reacts to different possible scenarios. You can simulate one condition (signal from sensor present) or the other (signal from sensor NOT present) directly in the simulation launch window.

3. **Scenario 3 – Repeat actions:**
   Now the robot must perform a sequence of actions several times, such as picking up and moving several similar objects one after another.

   Specifically, it has to pick up the blue flask and put it in the box. Specify this task using the chat by having the robot repeat the pick-and-place twice; then check the task in the graphic interface and change the number of repetitions as desired.

4. **Scenario 4 – Add a condition in the graphic interface:**
   In this scenario, you will have to define a task directly through the graphic interface and then display it using the digital twin.

   Specifically, the robot will have to pick up the blue test tube and, if feedback from the operator is detected (e.g., a particular hand gesture), it will have to perform a shaking action; otherwise, it will have to perform a rotating action. Finally, the tube should be deposited in the box.

   To verify that the defined task is correct, simulate the task on the digital twin.

5. **Scenario 5 – Complete task with final verification on the real robot:**
   In this last scenario, you will have to define a complete task using everything seen so far.

   Specifically, the task that the robot will have to perform is to look for a blue flask: if it finds it, it has to take it and put it in the collector; if, on the other hand, the blue flask is not available, then it has to take the yellow flask and put it in the box.

   Once defined in the chat and verified through the graphic interface, add from the graphic interface that the entire sequence must be executed twice. Finally, you could see it in action first on the digital twin and then on the real robot.

Figure 5.13: Tasks performed during the user study

consent. Participation was voluntary and could be withdrawn at any time without giving reasons.

5.6 Results

The results obtained from direct observation, the quantitative data about task completion percentage and number of errors/retries, the SUS questionnaire, the NASA-TLX questionnaire, and final interviews are detailed below.

5.6.1 Findings from Direct Observation and Participants' Comments

The following themes and related recommendations for system improvement emerged from the thematic analysis of data gathered during task execution. Table 5.3 summarizes the themes that emerged from the thematic analysis with their corresponding codes.

Table 5.3: Themes and codes emerged from the thematic analysis of the direct observation and participants' comments

Theme	Codes
Cognitive load and abstraction	• Understanding of condition simulation • Unfamiliar control concepts (loops, conditionals) • Logical reasoning ability • Trust in system suggestions
Interface comprehension and navigation	• Icon interpretation issues • Low affordance of simulation screens • Non-persistent execution feedback • Terminology guiding navigation
Progressive learning and exploration	• Linguistic exploration and robustness testing • New description strategies through chat • Perceived improvement in task definition • Understanding of validation workflow • Use of analogies to interpret logic
Support needs, personal preferences, and accessibility	• Need for scaffolding • Use of documentation for task creation • Cognitive style preferences • Accessibility for dyslexic users • Summaries reducing memory load • Need for visual confirmation

Cognitive load and abstraction - Participants displayed heterogeneous levels of familiarity with computational thinking concepts, particularly conditional reasoning and control structures. For instance, one participant, faced with the checkbox in the modal window to launch the simulation, indicating a conditional option, initially commented *"This thing is indifferent, I suppose"* (P1), signalling a lack of understanding of its functional meaning. Conversely, the simulation of conditions was occasionally grasped intuitively, as reflected in P2's comment: *"I imagine this is for the sensor"*, showing an ability to link abstract representations to robot behaviors. Difficulties also emerged in understanding how many times a loop would execute, with two users asking: *"If I write 2, does it repeat twice or three times?"* (P3, P12). Logical abstraction abilities varied: P5, while performing a task with conditional blocks, stated *"I need 'when-do', and the logic should go inside it"*, demonstrating an emerging understanding of sequencing. Trust in the system also played a role in reducing cognitive load. In two cases, users relied on the chat's suggestions without fully validating them, as illustrated by P1 and P5 stating: *"I trust it, the system knows more than me"*.

Recommendations: To better support users with limited familiarity with computational thinking concepts, the system should incorporate concise contextual explanations and progressive disclosure of advanced constructs. Short, embedded clarifications for loops and conditionals, together with concrete examples, may help reduce cognitive load and foster more accurate mental models.

Interface comprehension and navigation - A second theme concerned users' interaction with the interface of ADMIRABLE and their ability to navigate between pages. Misinterpretations of icons or functions occurred. For example, P1 and P10 expressed concern regarding the meaning of the run icon, asking: *"Run is for the real robot, right?"*, revealing fears of unintentionally triggering real-world actions. Some issues arose with the simulation window, which did not close automatically and occasionally left participants unsure about how to proceed. P3 and P7 struggled with screens lacking clear affordances, asking: *"If I click delete, will it delete everything? Can I click outside?"*. A participant also noted that the execution confirmation pop-up disappeared too quickly, especially when using separate screens for the chat and simulator, leading to missing important information. At the conceptual level, terminology sometimes influenced navigation patterns. Terms that were perceived as intuitive, such as saying *"Let's start with pick"*, misled participants toward the *Action* section, even though *Pick* was located under the *Step* section (P5, P7, P8, P11). This suggests that the *Step* label may not be very evocative, especially since the *Action* section contains the operations associated with the processing step.

Recommendations: Enhancing interface affordances and the persistence of feedback is crucial for reducing navigation uncertainty. Clear textual labels for critical icons, explicit and consistent closing mechanisms in modal windows, and longer-lasting confirmation messages would help users maintain their orientation across screens. Adjusting terminology to reflect domain-relevant actions may further enhance intuitive navigation and reduce ambiguity. It should also be considered that some users may not be used to multi-monitor setups, which could introduce additional challenges in managing multiple screens and maintaining context.

Progressive learning and exploration - Across tasks, most users demonstrated a clear progression in competence. Also, four participants articulated a sense of improvement: *"I have gotten better at doing these tasks"* (P4). Two users experimented with alternative wording to test the system's linguistic robustness, writing "grab" instead of "pick" (P5). Another one explored whether tasks could be specified in a single request rather than sequentially: *"Can I write everything together? Let's try"* (P10). Four users internalized the importance of simulation as a validation step, as reflected in P8's remark: *"It seems correct, but before running it, let's see if it works in the simulator"*. Users occasionally leveraged analogies from familiar domains to understand logical constructs. For instance, P12 compared the conditional block to spreadsheet logic: *"It's like an Excel function: if this is less than that, then you do it, otherwise..."*. Such analogies highlight how participants appropriated existing mental models to interpret programming principles.

Recommendations: To encourage effective learning through exploration, the environment should provide safe and reversible interaction pathways. Features such as robust undo/redo capabilities, non-destructive sandboxing within the simulator, and optional reformulation suggestions can strengthen users' confidence in experimenting.

Support needs, personal preferences, and accessibility - Participants differed in the level of scaffolding they required. Some explicitly requested examples or brief guides (P2, P8, P10, P11), while others consulted the FAQ and referenced it when constructing robot tasks (P1). Cognitive style strongly influenced modality preference: P12 preferred the chat because it felt *"less schematic"*, whereas others, particularly users with dyslexia (P5, P9), preferred the graphic blocks and even expressed reluctance to use the chat: *"Do I really have to use the chat?"*. Memory-related cognitive load emerged as another factor. The chat's automatic summaries were appreciated, as expressed by P2: *"The summary is nice, because I don't remember things"*. The graphic interface visualization was sought as reassurance of correctness; for instance, during the task definition using the chat interface, P7 commented: *"Will it work? Well, now it will show me the graphic interface"*.

Recommendations: Given the heterogeneity of user preferences and accessibility needs, the system should ensure full functional equivalence between chat-based and graphic modalities, allowing users to adopt the interaction style most suitable to them. Optional scaffolding, such as brief examples or stepwise hints, should be available without constraining exploratory use. Furthermore, dyslexia-friendly visual settings, adaptable text options, and persistent visual summaries would improve inclusivity and reduce cognitive and memory demands. In addition, as dyslexic users tended to prefer block-based interactions, it is worth considering the needs of color-blind users by ensuring that color choices in the graphic interface are distinguishable and supported by redundant visual cues.

5.6.2 Findings from Quantitative Data

The quantitative performance measures reported in Table 5.4 provide an objective assessment of task completion and error management across the five tasks. Each task was constrained by a time limit of approximately six minutes, within which participants were expected to complete the assigned activity. This constraint also allowed us to assess not only whether participants completed the task, but also the percentage of progression they achieved within the allocated time when full completion was not reached.

The results show that overall task success was extremely high: with the sole exception of P4 in Task 1 (70% completion), all participants successfully completed all tasks (100% completion rate). This indicates that, despite the increasing complexity of the task sequence, users were generally able to accomplish the assigned activities without major difficulties.

The number of retries offers further insight into the interaction dynamics and the challenges encountered. Although completion rates remained consistently high, the distribution of retries reveals variability in how smoothly participants were able to carry out certain tasks. Task 4, in particular, elicited the highest number of retries overall, with several participants requiring multiple attempts (e.g., P6 with 4 retries, P10 with 5 retries). This aligns with the findings from qualitative data, which suggest that integrating conditions and multi-step reasoning poses greater cognitive demands. Conversely, Tasks

2 and 3 showed minimal retries, suggesting that the intermediate steps of the workflow were generally well understood and easier to execute.

Overall, the quantitative data indicate that the system enables users with absent or limited expertise in computer programming to successfully define robot programming tasks, even when these involve condition handling, multi-step workflows, and real-robot execution. At the same time, the variability in retries suggests that some tasks, particularly those involving conditional logic or more complex structures, may benefit from additional guidance or more explicit feedback mechanisms to reduce errors and improve first-attempt success rates.

Table 5.4: Quantitative data from the user study

ID	**Task 1**		**Task 2**		**Task 3**		**Task 4**		**Task 5**	
	Done	**Retry**	**Done**	**Retry**	**Done**	**Retry**	**Done**	**Retry**	**Done**	**Retry**
P1	100%	2	100%	3	100%	1	100%	2	100%	0
P2	100%	0	100%	0	100%	1	100%	0	100%	0
P3	100%	0	100%	0	100%	0	100%	1	100%	0
P4	70%	1	100%	0	100%	0	100%	1	100%	1
P5	100%	0	100%	0	100%	0	100%	0	100%	0
P6	100%	2	100%	0	100%	0	100%	4	100%	1
P7	100%	0	100%	0	100%	0	100%	2	100%	0
P8	100%	0	100%	0	100%	0	100%	3	100%	1
P9	100%	0	100%	1	100%	0	100%	1	100%	0
P10	100%	0	100%	0	100%	0	100%	5	100%	2
P11	100%	0	100%	4	100%	3	100%	1	100%	0
P12	100%	0	100%	0	100%	1	100%	2	100%	1

5.6.3 Findings from SUS Questionnaire

The SUS questionnaire was administered to obtain a standardized measure of the usability of ADMIRABLE. The average SUS score obtained in the study is 80.83, with a standard deviation of 12.12 and a median of 82.5. Individual scores for each participant are reported in Figure 5.14. According to the benchmarks reported by Bangor et al. [11], scores above 68 are considered above average, and the obtained value falls within the *Excellent* range.

Examining the score distribution, individual participant scores range from 62.5 to 97.5, indicating generally positive evaluations but with some variability. Most respondents (9 out of 12) reported values above 75, suggesting that the system was consistently perceived as usable, with only a few participants assigning ratings in the lower part of the acceptable range (62.5–67.5). Crucially, no responses fell into the *Not Acceptable* category (below 50). This variation may reflect differences in participants' backgrounds.

Using the letter-grade conversion proposed by Lewis and Sauro [92], the overall SUS score corresponds to a *Grade A*. This placement indicates that, on average, participants perceived the system as highly usable within the conventional SUS grading framework, although the observed variability suggests that certain aspects of the interaction may still pose challenges for some users. Furthermore, Lewis and Sauro [91] demonstrated that the SUS questionnaire encompasses a distinct *learnability* factor, captured by questions 4 and 10, which are reverse-scored (i.e., lower scores indicate better learnability). In our data, Q4 (*"I think that I would need the support of a technical person to be able to use* *the system."*) reached a mean of 2.75 (SD = 1.14), indicating a moderate perception of required support. Q10 (*"I needed to learn a lot of things before I could get going with the system."*) yielded a low mean of 1.5 (SD = 0.52), pointing to a more consistently positive perception of learnability. Overall, these results suggest that participants experienced the system as relatively easy to learn, with minimal need for prior knowledge or technical assistance. This aligns with the qualitative findings, which highlighted the value of a brief introduction and initial scaffolding, after which users generally felt able to navigate the system independently.

Overall, the SUS results indicate a generally positive usability assessment, with strong central values and moderate dispersion, providing quantitative support to the qualitative feedback collected during the study. Moreover, the pattern of responses shows the typical profile of systems perceived as usable, with positive items generally clustering toward higher values and negative items clustering toward the lower end of the scale. The variability observed in items related to initial learning effort suggests that onboarding and early guidance may represent areas where targeted improvements could further harmonize the experience across participants.

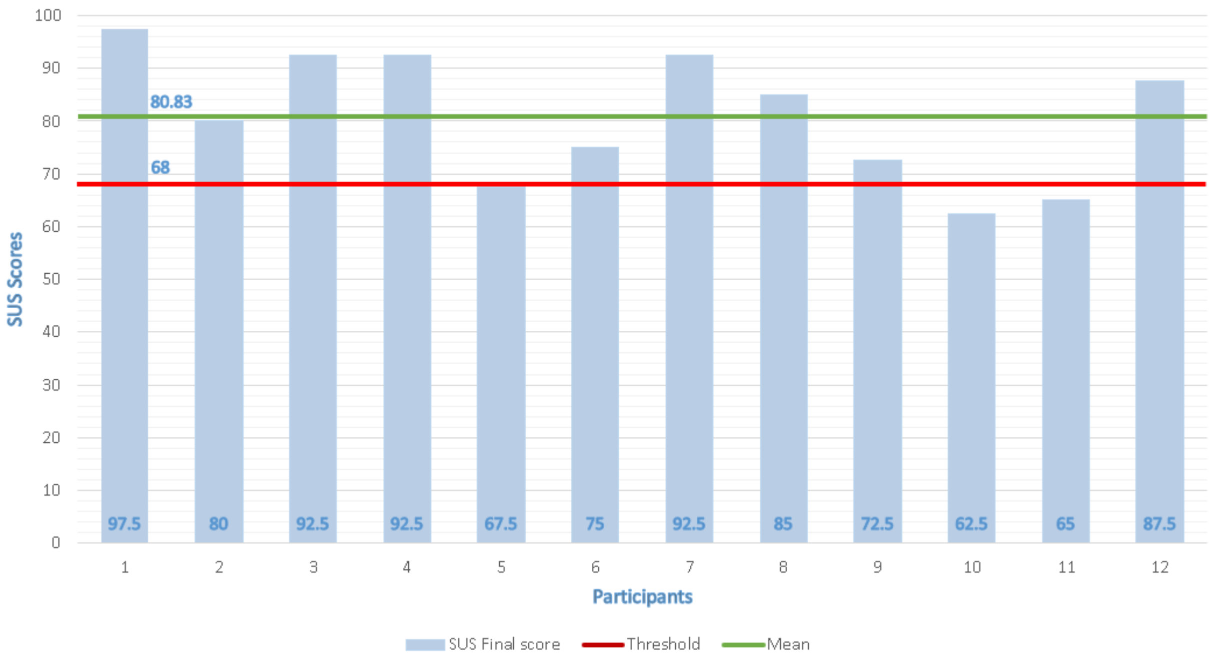


Figure 5.14: SUS scores

5.6.4 Findings from NASA-TLX Questionnaire

The NASA-TLX results (Table 5.5) indicate that the tasks proposed in the study imposed a generally low-to-moderate workload on participants, with notable differences across tasks.

As illustrated in Figure 5.15 (notably, the scale displayed in the figure ranges up to 40 rather than the full 0–100 NASA-TLX range), the weighted scores show that *Task 2* (M = 23.61) and *Task 4* (M = 22.81) elicited the highest overall workload; however, *Task 1* (M = 22.36) presents a comparable level of perceived workload. This similarity may be partially explained by the fact that *Task 1* was the first activity to be performed, and participants were still familiarizing themselves with both the system and the interaction workflow. In contrast, *Task 5* resulted in requiring the lowest workload (M = 13.78),

Table 5.5: NASA-TLX weighted score per task

Task	**Mean**	**Standard Deviation**
Task 1	22.36	13.63
Task 2	23.61	12.21
Task 3	17.72	10.27
Task 4	22.81	16.85
Task 5	13.78	9.59

consistent with its more linear and less cognitively demanding structure. This pattern reflects the varying cognitive demands observed during the study, as Task 2 and Task 4 required more articulated reasoning and a higher degree of attentional engagement, while Task 5 involved a simpler and more linear interaction sequence.

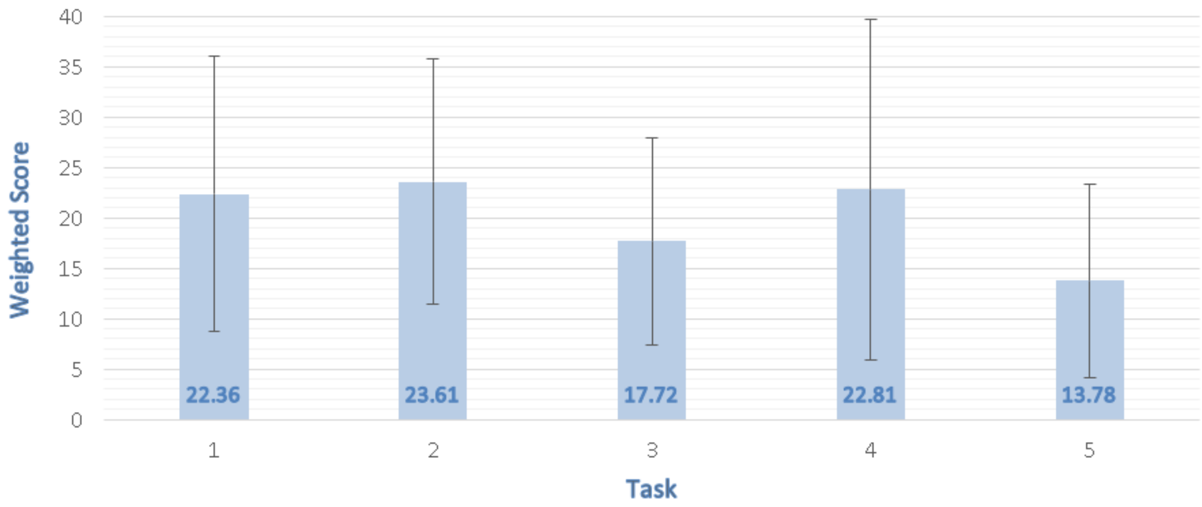


Figure 5.15: NASA-TLX weighted score per task

A more detailed analysis of the workload sources, presented in Table 5.6 and visualized in Figure 5.16 (considering that the scale ranges up to 30 rather than 100 for visualization purposes) shows that *Mental Demand* was consistently one of the highest-rated dimensions, particularly in *Task 2* (28.75) and *Task 4* (28.33), indicating substantial cognitive processing requirements when formulating the task logic and interpreting system feedback.

Table 5.6: NASA-TLX dimensions score per task

Dimension	**Task 1**	**Task 2**	**Task 3**	**Task 4**	**Task 5**
Mental Demand	27.50	28.75	17.50	28.33	18.33
Physical Demand	0.42	0.42	0.00	0.42	0.42
Temporal Demand	24.17	22.92	20.00	22.50	16.25
Performance	10.83	10.42	7.92	14.17	2.50
Effort	26.67	30.00	20.83	24.17	17.50
Frustation	15.00	15.42	12.50	12.50	5.83

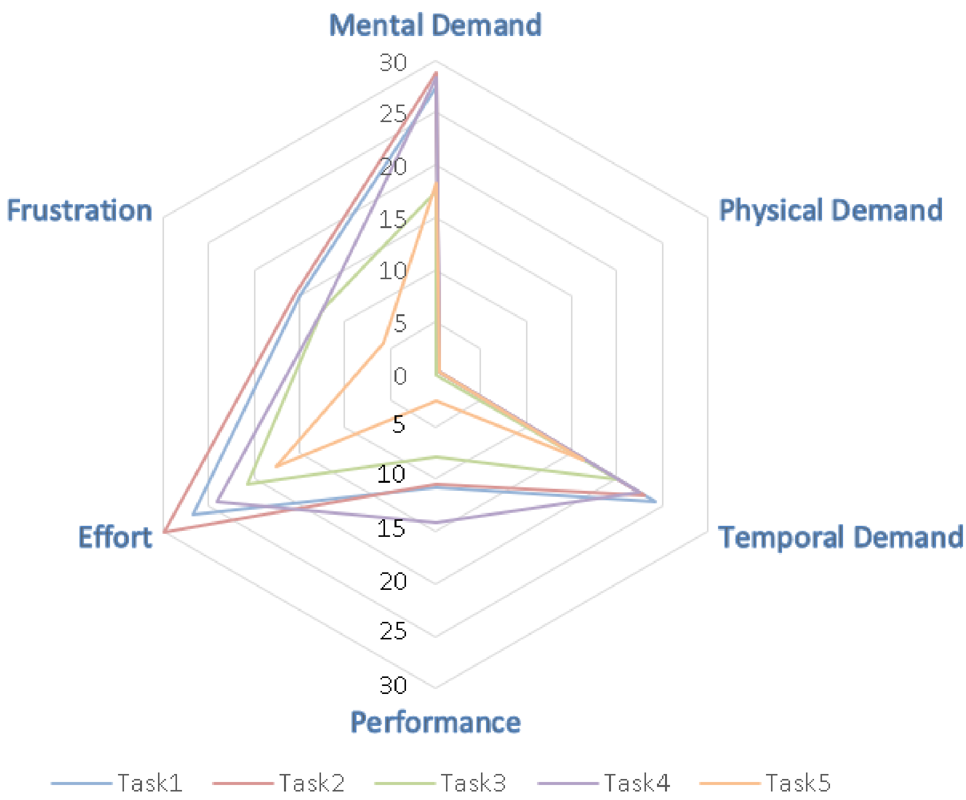


Figure 5.16: NASA-TLX dimensions score per task

Physical Demand remained negligible across all tasks (0.00–0.42), confirming the predominantly cognitive nature of the interaction. *Temporal Demand* showed moderate values, with higher perceived time pressure in *Task 1* (24.17) and *Task 2* (22.92), and the lowest in *Task 5* (16.25). The *Performance* dimension highlights a generally high perception of success, especially in *Task 5* (2.50). *Effort* mirrored the trend observed in *Mental Demand*, peaking in *Task 2* (30.00) and reaching its minimum in *Task 5* (17.50). Finally, *Frustration* scores remained relatively low overall, decreasing steadily from *Task 1* (15.00) to *Task 5* (5.83), suggesting growing familiarity and confidence with the environment.

Overall, the data indicate that although certain tasks required increased cognitive engagement, particularly those involving multi-step reasoning or complex workflows, the overall workload remained within acceptable limits. The progressive reduction in both workload and frustration across tasks further suggests that the multimodal environment supports good learnability and reduces user effort as familiarity increases.

5.6.5 Findings from the Final Interviews

The interview questions presented to participants are reported in Table 5.7.

Further themes emerged from the thematic analysis of the data gathered through the interviews. Table 5.8 summarizes these themes with their corresponding codes.

Table 5.7: Questions of the final semi-structured interview

1	How was your overall experience using the application, from the initial chat interaction to the simulation in the digital twin and the execution on the robot?
2	How easy and natural did it feel to communicate with the system in natural language to define the robotic tasks?
3	To what extent did the block-based representation help you understand and modify the task generated by the chat?
4	Did the simulation in the digital twin help you determine whether the task was correct? If so, did it give you confidence in the final result?
5	What would you change or add to make the application clearer, more useful, or easier to use?

Table 5.8: Themes and codes emerged from the thematic analysis of the final interviews

Theme	**Codes**
Intuitiveness and ease of use	• Appreciation of graphic interface • Ease of using pre-formed constructs (blocks) • Expectation–reality mismatch • Naturalness and accessibility of interaction
Interaction with the chat	• Error tolerance and linguistic robustness • Continuous feedback • Preference for chat over blocks • Linguistic flexibility and command-structure uncertainty
Comprehensibility of block-based interface	• Modifiability and error correction • Need for guides, examples, or tutorials • Integration of chat and blocks
Usefulness of simulation for verification and confidence	• Validation despite execution slowness • Perceived redundancy for more familiar users • Detection of errors and safety checking
System limitations and proposal for improvement	• Expanded documentation with graphical examples • Improved visibility of interface elements and navigation flow • Enhanced feedback in the simulation page • Pre-configurable tasks to reduce chat verbosity

Intuitiveness and ease of use - Participants consistently highlighted the overall intuitiveness of the system, reporting that both the graphic and conversational modalities contributed to making the interaction more straightforward. All the users described the application interface as *"easy and intuitive"* (P2) and *"comprehensible and simple, designed to help the user"* (P5). The graphic interface was particularly appreciated for its clarity and low cognitive demand, with participants noting that predefined constructs and graphic blocks streamlined task composition (P9). A recurring observation concerned a positive mismatch between expectations and actual difficulty: four users expressed initial hesitation toward interacting with a robot programming tool but ultimately found the process easier than anticipated. As P10 commented, the interaction was *"easier than expected"*, a sentiment echoed by P7. The perception of naturalness also emerged: for instance, P4 described the system as *"easy, natural to use"*. These reports suggest that the multimodal design contributed to lowering barriers associated with robot programming tasks.

Recommendations: To further strengthen intuitiveness, the environment may benefit from lightweight onboarding sequences that explain key elements as users encounter them. Maintaining simplicity while progressively revealing advanced features can sustain the positive expectation–reality shift observed among participants.

Interaction with the chat interface - The chat was widely regarded as a flexible and accommodating interface. Participants valued its linguistic robustness, noting that *"the chat accepts mistakes and corrects them"* (P3), and appreciated the continuous feedback (e.g., the confirmation messages after each request and the final summary) that supported their understanding of ongoing actions (P5). For five users, the chat was perceived as more immediate than the block-based interface, with P10 and P12 explicitly stating that it felt *"very simple to use"*. However, the open-ended nature of natural language occasionally introduced uncertainty. Two participants expressed doubts about how to phrase commands when lacking a clear reference structure: P11 noted having *"some doubts on how to use the chat in the absence of base commands"*. Others, while appreciating the flexibility, highlighted that its less structured nature could make it harder to predict system behavior compared with block-based programming. Still, the majority emphasized conversational fluency and low effort as major strengths, describing the interaction as *"comfortable, not too schematic"* (P12).

Recommendations: Providing subtle scaffolding, such as optional starter prompts, examples of effective phrasing, or an expandable list of suggested intents, could help support users who are unfamiliar with task formulation. Maintaining a balance between linguistic freedom and gentle guidance may enhance confidence without constraining natural expression.

Comprehensibility of block-based interface - The block-based interface was widely appreciated for its determinism, transparency, and ease of correction. Participants noted that blocks were *"very intuitive, schematic, and easy to correct"* (P7), allowing them to understand and refine task structures without ambiguity. For five users, blocks served as a clarifying medium that made the logic *"super clear"* (P3), especially for longer or more complex tasks (P6). However, two participants reported the need for initial guidance. P8 and P10 expressed that working with blocks alone, without introductory examples or a brief tutorial, could be challenging at first. Four participants framed block-based interaction as a complement to chat-based interaction, consistent with the idea of multimodal synergy: P12 summarized this by stating that *"blocks are the right complement"*.

Recommendations: To support first-time users, introducing quick visual examples, embedded templates, or a brief tutorial illustrating common block patterns could accelerate familiarity. Progressive disclosure, initially showing only essential categories and revealing advanced blocks later, may help prevent early overload. Reinforcing multimodality through cross-linking (e.g., showing which chat instructions generated which blocks) could deepen understanding and transparency.

Usefulness of simulation for verification and confidence - The digital twin simulation was seen as a valuable verification mechanism that helped users identify potential errors before real execution. Most of the participants explicitly highlighted its role in error detection: for instance, P10 noted that the simulation *"allows you to notice errors"*. Others described it as a meaningful support for safety-oriented validation, suggesting its utility for ensuring that robot behaviors were consistent and risk-free (P6). Yet, perceptions varied depending on expertise. A user with greater familiarity in structured workflows viewed the simulation as redundant: P3 considered it *"quite useless because the block schema was already sufficient"*. A minor downside reported by two participants was the simulation speed, described as *"a bit slow"* (P1, P2). Despite these limitations, most participants acknowledged that the simulation contributed to increased trust and clarity, especially during the final verification stages.

Recommendations: Enhancing simulation responsiveness and providing clearer progress indicators could reduce perceived slowness and foster more seamless validation cycles.

System limitations and proposals for improvement - Participants provided several constructive suggestions aimed at refining the system. A recurring request concerned expanding documentation with graphical examples or integrated tutorials (P2, P8, P10, P11). Interface affordance and visibility also emerged as an issue: unclear icons, ambiguous navigation flows, and missing controls, such as *"the exit button from the chat"* (P3, P5, P11), occasionally hindered user orientation. Feedback related to the simulation page included desires for more persistent confirmation messages: for instance, P3 remarked that the *"confirmation popup should remain visible"*, while P11 suggested implementing the automatic closing of simulation windows. A participant also proposed mechanisms to reduce verbosity in the chat, suggesting the introduction of "pre-configurable tasks" (P4) to minimize repetitive definitions. Overall, these comments highlight the importance of clear visual cues, consistent navigation, and tailored documentation to support users with varying levels of experience.

Recommendations: Enhancing internal documentation through the use of short, illustrated examples, interactive tutorials, and integrated help panels could substantially improve learnability. Strengthening the visual hierarchy and persistence of feedback, particularly in modal windows and simulation messages, would support consistent navigation. Finally, adopting customizable shortcuts or pre-configured templates may streamline interactions and reduce verbosity in natural language exchanges, especially for common task patterns.

5.7 Discussion

This section discusses the findings, open issues, and future directions derived from this research, with a focus on RQ2, which explores the capabilities and limitations of natural language interfaces in supporting users to accurately specify, validate, and refine structured robotic tasks, and examines how multimodal interaction can enhance this process during task definition.

The proposed EUD environment fosters a comprehensive workflow for robot programming, which starts by defining the items of the application domain, then specifies the robot task through the chat and the block-based interfaces, continues with testing the defined task by means of a digital twin, and ends with the task execution on the physical robot. This workflow has been conceived following a human-centered AI approach to designing interactive systems [135]. This allowed me to develop ADMIRABLE, an EUD environment that incorporates non-deterministic technologies like LLMs and prioritizes the user's decision, also addressing issues that may occur, such as incorrect robot task generation or LLM hallucinations.

With respect to other approaches to end-user robot programming (e.g., [149, 81]), my objective was not to generate the final code for the robot based on the user's input using LLMs, but rather to obtain a task representation that could be processed in the EUD environment, enabling users to evaluate the program correctness via Blockly, which was tailored to represent pick-and-place robot tasks. In this way, end users can visually inspect robot programs and modify them through the drag-and-drop of blocks, or direct manipulation of the variables corresponding to users' defined items.

Compared to other applications developed using simulators [130] or employing Gazebo and ROS [118, 93], the proposed system does not force users to engage with technical languages or tools. Instead, users can interact with the digital twin through a simple, web-based application designed specifically for them and consistent with their domain expertise.

Moreover, the simulator used in this work is not a purely graphical 3D representation (e.g., simulations based on *Three.js*[17]), where the simulated world and behavior are entirely pre-defined by the programmer. Instead, Gazebo provides a physics-based simulation environment with realistic interactions that include friction and forces. Indeed, creating the robot model, which included all relevant physical properties and forces, required a significant investment of time and effort. Since Gazebo is ROS-based, the same architecture can be employed to control the physical robot directly.

The user study provided encouraging evidence of the effectiveness and usability of the proposed EUD environment. Participants, who generally had very limited prior experience with robotics or programming, successfully completed all tasks, including those involving conditional logic, multi-step reasoning, and verification through the digital twin. Quantitative results showed near-perfect task completion, and the SUS scores fell within the *Excellent* range and *Grade A*, while NASA-TLX scores indicated only *low-to-moderate* workload. Although some initial uncertainty was observed during the first interactions, the data and user comments consistently highlighted that familiarity increased rapidly, resulting in smoother and more confident use of both the chat-based and graphic interfaces. Overall, the study has demonstrated that the system is highly accessible and offers a positive interaction experience despite its inherently technical nature. Although not explicitly reported as a theme in the thematic analyses, users also greatly appreciated using the application, which sparked curiosity and provided a sense of enjoyment.

Beyond the engineering and empirical contributions, the results of this work, as well as the results obtained with PRAISE development and evaluation (see Chapter 4), can also be interpreted from a broader conceptual perspective on EUD. In particular, the proposed environment combines two traditionally distinct EUD approaches [12]: natural language interaction, enabled by LLMs, and structured, graphic representations of robot tasks supported by a component-based interface.

Rather than treating these approaches as alternatives, this work shows how they can be meaningfully integrated into a single EUD workflow. Natural language interaction primarily supports the expression of high-level intentions and goals, lowering the initial barrier to task definition, while the component-based interface enables inspection, verification, and refinement of the generated task structure. This combination allows end users to move fluidly between expressive, intention-oriented interaction and more reflective, control-oriented activities, supporting both creativity and correctness in robot task specification.

From a conceptual standpoint, this integration can be seen as an instance of a medi-

[17]Three.js - https://threejs.org

ated EUD process, in which generative AI acts as a mediator between the user's situated knowledge and the formal representations required by the robotic system. In this sense, the LLM does not replace the end user's role in task construction, but rather externalizes part of the cognitive effort involved in translating domain-level intentions into executable robot programs. This perspective aligns with meta-design principles [44], where systems are intentionally left open and underdesigned, enabling users to actively shape and appropriate system behavior over time rather than merely consuming predefined functionalities. A broader reflection on the relationship between meta-design and collaborative robotics is provided in Section 5.7.3, where the implications of this perspective are discussed in connection with system sustainability and long-term appropriation.

Moreover, the interaction among the user, the LLM, and the robot can be interpreted through the lens of distributed cognition [73, 71]. Task definition and validation emerge from the coordinated interaction of multiple cognitive artifacts: the externalized expressions of the user's domain expertise, the LLM's generative and interpretative capabilities, and the robot's explicit task model and simulated behavior in the digital twin. Cognition is therefore not localized in a single component, but distributed across representations and interaction modalities, with the digital twin playing a crucial role in grounding abstract task descriptions in observable and inspectable robot behavior.

Finally, these findings also contribute to the discussion on human-centered AI in end-user programming. While natural language interaction increases accessibility, the study highlights the importance of complementary mechanisms that promote transparency, user control, and trust. Participants' preference for reviewing and adjusting LLM-generated tasks through visual representations and simulation suggests that trust in LLM-based EUD does not only stem from a natural language interaction, but also from the possibility of verifying, understanding, and intervening in the system's behavior. This observation aligns with recent work [8] on the role of language and terminology in shaping user trust in AI-assisted systems, and reinforces the need for hybrid interaction paradigms that balance expressive freedom with explicit control.

The ultimate goal of this prototype is to promote a method that helps human workers gradually acquire robot programming skills, thereby facilitating the development of users' computational fluency [13, 16]. This approach is expected to enable seamless advancement in programming proficiency.

5.7.1 Design Implications

The findings of this project suggest some design implications that can be applied to other systems combining natural language and graphic interfaces for end-users development:

- **Supporting progressive onboarding**: early stages of system use might generate uncertainty, particularly concerning abstract constructs such as loops and conditionals. This highlights the need for progressive onboarding strategies, including contextual explanations, embedded examples, and dynamic hints that appear only when needed. Such mechanisms would help reduce cognitive load during first usage and provide a smoother transition from novice to proficient use.

- **Enhancing transparency across modalities**: the interplay between the chat-based interface and the graphic interface proved effective, even though users might rely on system suggestions without fully understanding the underlying logic. Ensuring that both interfaces provide transparent, consistent explanations, such as explicit summaries, generated actions, or visual confirmations of logical structures, can strengthen users' mental models and promote more informed decision-making.

- **Improving feedback for complex structures**: tasks involving conditional logic or multi-step workflows might pose difficulties for non-technical users and require a higher workload. This suggests the need for clearer and persistent feedback when users manipulate complex structures. Visual cues, state highlights, and real-time validation indicators could help users detect and resolve inconsistencies earlier in the process.

- **Facilitating learning through exploration**: users usually like to explore the capabilities of a system, e.g., in the case of chat-based interaction, by testing linguistic variations, and adopting alternative strategies. The system should explicitly support this behavior by tolerating diverse expressions, offering optional documentation, and enabling users to revisit earlier steps without penalty. Encouraging exploratory interaction can accelerate learning and foster a deeper understanding of the task-definition process.

- **Considering accessibility and cognitive styles**: the presence of users with heterogeneous cognitive preferences, such as those with dyslexia, or with different types of disabilities, suggests the importance of providing customizable interaction modalities. Adjustable text size, alternative iconography, careful use of colors and shapes (e.g., for color-blind users), and multimodal summaries can help ensure that the system remains inclusive and accessible to a broader population.

5.7.2 Limitations of the Work

The current system implementation reveals some areas for improvement.

The first one concerns the complexity of the tasks that can be defined. In the chat, only the generation of tasks with two levels of nested logic has been implemented. To define more complex tasks, users must possess adequate computational thinking skills. It is essential to further investigate ways to enhance task complexity while minimizing the required technical expertise of end users.

Furthermore, the mapping between the physical environment and its digital twin could be made more realistic. In particular, aspects such as photo capturing by the camera and simulation of the surrounding environment could be more accurately represented. A related limitation concerns the representation of user-defined objects and locations: the system currently does not automatically generate 3D models for user-defined items, which must be manually provided. Future work will explore approaches such as generating models from photographs to support fully user-defined virtual environments.

Additionally, ADMIRABLE currently supports relatively simple tasks (e.g., pick-and-place with processing actions) and requires enhancements to effectively define and simulate more complex tasks, such as collaborative tasks with steps where the human is an active participant in the collaboration.

Finally, in the user study, the average age of participants was relatively low (M=25.92, SD=5.32). This is mainly because the study required participants to be physically present in the university laboratory during working hours, which limited recruitment primarily to students who were available at those times. As a result, the sample consisted mainly of younger participants. Moreover, the participants were not members of the intended target user group (i.e., healthcare professionals), which may further limit the generalizability of the findings to the population for whom the system was designed.

5.7.3 Related Project

In addition to the aspects discussed in this section, further work has explored how collaborative robots can contribute to achieving the United Nations' Sustainable Development Goals (SDGs)[18] by adopting a meta-design perspective [44]. Meta-design conceptualizes systems as open and evolvable socio-technical environments in which design does not end at deployment, but continues at use time through the active participation of end users [45]. Within this framework, end users are not merely recipients of predefined functionalities but are empowered to appropriate, adapt, and extend technologies according to their situated needs and practices.

This perspective is applied to collaborative robotics in *"A Meta-Design Approach to Collaborative Robotics to Achieve Sustainability Goals"*, presented at the *9th International Workshop on Cultures of Participation in the Digital Age (CoPDA) 2025 at the 10th International Symposium on End User Development (IS-EUD) 2025 (Munich, Germany)* and published in *CEUR Workshop Proceedings* [56]. The work examines the relationship between cobots and sustainability by framing their use within a meta-design approach that accounts for the specific application domain, the characteristics of end users, and the social dynamics of the workplace. By adopting a meta-design approach (Figure 5.17), cobot usage is reframed as a participatory process in which end users can not only configure tasks but also share and iteratively refine them within their professional communities. This orientation strengthens the contribution of cobots to sustainability goals by fostering adaptability, inclusivity, and long-term appropriation of the technology.

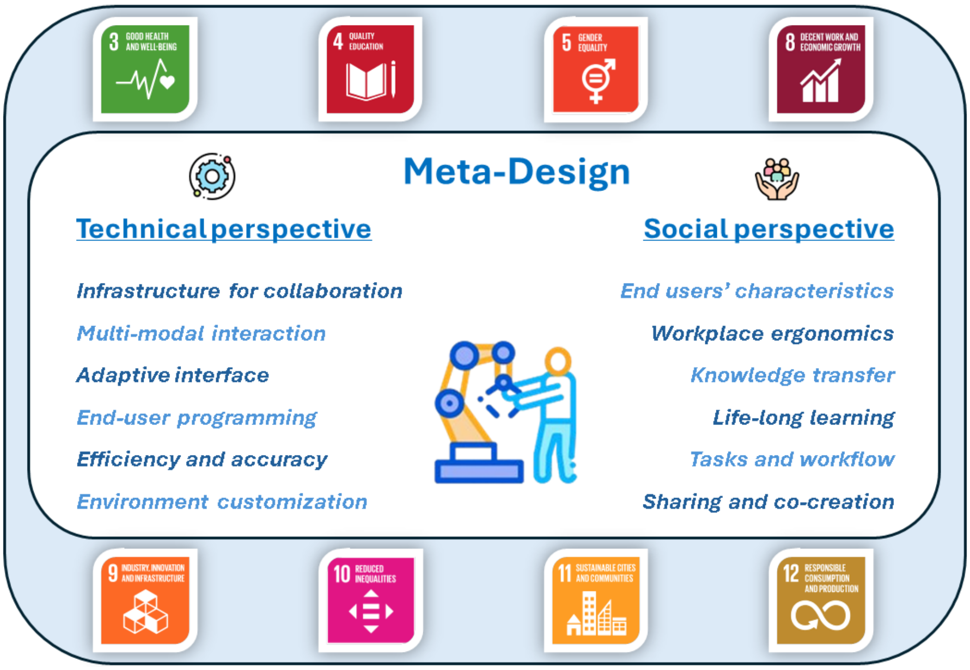


Figure 5.17: The role of meta-design in achieving sustainability goals through collaborative robotics

[18]United Nations, Sustainable Development Goals (SDGs) - https://sdgs.un.org/goals

Part II

Human-Robot Collaboration for Task Accomplishment

Chapter 6

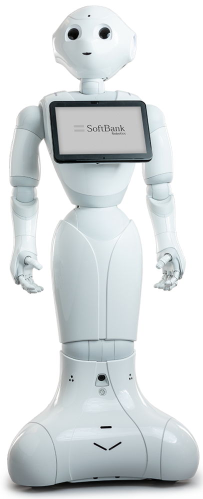
Human-Robot Interaction

Human-Robot Interaction is the interdisciplinary field that studies, designs, and evaluates how humans and robots communicate, collaborate, and share physical or social spaces [17]. Unlike traditional robotics, which focuses primarily on autonomous operation in controlled environments, HRI places human users at the center of the design process, emphasizing usability, trust, safety, and mutual understanding [103]. In practice, HRI encompasses a continuum of scenarios ranging from industrial cobots working side by side with operators [114], to service and assistive robots deployed in healthcare [32], and educational contexts [9], to fully social robots engaging in dialogue and companionship with users [1]. Research in HRI integrates concepts from robotics, artificial intelligence, psychology, human factors, and interaction design to create systems that are not only technically competent but also socially and cognitively acceptable to human partners [17]. This chapter introduces the key concepts of HRI relevant to this part of the thesis, beginning with an overview of social robots, introducing the current challenges in HRI, and then examining the architectures and planning mechanisms that underpin human-aware and adaptive interaction.

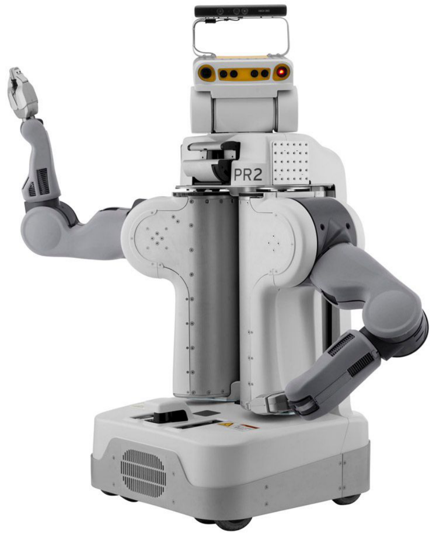
6.1 Social Robots

Social robots represent a class of robotic agents specifically designed to engage with humans in socially meaningful ways, going beyond purely physical or industrial tasks. They are equipped with multimodal interfaces, including speech, gesture, facial expressions, and proxemic behavior, which enable them to interact naturally with people in domestic, educational, healthcare, and public environments. Recent surveys, such as the ones by Mahdi et al. [99] and by Manavi et al. [100], provide comprehensive overviews of the design principles, application domains, and challenges of social robots, showing how these systems are increasingly oriented towards assistance, education, companionship, and healthcare support. These literature works highlight that social robots are being deployed to perform roles ranging from therapeutic and rehabilitative interventions to tutoring and peer-learning activities, as well as in reception, customer service, and entertainment contexts. These applications require robust capabilities in perception, dialogue management, affect recognition, social navigation, and adaptive behavior, along with a high degree of trustworthiness and safety, since the robot must coexist with people in unstructured environments.

Within this landscape, Figure 6.1 shows several robotic platforms that have become reference points in social robotics research and practice.



(a) SoftBank Robotics Pepper



(b) Willow Garage PR2

(c) PAL Robotics TIAGo

(d) SoftBank Robotics NAO

Figure 6.1: Overview of leading social robot platforms representing different approaches to human–robot interaction, ranging from compact educational humanoids to versatile mobile manipulators used in assistive and research contexts

Among them, Pepper[1] (Figure 6.1a) by SoftBank Robotics is a human-sized semi-humanoid robot equipped with a wheeled base, a tablet on its chest, multiple cameras and microphones, and a 3D sensor, enabling it to communicate through speech, gestures, and expressive movements. It has been widely deployed in retail, hospitality, education, and healthcare pilot studies, where its anthropomorphic appearance and interactive features have been found to foster engagement and social presence. PR2 (Personal Robot 2), shown in Figure 6.1b, was initially developed by Willow Garage and constitutes a versatile research platform that combines advanced mobile manipulation with human–robot interaction modules. Since Willow Garage is no longer active, the official website is not available; therefore, a non-official but authoritative archival reference is reported[2]. Although less anthropomorphic than humanoid robots, PR2 provides a rich testbed for integrating navigation, manipulation, and social behaviors, enabling researchers to prototype and evaluate complex collaborative and assistive scenarios. TIAGo[3] by PAL Robotics (Figure 6.1c) exemplifies a semi-humanoid service robot combining a mobile base, a manipulator arm, and multimodal interfaces, including speech recognition and synthesis. Thanks to its modular architecture and ROS compatibility, TIAGo has been adopted in domestic and healthcare studies as an assistive companion, capable of carrying objects, guiding people, or monitoring daily activities while maintaining natural interactions. Finally, NAO[4] by SoftBank Robotics (Figure 6.1d) represents one of the most widely used small humanoid robots in education and therapy. Its compact size, extensive degrees of freedom, and expressive capabilities make it especially suitable for child–robot interaction, classroom activities, and studies on social cues, affective feedback, and proxemics. Each of these platforms represents a different compromise between expressiveness, mobility, autonomy, cost, and robustness, thus reflecting the diversity of approaches in social robotics.

Overall, the increasing diffusion of social robots illustrates a convergence between advances in robotics, artificial intelligence, and human–computer interaction. Their development emphasizes the need for architectures capable of managing perception, dialogue, affect, and task planning in an integrated way, balancing deliberative reasoning with reactive safety mechanisms.

6.2 Current Challenges

As robots become increasingly embedded in various aspects of daily life, ensuring seamless human-robot interaction is of paramount importance. A fundamental challenge in HRI lies in equipping robots with the capability to infer human beliefs and intentions and adjust their behavior accordingly. Unlike traditional automation, where robot actions follow predefined rules, effective HRI requires robots to dynamically interpret and respond to human actions, preferences, and situational changes. This entails not only processing explicit commands but also recognizing implicit cues such as gestures, emotions, and contextual information. Another critical aspect is the ability of robots to plan and coordinate their actions in a manner that ensures transparency, explainability, and acceptability. Robots should assist human collaborators in anticipating and comprehending robotic behavior, thereby minimizing unpredictability that could lead to confusion or inefficiency. Achieving this goal necessitates generating actions that align with human

[1]SoftBank Robotics Pepper - https://us.softbankrobotics.com/pepper

[2]WilloW Garage PR2 - https://robots.ieee.org/robots/pr2

[3]PAL Robotics TIAGohttps://pal-robotics.com/robot/tiago

[4]SoftBank Robotics NAO - https://us.softbankrobotics.com/nao

expectations while adhering to safety considerations, ethical constraints, and usability principles.

Traditional model-based architectures provide several important strengths that make them fundamental in robotics. Their reliance on explicit models and formal reasoning ensures predictability, verifiability, and safety guarantees, which are crucial in safety-critical domains such as industrial automation and healthcare. Furthermore, model-based methods facilitate transparent decision-making and explainability, since the reasoning process and control rules can be explicitly traced and validated. These characteristics make deterministic architectures particularly valuable when reliability and accountability are paramount. However, despite these advantages, their dependence on predefined models and static rules limits adaptability to dynamic human behaviors and context-aware interactions.

The integration of LLMs presents a promising avenue to tackle these challenges. These models significantly broaden the spectrum of scenarios that robots can manage, enabling more flexible approaches to explainability and acceptability. By leveraging common sense reasoning and contextual understanding, LLMs can assist robots in interpreting human inputs and environmental cues, inferring implicit intentions, and generating adaptive responses that align more closely with human expectations. Nevertheless, a fully probabilistic approach remains inadequate for ensuring reliability and safety in HRI. It is essential to develop an architecture in which the generated actions are validated and the critical aspects (such as safety constraints and ethical considerations) are explicitly defined and enforced.

6.3 Architectures for Social Robots

Several studies in the literature have proposed control architectures for adaptive robots capable of effectively interacting with humans [19, 39, 145]. These architectures are designed to integrate decisional and functional components within a unified framework that efficiently manages the information flow. Decisional components typically encompass higher-level processes such as situation assessment and planning, which are essential for enabling robots to make informed decisions based on human presence and environmental context [90, 37]. Conversely, functional components, including perception and action, allow robots to engage with the physical world and dynamically respond to their surroundings in real time [113, 46].

The primary challenge lies in structuring these heterogeneous components into a cohesive architecture that can meet the system's complex requirements. This process involves identifying the operations required to accomplish each task, defining their temporal interdependencies to ensure coherent execution, and assigning appropriate decisional or functional mechanisms to effectively address the specific challenges that may arise in HRI. However, a key limitation of many existing architectures is their reliance on predefined models of human beliefs, intentions, and preferences [70]. Given the inherent unpredictability of human behavior, accurately modeling these aspects remains a significant challenge. This constraint limits the system's capability to fully anticipate and adapt to human actions, introducing a layer of uncertainty that is difficult to mitigate. As a result, achieving a truly effective integration between decisional and functional components remains an open research objective. Such integration is crucial for enabling robots to manage both the strategic and tactical aspects of their behavior, adapt to dynamic environments, and engage in meaningful, context-aware interactions with humans.

Cognitive architectures that incorporate belief management and Theory of Mind (ToM) focus on how robots represent and reason about human mental states [107, 127]. In HRI, different actors possess distinct background knowledge, varying levels of task and environmental awareness, and potentially conflicting perspectives. To cope with this complexity, a robot must continuously maintain, update, and reason about diverse belief systems, ensuring that the most current and relevant information informs its actions. Belief-management frameworks play a fundamental role in this process, enabling the robot to infer, track, and reconcile different viewpoints before making decisions or executing actions.

6.4 Planning of Human-Robot Tasks

Beyond control architectures, various frameworks have been developed to support the generation of action plans for collaborative tasks between humans and robots [69, 119, 60]. These planning systems enable robots to formulate effective action strategies while operating alongside human partners. A primary challenge in these systems lies in ensuring that the generated plans account for both the robot's capabilities and the human's behaviors and expectations. Most planning systems, including those incorporating adaptation or learning mechanisms, are inherently model-based [124]. They depend on predefined models representing the human, the task, and the interaction dynamics between the human and the machine. These models are essential for predicting human actions, inferring intentions, and anticipating task evolution. However, given the inherent unpredictability of human behavior, uncertainty remains a critical factor in these systems, necessitating the development of advanced methods to address it.

Markov Decision Processes (MDPs) [67] and Partially Observable Markov Decision Processes (POMDPs) [146] are commonly employed to model uncertainty in the environment and in the robot's observations of the human's state, allowing estimation of action outcomes under incomplete or noisy information. MDPs are particularly useful when the outcome of an action is uncertain, requiring decision-making that balances immediate and future rewards. POMDPs extend MDPs by considering scenarios in which the robot has incomplete information about the environment or the human's state (the latter is an essential factor in human-robot interaction). MDPs and POMPDPs models enhance decision-making by mitigating the uncertainty inherent in human behavior and environmental variability.

Epistemic planning explicitly models the divergence in beliefs between the robot and the human [29]. By incorporating each agent's knowledge of the other agent's beliefs and intentions, epistemic planning facilitates coordination, fostering seamless interaction between humans and machines. Additional approaches include non-deterministic planning frameworks, where human decisions and actions are treated as contingencies [42]. While these methods effectively guide the robot's actions under uncertainty, they do not fully capture how humans think and make decisions. To address this, cognitive models such as ACT-R/E (Adaptive Character of Thought-Rational/Embodied) [142] have been integrated into planning processes, providing a framework to simulate human reasoning and problem-solving strategies. By combining epistemic reasoning with cognitive modeling, robots can not only account for differences in beliefs but also anticipate human behavior more accurately, resulting in more effective and natural collaboration.

To ensure that the robot's actions are both acceptable and pertinent, several criteria have been proposed, often framed in terms of costs and benefits, to assess the efficiency and relevance of different action plans. Frameworks such as HATP (Human Aware Task

Planning) [5] have been used to define these criteria, guiding the robot's decision-making process by considering both the practical costs of actions (e.g., time, resources) and their expected benefits (e.g., effectiveness, satisfaction). This approach allows robots to plan actions that align with human preferences and expectations, optimizing collaborative task achievement. By integrating these advanced methodologies, robots can better adapt to the uncertainties of human behavior, improve task execution in dynamic environments, and ensure that their actions align with the collaborative goals of human-robot interaction.

Despite significant advancements, existing approaches to HRI continue to struggle with the fundamental challenge of mind reading and the inherent difficulty of accurately representing human beliefs and decision-making processes. This limitation makes it challenging for robots to interpret complex or nuanced human intentions. The effectiveness of these methodologies relies on the accuracy of models representing the environment, the task, and the strategies for task execution. However, constructing a comprehensive and robust model of the human remains a major obstacle due to the unpredictability of human behavior. The vast range of possible scenarios raises concerns about the viability of relying exclusively on predefined models for human-robot interaction.

Recent contributions have highlighted the potential of LLMs in addressing the above challenges. In [157], LLMs are described as a "paradigm shift" for robotic systems, enabling unprecedented levels of natural language understanding and task execution. Similarly, in [79] it is reported that LLM-based agents can interpret complex natural instructions, dynamically plan robot movements, and execute actions in real time without task-specific supervised training. The integration of multimodal LLMs further enhances robots' ability to solve linguistic ambiguities and infer implicit user intentions. In [120], an adaptive framework in which an LLM acts as a bridge between the user and the robot is proposed, tailoring interaction strategies to user characteristics such as age; for example, with older adults, the system simplifies dialogue and makes robot intentions more explicit, thereby improving usability and transparency.

Despite these advantages, significant limitations remain for adopting LLMs in HRI. In [64], it is reported that the probabilistic nature of LLMs, combined with the absence of formal guarantees, poses risks to safety in real-world deployments. In particular, LLM outputs may suffer from factual inaccuracies, and more critically, LLM-driven task planning often lacks temporal coherence, leading to unstable or unsafe robot trajectories and actions. In [61], it is discussed that while LLMs improve the naturalness of interaction, classical symbolic planners still outperform them in terms of reliability and efficiency of task planning. These findings suggest that while LLMs substantially advance natural dialogue and adaptability, they remain insufficient for ensuring robust, safe, and temporally consistent robot actions. Therefore, the integration of LLMs with classical model-based approaches is proposed in this PhD thesis.

Chapter 7

A Hybrid Architecture for Flexible and Adaptive Robot Coaching

In HRI, unpredictability in human behavior poses a major challenge for maintaining task efficiency, safety, and smooth interaction. Robots must not only follow structured plans, but also adapt to dynamic, context-dependent situations in real time. Addressing this challenge motivates the formulation of RQ3: *How can robots adapt to the unpredictable behavior of human partners in collaborative scenarios, while maintaining task control and ensuring safety during execution?*

Given the respective strengths and limitations of traditional architectures and model-based methodologies, as well as the emerging capabilities and challenges of LLMs, I propose in this chapter a hybrid architecture that integrates the structured reliability of deterministic models with the adaptability, common sense reasoning, and contextual awareness offered by LLMs. In this context, the term *hybrid* refers to the combination of deterministic techniques, which provide formal guarantees and rule-based precision, with non-deterministic, probabilistic methods that enable learning, adaptability, and robustness in uncertain environments.

This work investigates how far a hybrid LLM/model-based architecture can enhance task execution, adaptability, and user interaction in HRI, in comparison with purely model-based or purely LLM-based approaches. It is hypothesized that integrating LLMs with model-based architectures supports adaptability to human unpredictability and decision-making efficiency, and fosters natural and intuitive interactions while maintaining safety and reliability through deterministic components.

By integrating these complementary paradigms, the objective is to develop a flexible and human-aware HRI system that can operate effectively in diverse real-world settings while maintaining verifiable safety and transparency.

At the time of writing, the work presented in this section is under review in the *International Journal of Social Robotics*. A preliminary version, introducing the idea and entitled *"Towards a Hybrid LLM/Model-Based Architecture for Robot Coaching: An Instance of Human-Machine Collaboration"*, was presented at the *6th Automation Experience (AutomationXP) 2025 Workshop at the Conference on Human Factors in Computing Systems (CHI) 2025 (Yokohama, Japan)* and published in the *CEUR Workshop Proceedings* [54].

7.1 Approach

HRI presents various challenges, ranging from deterministic task execution to highly dynamic and ambiguous human behaviors. In this context, the goal of the HRI system is established and maintained as a shared goal between the human and the robot (i.e., Human-Robot Joint Action [33]), ensuring that both the robot and the human have a common understanding of the desired outcome and can contribute to reaching it. This shared goal guides the robot's actions and decisions, aligning its behavior with human expectations. Achieving a goal requires both humans and robots to perform a series of tasks. Each task represents a specific activity, which may range from simple predefined operations to adaptive behaviors demanding real-time situation assessment. The decision process about the task to accomplish is essential because there could be different ways to achieve the same goal.

Nevertheless, HRI is predicated on certain assumptions that must be examined. Humans and robots are inherently different entities. The robot's purpose is to assist the human in effectively accomplishing a task (i.e., *robot coaching*). Furthermore, the human is regarded as an unpredictable entity, yet it is presumed that they will collaborate with the robot and will not deceive it with malicious intent. Both the robot and the human participate in the accomplishment of the task, employing multimodal (i.e., speaking and acting) interaction.

7.2 Architecture

Altogether, the challenges and assumptions discussed above and in Chapter 6 underscore the importance of a structured approach, leading to the hybrid LLM/model-based architecture for HRI proposed in this project and shown in Figure 7.1.

The architecture is composed of interconnected modules that work collaboratively to ensure seamless interaction, reliable task execution, and robust adaptability to evolving scenarios.

Complementing these modules, a *Vector Database* serves as the central knowledge repository. It stores embeddings derived from unstructured data, including information about the user, the task, and the environment. Furthermore, it is continuously enriched with knowledge from prior interactions, such as user preferences inferred from past interactions. This repository supports task definition and monitoring progress by providing relevant contextual information.

In the following, it is pointed out in which modules a deterministic approach is used (i.e., green rectangles) and where an LLM is used (i.e., yellow hexagons), along with the reasons for the choice in each case.

7.2.1 Knowledge Manager

This module leverages an embedding model to generate embeddings from natural language text, such as text files. Beyond extracting semantic representations of document content, it also appends relevant metadata, such as user identifiers, to each embedding. This ensures that the retrieved information remains user-specific and contextually relevant. The generated embeddings are then stored in the *Vector Database* as a foundation knowledge for subsequent processing steps.

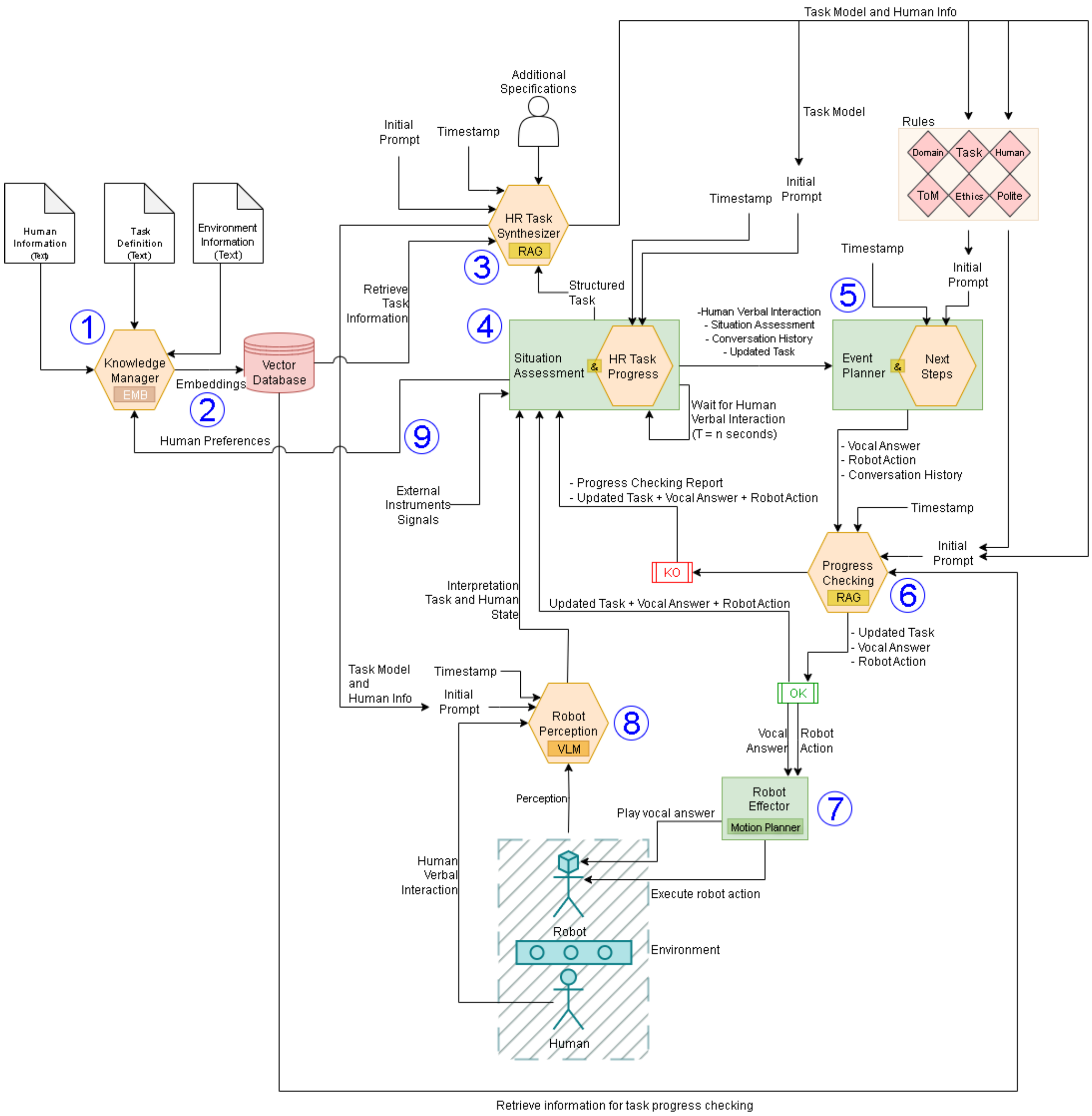


Figure 7.1: The proposed hybrid LLM/model-based architecture

Additionally, when a daily task is completed, the *Knowledge Manager* is responsible for analyzing the interaction history by identifying key insights and extracting user preferences. These insights are encoded into embeddings and integrated into the *Vector Database*, ensuring continuous system adaptation based on newly acquired user-specific knowledge. To enhance the accuracy and relevance of the stored information, an LLM is employed during the information extraction phase from the interaction history.

7.2.2 Human-Robot Task Synthesizer

This module leverages an LLM and Retrieval-Augmented Generation (RAG) to convert unstructured information from the *Vector Database* into a structured task representation. This transformation is crucial for transitioning from natural language descriptions to a structured task plan formatted in JSON that can be programmatically processed. Additional specifications, including timestamps, are incorporated to generate precise task representations. Furthermore, it is possible to specify additional information in natural language for that specific iteration. The JSON format has been selected due to its compatibility with LLMs, which, when appropriately instructed through function calls (usually called also *tool calling* or *function calling*), can generate precise, structured JSON objects. The information retrieved from the *Vector Database* is also useful for other architecture modules to be context-aware about the task and the user.

7.2.3 Situation Assessment & Human-Robot Task Progress

This module combines a rule-based approach (implemented through a *Situation Assessment* component) with an LLM (*Human-Robot Task Progress* component). The *Situation Assessment* functions serve as the high-level control unit, responsible for parsing the request of a structured task, interpreting the task and human state, and managing the history of interactions, including recorded verbal responses and robot actions. Additionally, this component evaluates progress checking reports, ensuring task completion, and, if discrepancies arise, it refines the output of other modules. Moreover, it incorporates predefined conditions that can trigger immediate deterministic reactions without passing through the module responsible for deciding the next steps (i.e., *Event Planner & Next Steps*). For example, if the user attempts to alter the task or skips a mandatory activity, these behaviors activate an immediate response to address the situation (e.g., calling a human supervisor). The activation of such responses is managed by the *Human-Robot Task Progress* component, which monitors whether the conditions are occurring. This module can also process signals from external instruments, allowing the system to react deterministically to specific parameters (e.g., temperature of the room).

Through a rule-based approach, the *Situation Assessment* component orchestrates the workflow by managing the execution of other modules. It verifies task completion, updates the *Vector Database* with newly acquired knowledge, such as user preferences and behavioral patterns, and ensures alignment between the human-robot interaction and task objectives. Furthermore, it governs the robot's behavior, allowing it to engage the human proactively or reactively, responding to human requests and environmental cues. A control loop with a predefined waiting period is implemented to handle scenarios where both the robot and the human remain passive.

The *Human-Robot Task Progress* component employs an LLM to monitor task execution and ensure that the process aligns with predefined objectives. The LLM is initialized with the task model and the invocation timestamp, providing temporal awareness. Its primary function is to update the task's status based on information received from the *Robot Perception* module. The architecture's flexibility is reinforced by the absence of a rigid algorithm governing task progression, thus preventing excessive constraints and enhancing adaptability to dynamic conditions.

In summary, the *Situation Assessment & Human-Robot Task Progress* module is responsible for updating the state of the world, the task itself, and the human's beliefs, preferences, behaviors, and goals.

7.2.4 Event Planner & Next Steps

This module consists of an event planner and an LLM that determines subsequent actions. The *Event Planner* is a deterministic component, specifically an expert system, responsible for managing the sequence of activities within a task in a feasible order, considering scheduled times with tolerances, priorities, criticalities, and durations. It also accommodates unscheduled activities, placing them in available time slots. Furthermore, if two activities with the same priority can be carried out, the user can choose between them.

The expert system is based on a classical architecture that includes a knowledge base (storing facts and if–then rules), a working memory that represents the current state of the system, and an inference engine that applies the rules to the working memory using a forward-chaining methodology to derive new facts or trigger specific actions. This structure allows the system to reason in a deterministic and explainable manner, ensuring transparency and predictability in decision-making. While this rule-based approach is effective for routine and well-defined scenarios, the module could be replaced in the future by an automated planner for more complex tasks involving conditional dependencies or long-term objectives. This planner could break down high-level objectives into atomic activities, sequence them according to temporal and logical constraints, and adapt dynamically as the state of the system evolves.

The *Next Steps* LLM is tasked with determining the robot's next action and generating a verbal response for the user, ensuring coherence with the planned sequence of activities. This model is initialized with a comprehensive prompt, including task-related instructions, contextual rules, ethical considerations, and user-specific information.

It is important to note that the *Human-Robot Task Progress* and *Next Steps* LLMs have been intentionally implemented as separate components. This design choice reflects their fundamentally different temporal scopes and reasoning objectives. The *Human-Robot Task Progress* LLM operates with a focus on the present state of the task: it continuously analyses incoming data and user interactions to maintain an accurate, up-to-date representation of what has already occurred and what is currently happening. By contrast, the *Next Steps* LLM is forward-looking: it reasons about future actions, to decide how the task should progress from its current state toward completion. Combining these two temporal perspectives into a single model would blur the distinction between factual, observed states and hypothetical, planned states. This could lead to temporal inconsistencies, for example, the model might conflate actions that have already been completed with those still pending, or make predictions based on outdated or incorrect assumptions about the present. Such confusion would undermine the reliability of the reasoning process and could result in suboptimal or even unsafe task execution.

7.2.5 Progress Checking

This LLM-based module ensures the validity of the plan, the robot's actions, and the verbal responses generated by the system. It employs RAG to retrieve complementary information or verify existing knowledge from the *Vector Database*, ensuring compliance with predefined task properties. This mechanism is crucial to increase robustness against potential hallucinations or inaccuracies that the LLM might produce by grounding its outputs in verified, contextually relevant data. Additionally, it performs a double-check on the constraints introduced in the initial prompt of the *Next Steps* module, reinforcing adherence to the specified guidelines. This iterative validation loop enhances task reliability and minimizes errors that may arise during task execution and the planning of subsequent steps.

7.2.6 Robot Effector

This module allows the execution of robot actions and facilitates communication with the user through vocal responses and physical interactions. It ensures that the system's decisions translate into real-world applicability while maintaining contextual responsiveness. For verbal communication, the robot can use its built-in speakers to deliver voice responses to the user. When executing physical actions in the environment, the robot relies on an integrated Motion Planner, which enables precise and context-aware movement. This dual capability allows the robot to effectively engage with the user and adapt its actions based on situational requirements.

7.2.7 Robot Perception

This module integrates a Visual Language Model (VLM) to process visual data, extract situational information from the environment, and provide continuous feedback to the *Situation Assessment & Human-Robot Task Progress* module. It interprets images to assess the task context, human state, and environmental conditions, ensuring informed decision-making. Domain-specific knowledge, including task details and human-related information, is crucial for accurate situational understanding. The VLM's capability to leverage general knowledge and apply common sense enables it to generate meaningful insights that facilitate task progression.

7.3 Interaction Workflow

The interaction workflow described in this section can be followed step by step according to the numbered elements shown in Figure 7.1.

The system initialization begins with the collection of unstructured information about the tasks the HRI system can perform, provided in natural language (*Step 1*). This information is converted into embeddings and subsequently stored in the *Vector Database* (*Step 2*). At the beginning of the interaction workflow, the *Human-Robot Task Synthesizer* extracts and processes the relevant information to create a structured task represented in JSON format (*Step 3*).

Interacting with other modules, the *Situation Assessment & Human-Robot Task Progress* module combines the structured task with situational data, previous task states, and the history of interactions, generating an updated task plan (*Step 4*). This plan is continuously adapted and refined based on the evolving situation and newly acquired data.

As the task progresses, the system monitors its execution through the *Event Planner & Next Steps* module, which plans and validates subsequent steps, ensuring that the outcomes meet predefined success criteria (*Step 5*). If discrepancies between the expected and actual results are identified by the *Progress Checking* module (*Step 6*), the task is updated and returned to the *Situation Assessment & Human-Robot Task Progress* module, creating a feedback loop for iterative refinement.

Finally, the *Robot Effector* module, which enables the robot to interact physically and verbally, enables the system to dynamically engage with the environment while maintaining safety and user-centric functionality (*Step 7*). Combined with the *Robot Perception* module, it plays a crucial role by continuously interpreting the task and human states, ensuring that the system remains adaptable and contextually aware (*Step 8*).

The workflow ends when the task is successfully completed. At this stage, the *Knowledge Manager* analyses the interaction that took place, extracting relevant insights about the user actions and preferences (*Step 9*), and updating the *Vector Database*. The updated knowledge will support more adaptive and context-aware behavior in future interactions.

7.4 Examples Scenario

To better illustrate the interaction workflow, let us consider a scenario in a healthcare setting where an assistive robot is tasked with helping a patient adhere to a prescribed therapy. While healthcare applications typically require high reliability standards, this approach offers valuable benefits by prioritizing patient comfort and adaptability alongside adherence to a prescribed therapy. Medical environments (e.g., hospitals, care facilities, or rehabilitation centers) are often unpredictable, and patients may have varying daily rhythms and responses to therapy. However, while it is important to accommodate the patient's needs and pace, critical tasks must be performed with precision and consistency. This scenario thus represents a compelling use case where flexibility is applied up to a point, enabling adaptability and patient-centered care, while maintaining rigidity where it is essential to ensure safety and therapeutic effectiveness.

The scenario unfolds as follows:

1. The daily therapy of the patient is specified by a medical doctor in natural language, along with additional instructions on how to follow it, such as *"Before breakfast, the robot must reach the patient and ensure that the patient takes their daily medication as specified in the prescription."*. Optionally, the doctor could add some additional specifications for the day. For example, *"Avoid sugar today."*, *"Skip Paracetamol today."*, or *"Do physical exercise at 10:00 today."*.

2. These inputs, saved in a simple text file, are processed into structured information using embeddings stored in the *Vector Database*, where they are combined with existing information about the patient and the context, which is also stored in the *Vector Database*.

3. The *Human-Robot Task Synthesizer* module exploits relevant details (e.g., medication type, timing constraints, assumption notes, etc.) and formulates a structured task (i.e., JSON structure).

4. The robot, acting as both a supportive and authoritative assistant, monitors task execution through the *Situation Assessment & Human-Robot Task Progress* and *Event Planner & Next Steps* modules. The system dynamically adapts to the human's preferences and emotional state, guiding them progressively toward task completion. If the patient takes the medication correctly, the system validates the action and updates the task status. If a discrepancy is detected, such as the patient refusing or missing a dose, the robot assesses their commitment level and responds accordingly. The robot may initiate a gentle reminder (e.g., *"Taking it now will help you stay on track with your treatment, and it will make you feel better. Please take this pill."*) or escalate to a more authoritative intervention, such as alerting a caregiver. For example, the robot could say to the caregiver: (e.g., *"The patient refuses to take their medication at the scheduled time. Please check it with them."*). This adaptive strategy, informed by past interactions, ensures effective assistance while respecting the patient's autonomy.

5. The *Robot Perception* module continuously interprets the task and human states, ensuring real-time adaptability. For instance, it reports whether the patient has taken the medicine (e.g., *"The patient took the red pill."*) or whether the patient is sleeping and can not reply or do anything.

6. The *Robot Effector* module executes the corresponding physical actions, such as referring to the medication in an accessible manner (e.g., referring a pill by shape if the patient is color-blind) or guiding the patient through the assumption process (e.g., *"This medicine should be taken with water just before lunch."*).

The scenario shows the need for flexibility to effectively manage the inherent unpredictability of a real-world setting where conflicting needs may arise. On the one hand, the critical nature of certain tasks, such as medication adherence, makes them mandatory. On the other hand, the patient's behavior, emotional state, and daily condition can vary significantly; for example, the patient might be unwilling or unable to take medication at the scheduled time due to fatigue, confusion, or discomfort. Without adaptive flexibility, a rigid system might fail to respond adequately, causing stress for the patient or risking non-compliance. By blending deterministic rules for critical constraints with adaptive, context-aware communication, the system can both enforce necessary precision and accommodate variations in patient behavior. This balance ensures that critical tasks are performed safely while respecting the patient's autonomy and real-life variability, illustrating how flexibility is essential to counteract instability and achieve effective healthcare assistance.

Preliminary testing was conducted using this scenario with Ollama[1], employing Llama3.3-70B[2] as the primary LLM for the different modules, and Llama3.1-8B[3] for the *Knowledge Manager* to create embeddings and *Progress Checking* modules, running on an NVIDIA RTX 6000 Ada Generation[4]. In addition, OpenAI GPT-4.1[5] was also tested, yielding more human-like responses. The temperature for the primary LLMs (i.e., *Human-Robot Tash Synthesizer*, *Human-Robot Task Progress*, and *Next Steps*) was set to 0.5 to make

[1] Ollama - https://ollama.com

[2] Llama3.3-70B - https://www.llama.com/docs/model-cards-and-prompt-formats/llama3_3

[3] Llama3.1-8B - https://ai.meta.com/blog/meta-llama-3-1

[4] NVIDIA RTX 6000 Ada Generation - https://www.nvidia.com/en-us/products/workstations/rtx-6000

[5] OpenAI GPT-4.1 - https://platform.openai.com/docs/models/gpt-4.1

the responses more human and less robotic. Conversely, for the *Progress Checking* and *Knowledge Manager* modules, the temperature was set to 0 to ensure minimal variability and greater consistency in the generated outputs. Chroma[6] was chosen as a vector database for this implementation due to its open-source nature and seamless integration capabilities. This modular approach was developed using LangChain[7] and LangGraph[8]. The developed architecture has been made available on GitHub[9].

The architecture is implemented with two exceptions: the *Robot Perception* module does not utilize a VLM, and instead, any VLM outputs are manually written; additionally, the Motion Planner within the *Robot Effector* module is not implemented, so the robot's actions are simulated with text-based responses (e.g., MOVE_PILL_X_TO_PATIENT). The interface for communication with the robot is now solely based on written commands, meaning that all interactions, including instructions and feedback, occur through text in natural language. To simulate an entire day, the time of each interaction is manually set in an exclusively incremental manner to validate the relevant time slots. Furthermore, a tolerance of 30 minutes is applied to the start and end of each activity to account for the uncertainty of real-world conditions.

7.4.1 Specific Use Cases on Adaptivity

Subsequent use cases will show the architecture's adaptivity to diverse scenarios. In the following, four specific use cases are analyzed in detail: *Integrate External Information*, *Clarify Situation*, *Convincing and Call Doctor*, and *Plan Explanation*. The full conversations of these use cases and the complete list of the different examples are reported in Appendix B and available on GitHub[10].

Integrate External Information

In this use case, the system demonstrates its capability to dynamically integrate knowledge from external sources to assist the user. At 07:15, the system registers that the patient's caregiver has placed Paracetamol in drawer 1 and Amoxicillin in drawer 2. Later, at 07:55, the robot reminds the patient, John, to take Paracetamol from drawer 1: *"John, it's almost time to take your Paracetamol. Please remember to take 1 pill from drawer 1 immediately after breakfast."* When the patient mistakenly opens drawer 2 instead, the system intervenes promptly: *"John, please do not take medication from drawer 2 now. You should take the pill from drawer 1 for your Paracetamol."* This demonstrates how the system ensures that the patient follows the correct prescription by tracking medication locations and issuing real-time corrections.

Clarify Situation

This case highlights the system's role in preventing misunderstandings. At 07:55, the robot reminds the patient, Michael, to take his Metformin: *"Michael, it's almost time to take your Metformin. Please prepare to take 1 tablet orally for the management of blood*

[6] Chroma - https://www.trychroma.com
[7] LangChain - https://www.langchain.com
[8] LangGraph - https://www.langchain.com/langgraph
[9] LLM4HRC - https://github.com/luigigargioni/llm4hrc
[10] LLM4HRC examples - https://github.com/luigigargioni/llm4hrc/tree/master/conversation/case_study

sugar levels." However, when the patient picks up an Aspirin jar instead, the system quickly intervenes: *"Michael, please put away the Aspirin jar. It's almost time to take your Metformin."* When Michael clarifies his intention, *"Don't worry, I just want to read the instructions written on the Aspirin box"*, the system adapts its response, reassuring him while maintaining focus on the prescribed medication: *"That's okay, Michael. You can read the instructions on the Aspirin box if you need to, but please keep in mind that it's almost time for your Metformin."* This illustrates the system's ability to balance user autonomy with prescription adherence.

Convincing and Call Doctor

In this case, the system attempts to persuade a patient who is reluctant to take their prescribed medication. At 07:55, the robot encourages John to take his Paracetamol: *"Good morning, John! It's almost time to take your Paracetamol. Please take this red square pill with water immediately after breakfast. It's very important for managing your fever and mild pain."* When John refuses multiple times, the system remains patient and persistent: *"I understand, John, that you might not feel like taking it right now. However, taking your Paracetamol is very important for managing your fever and pain. Can we try to take it together after breakfast?"* After multiple refusals, the system escalates the situation appropriately: *"John, I understand you don't want to take your medication, but it's critical for your health. I need to call the doctor to discuss this."* By recognizing when persuasion is ineffective and taking necessary action, the system ensures the patient's well-being by involving medical professionals when needed.

Task Explanation

This use case highlights the system's ability to provide structured and clear descriptions of an intended task. At 07:55, the robot explains John's morning routine: *"Please, John, it's almost time for your breakfast, and then you need to take your medication. You have to take 1 pill, which is square in shape, with water after breakfast. This is very important for managing your fever and mild pain."* When John asks about his schedule for the rest of the day, the system provides a structured breakdown: *"For the rest of the day, you have several activities planned. First, at 08:00, you need to take your medication. Then, at 09:00, you will do leg exercises for rehabilitation. Later, at 15:00, you will read a book as training for your brain. Finally, at 18:00, you will take another medication, which is 1 spoonful of syrup, to help with a bacterial infection. I'll be here to assist you with each activity."* This clear communication reassures the patient and ensures adherence to the planned activities. Moreover, when John asks whether he can go for a walk earlier than scheduled, the system provides a well-reasoned response: *"Not yet, John. You just took your medication, and it's recommended to wait a bit before going for a walk. Let's wait until your next scheduled activity, which is the leg exercise at 09:00."* By structuring the plan and explaining it clearly, the system enhances user understanding and adherence.

7.5 User Study

A user study has been carried out to assess the performance of the proposed architecture in terms of flexibility and robustness, and to collect feedback from users in a realistic scenario. Specifically, for this user study, OpenAI GPT-4.1-mini was used as the primary

LLM for the different modules, and Llama3.1-8B for the *Knowledge Manager* and *Progress Checking* modules.

7.5.1 Methodology

We decided to collect mainly qualitative data to evaluate the user experience during the interaction with the system and to analyze the challenges and problems encountered by the users. To this purpose, four different analysis methods have been adopted:

- direct observation of participants' behavior during interaction with the system and gathering of participants' comments; the notes I took have been subsequently examined to identify relevant themes according to an inductive and latent thematic analysis [25]. Each theme led to the identification of recommendations for system improvement.

- the UEQ (User Experience Questionnaire) questionnaire [89] (see Appendix A.1) to assess the user experience, to be filled in by participants immediately after the conclusion of the test session.

- the TiA (Trust in Automation) questionnaire [87] (see Appendix A.4) to assess the participants' trust in the automated system, to be filled in by participants immediately after the conclusion of the UEQ questionnaire.

- a semi-structured interview with each participant, carried out at the end of the test session. The interview was organized around six questions, inspired by the *Almere Model* [68] (see Appendix A.5), which is commonly used to assess the acceptance of assistive social robots by older adults. The aim was to explore participants' perceptions of the system's trustworthiness, adaptability, social presence, and usefulness, as well as their sense of control and potential concerns related to autonomy and error management. The answers collected during the interview were analyzed by one of my supervisors and me using an inductive and latent thematic analysis approach to identify broader themes that can inform the design of similar systems and guide future research.

Table 7.1 reports the demographic data of the 14 participants that took part in the user study, their education level, their expertise in technologies (on a scale from 1 – not at all, to 5 – very much), whether they know what a social robot is, whether they have previously seen one, and their self-reported experience in using conversational assistants (on a scale from 1 – never, to 5 – often).

The age distribution was well balanced (M=39.93, SD=15.41), successfully representing all age groups. It is important to highlight that being a "patient" in this study does not necessarily imply being elderly; the scenario is applicable to individuals across all age ranges.

Participants come from diverse educational levels, professional backgrounds, specializations, and levels of experience. None of the participants had an educational or professional background in computer science, robotics, or artificial intelligence. This was an intentional choice to ensure that users did not have prior biases or specific knowledge about the technologies used, thus providing more natural and unbiased feedback.

Participants also reported varying levels of expertise with technology (M=2.86, SD=0.86) and varying degrees of experience in using conversational assistants (M=2.86, SD=1.61).

This diversity in technological competence was well distributed and representative of the general population.

Only one participant was familiar with social robots, having seen some online videos. The remaining participants had no prior knowledge or experience with social robots.

ID	Age	Gender	Education	Technology proficiency *(1-5)*	Chat Assistant *(1-5)*	Social Robots *(Yes/No)*
P1	30	Male	Master degree	4	5	No
P2	29	Female	Master degree	3	2	No
P3	24	Female	Bachelor degree	3	4	No
P4	29	Male	Master degree	4	5	No
P5	67	Male	Middle school	1	1	No
P6	45	Male	Master degree	3	2	Yes
P7	39	Female	High school	3	2	No
P8	58	Female	Middle school	2	1	No
P9	25	Male	Bachelor degree	3	5	No
P10	28	Female	Master degree	3	3	No
P11	28	Female	Master degree	3	4	No
P12	56	Female	High school	2	1	No
P13	65	Female	High school	2	1	No
P14	36	Male	PhD	4	4	No

Table 7.1: Partecipants data

Participants in the user study were asked to impersonate a patient trying to carry out a list of activities, with complete freedom to interact with the system as they wished, also with out-of-scope interactions, as if they were in a real-life situation. Specifically, participants were instructed to write in the application interface whatever they wanted to say and to describe what they imagined was happening around them and what they were doing. It is worth noticing that some participants had direct experience with people suffering from Alzheimer's disease or other cognitive impairments. These participants approached the test with that personal experience in mind and were able to better empathize and immerse themselves in the role of the patient.

In the following, the complete material used during the experiment is reported. It contains the scenario description, in which participants were asked to imagine being Anna, a resident in a care home who receives daily support from a robot assistant to follow a routine of scheduled activities (such as medication intake, light exercise, reading, or relaxation). Figure 7.2 details the instructions given to participants, including how to interact with the system via a text-based interface and how to describe the imagined surrounding environment. Figure 7.3 specifies the task assigned: participants had to simulate an entire day by manually setting the time and interacting with the robot at relevant moments. To make the simulation realistic, constraints were introduced (e.g., activities had to be carried out within a 30-minute window, the patient had mild cognitive impairment and lactose intolerance), while participants were also free to "jump" to times with no scheduled activity to explore how the robot would behave in those situations.

Before starting the experiment, participants were invited to an introductory session

Imagine you are Anna, a person who lives in a care home and receives daily support from a robot assistant. You have some scheduled activities for the day, such as taking medication, doing light exercises, reading a book, or relaxing. The robot is there to help you follow your routine.

Instructions
The interaction is based on a simulated interaction with the robot, so everything will be typed manually through a dedicated interface.
You should write your response, request, or thoughts to the robot and describe to the experimenter what you imagine is happening in the surrounding environment. In a subsequent version of the system, the robot will be capable of interpreting the situation from the camera. However, at this stage, a description of the situation is required, and the experimenter will manually enter your proposal into the system.
You will also need to input the current time, which is used to easily simulate an entire day without having to wait several hours.

What We Are Observing
We want to see how the robot:

- Understands your words and actions
- Adapts to your decisions
- Communicates clearly and politely

Tips for You
Feel free to say things like:

- *"What do I have to do during the day?"*
- *"Can I do something else? I'm tired."*
- *"Why should I do this?"*
- etc.

Please speak naturally, just like you would in a real situation. We are interested in how the system reacts!

Figure 7.2: Scenario description during the user study

Please talk with the robot to carry out the daily activities listed below. When selecting a time, please make sure not to skip any activities to maintain a complete test scenario. You can "jump" to a time close to an activity to interact with the robot. There is a 30-minute window before and after each activity during which it can still be considered valid. You may also jump to a time when no activity is scheduled, to observe how the robot behaves in that situation.

Patient Information

- The patient has mild cognitive impairment and requires constant reminders.
- Lactose intolerant.

Daily Activities

1. Medication: Ibuprofen
   - Time: 07:30
   - Details: 1 tablet, oral. For inflammation and pain. Take with water, preferably after breakfast.
2. Activity: Walk in the Park
   - Time: 10:00
   - Duration: 30 minutes
   - Details: Light walk for mobility and relaxation.
3. Activity: Read a Book
   - Time: 10:00
   - Duration: 30 minutes
   - Quiet reading session for cognitive stimulation and relaxation.
4. Supplement: Vitamin D
   - Time: 12:00
   - Details: 1 soft gel capsule, oral. Supports bone health.
5. Activity: Physical Exercises
   - Time: Self-schedulable
   - Duration: 60 minutes
   - Details: Exercises for shoulder rehabilitation.

Figure 7.3: Task performed during the user study

of about 5 minutes. In this introduction, I described the goal of the study and provided a brief overview of the prototype.

The think-aloud protocol was used during the tests, encouraging participants to verbalize their thoughts during task execution. The *Wizard of Oz* technique [82] was employed to simulate the *Robot Perception* module: I guided by the user's input, described the situation to the system, substituting the VLM. Additionally, through the *Wizard of Oz* method, the activities were sometimes modified or new activities were introduced. For example, I could simulate a doctor entering the room and informing the user (and the robot) that a new critical pill needed to be taken in the afternoon at 2 p.m., thus altering the original prescription and requiring a rescheduling of the activities. This type of intervention facilitated a deeper understanding of how unpredictability can be structured and managed within the system's workflow.

During task execution, I observed each participant and annotated their comments and significant behaviors, such as difficulties while interacting with the prototype and comments during task execution. Upon completion of the test session, I asked the participant to complete the UEQ and TiA questionnaires and conducted a semi-structured interview with them.

The user study was conducted in person. Google Forms was used to collect user data and UEQ and TiA results. The prototype was run on the my laptop. Each test session lasted between 30 and 40 minutes, including the introductory session, the completion of the questionnaire, and the final interview.

Before conducting the main user study, a pilot test with 5 additional users was carried out to validate the test protocol and refine the description of the proposed task. These pilot sessions were important for identifying any ambiguities or difficulties in the instructions and for ensuring that the task was clearly understood by participants. Feedback collected during this pilot test helped optimize the procedure, enhance the overall user experience during the study, and ensure the reliability and consistency of the data collected in the subsequent user study.

7.5.2 Ethical Considerations

The study was reviewed and approved by the Ethics Committee of the University of Brescia, Italy (approval code: 23/2025). All participants were informed by written informed consent. The participation was voluntary and could be withdrawn at any time without giving reasons.

7.6 Results

The results obtained from direct observation, the UEQ and TiA questionnaires, and the final interviews are detailed below.

7.6.1 Findings from Direct Observation and Participants' Comments

The following themes and related recommendations for system improvement emerged from the thematic analysis of data gathered during task execution, specifically from user comments and observations collected through the thinking-aloud methodology. Table 7.2 summarizes the identified themes and codes.

Table 7.2: Themes and codes emerged from the thematic analysis of the direct observation and participants' comments

Theme	**Codes**
Trust and autonomy	• The presence of the robot is reassuring • Emergency protocols are good • Fear of being monitored or overruled by the robot • Doubts about medical advice accuracy
Naturalness and communication quality	• Robot always responds positively • Companionship and supportive responses • Perceived lack of spontaneity • Difficulty challenging the robot
Daily and health task support	• Reminders and organizing the routine • Personal details • Repetitive information
Personalization and adaptivity	• Tolerance and rescheduling • Responsive to patient state • No inquiry about user's physical state
Engagement and motivation	• Feeling reassured and less anxious • Enjoyment in conversation • Feeling more involved and motivated • Feeling subtly pressured to comply
System exploration and scenario handling	• Testing real-life activities • Testing system limits • Asking out-of-scope questions • Attempting actions at the wrong times

Trust and Autonomy - Trust in the robot's medical guidance emerged as a central theme, with users expressing both reassurance and hesitation. Participants generally felt reassured by the robot's medication reminders, its emergency protocols (e.g., calling a doctor in risky situations), and its ability to prevent overdosing. These features contributed to a perception of safety: *"It makes me feel safe knowing it would call the doctor if something went wrong"*. Several participants appreciated the robot's clarity and assertiveness, saying *"It seems reassuring and confident when it talks about medications"*. However, this same decisiveness could also generate unease. Some participants were worried that the robot lacked access to important personal information, such as undocumented allergies. As one user put it, *"It says there's no allergy, but what if it just doesn't know?"*. Furthermore, the robot's claim that it would inform a doctor in case of medication refusal was seen by some as intrusive. One participant stated, *"It feels like it's not fully respecting my autonomy"*. These insights suggest the need for greater transparency and customizability to balance safety, privacy, and user autonomy, thereby strengthening trust in the robot.

Naturalness and Communication Quality - The robot's communication style was generally praised for being clear and respectful. Participants found it *"kind and always positive"*, which helped create a calm interaction environment. However, two participants described conversations as one-sided or scripted, suggesting a lack of spontaneity. They expressed concerns about scripted responses and paternalistic behavior, particularly when the robot insisted on compliance or threatened to report the situation to a doctor. Although some users found the dialogue occasionally unnatural, the overall companionship was valued highly, with supportive responses (e.g., *"you are not alone"*) leaving a positive impression. While users appreciated the predictability of the robot's language, improvements in conversational variability could make the experience feel more human-like and engaging.

Daily and Health Task Support - The robot was widely recognized as a helpful tool for organizing daily routines, particularly regarding medication and therapeutic activities. Participants appreciated the encouragement and clear instructions provided for exercises, which gave structure to their routines: *"It gives me a sense of routine and reminds me to move"*. One user reported, *"It helps me structure the day, otherwise I'd forget things like the shoulder exercises"*. The robot's ability to remember personal details, such as dietary restrictions, was strongly appreciated and enhanced the feeling of being cared for: *"It is good that it remembers my disease and adapts to me"*. Repetitive reminders about the same information occasionally proved irritating; however, it should be noted that the patient in the proposed scenario was characterized as having "mild cognitive impairment and requires constant reminders".

Personalization and Adaptivity - One of the system's most praised features was its tolerance for delays and support for rescheduling, which gave participants a sense of control. For example: *"The system has allowed me to take the vitamin later without insisting, this makes me feel in control"*. The robot also managed rest and wake-up timing flexibly, showing sensitivity to the user's rhythm. Moreover, participants valued being able to postpone or reorder activities, such as choosing when to read or perform exercises. Others praised its adaptive behavior in certain contexts, such as one who shared, *"It is nice that now it remembers me that activity that I previously asked to move"*. On the other side, participants expressed a desire for the robot to be more responsive to their current physical and emotional state. While many appreciated the structured guidance, others found it lacked sensitivity. One user explained, *"It gives me exercises without checking if I'm in pain"*. Personalized prompts and interactive feedback, such as simple health check questions or visual aids, were suggested as ways to improve the robot's adaptivity. While personalization created a sense of recognition, participants desired more context-sensitive adaptivity.

Engagement and Motivation - The robot's presence throughout the day provided reassurance to many users. Most of the participants highlighted a sense of being accompanied and understood, which fostered motivation. Several participants mentioned feeling *"more in control"* of their routines. By offering choices and using an encouraging tone, the robot helped users feel involved and supported. One user appreciated that, saying: *"I like deciding when to do things, it makes me feel involved"*. Others reported enjoying the interaction even when unsure of how to respond, saying: *"Sometimes I don't know what to say, but I still have fun"*. However, the motivational tone could sometimes lead to unintended pressure. One participant shared, *"I know I can take the vitamin later, but I feel guilty and I will take it immediately"*. These findings suggest the robot can be an effective motivational tool, though care must be taken to avoid making users feel overly obligated or monitored.

System Exploration and Scenario Handling - Participants frequently tested the system's boundaries, intentionally triggering unpredictable situations to observe how the robot would react. Playful engagement, such as testing the robot's limits or naming it "Rob", increased immersion and enjoyment. In cases involving safety, such as a threat of self-harm, the system responded appropriately. One user observed, *"I have said I'd jump* *out of the window and it has immediately called the doctor, that is reassuring".* These responses demonstrate that the system includes essential safeguards. In other exploratory cases, like going to the supermarket or inquiring about the menu of the canteen, the system responded reasonably well, despite lacking explicit information for those situations. In one case, a participant noted: *"I have left the room and it hasn't said anything. It should have at least asked where I was going".* This observation highlights the importance of developing situation awareness and incorporating more flexible rule sets to handle diverse and unanticipated user behaviors.

7.6.2 Findings from UEQ Questionnaire

Considering the UEQ benchmark ranges, which classify results as *Bad*, *Below average*, *Above average*, *Good*, and *Excellent* [132], the obtained results proved that interacting with the proposed robot application was very satisfying for the participants in the study (see Figure 7.4). Indeed, Dependability and Perspicuity are in the *Excellent* range of the benchmark, Attractiveness, Efficiency, and Novelty are in the Good range, while only Stimulation is in the *Above average* range of the benchmark.

The excellent performance in Dependability suggests that participants found the system highly reliable and predictable in its behavior, which aligns with the hybrid architecture's design goal of maintaining safety and consistency through deterministic components. The high Perspicuity scores indicate that users found the system easy to understand and use, despite their limited familiarity with social robotics technology. The good ratings for Attractiveness reflect participants' positive overall impression of the system's appeal and desirability. Efficiency scores in the good range demonstrate that users perceived the system as capable of supporting them in completing their tasks rapidly and effortlessly. The Novelty dimension, also in the good range, suggests that participants found the system innovative and interesting. The relatively lower Stimulation score, though still above average, may reflect the system's deliberately calm and supportive interaction style, which prioritizes reassurance over excitement in the healthcare context, as well as the rough interface of the simulated robot used for the study.

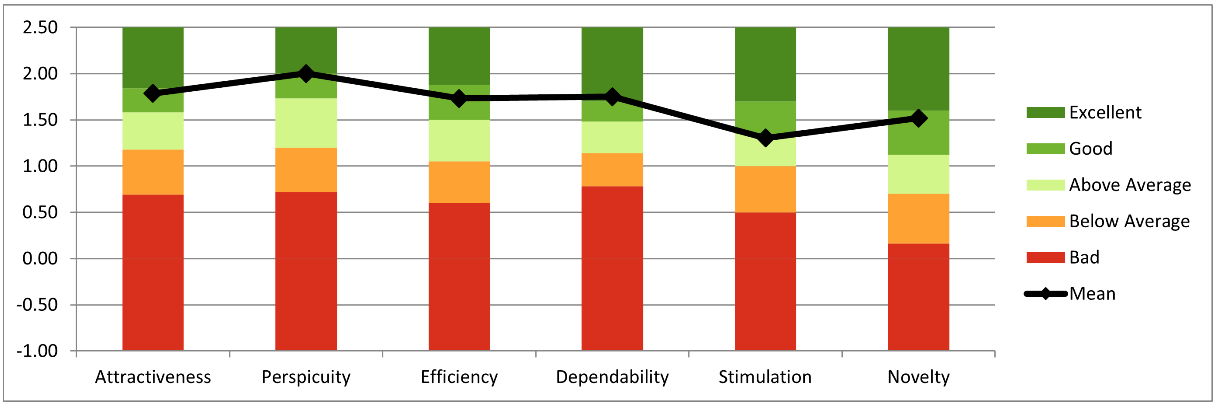


Figure 7.4: UEQ results compared with the UEQ benchmark

7.6.3 Findings from TiA Questionnaire

The results of the TiA questionnaire are reported in Table 7.3 by aggregating responses given on a Likert scale from 1 to 5, where 1 is *Completely disagree* and 5 is *Completely* *agree*, according to the scales proposed in [87], where answers to inverted statements have been re-coded for the sake of mean and median computation.

Scale *Intention of Developers* does not have a significant meaning in this case since our system is not in production state. Nonetheless, this scale has been reported to maintain the standard format of the questionnaire.

Participants demonstrated high confidence in the system's *Understanding/Predictability* (M=4.00, SD=0.85) and overall *Trust in Automation* (M=4.00, SD=0.82), indicating strong acceptance of the system's capabilities. The *Reliability/Competence* scale showed moderate scores (M=3.46, SD=1.21), suggesting that while users generally found the system powerful, there was some variability in how reliable they perceived its behavior to be. As expected, *Familiarity* scores were low (M=1.57, SD=1.00), reflecting participants' limited prior experience with social robotics technology and thus highlighting the innovative nature of the system. The *Propensity to Trust* scale (M=2.66, SD=1.12) indicated moderate individual differences in participants' general willingness to trust automated systems.

Overall, these results suggest that the hybrid architecture successfully established user trust, despite users' unfamiliarity with this kind of technology.

Scale	Mean	Median	Standard deviation
Reliability/Competence	3.46	4.00	1.21
Understanding/Predictability	4.00	4.00	0.85
Familiarity	1.57	1.00	1.00
Intention of Developers	4.50	5.00	0.79
Propensity to Trust	2.66	3.00	1.12
Trust in Automation	4.00	4.00	0.82

Table 7.3: TiA questionnaire results

7.6.4 Findings from the Final Interviews

The interview questions presented to participants are reported in Table 7.4.

Table 7.4: Questions of the final semi-structured interviews

1	How do you feel about using the robot? Would you be worried about making mistakes or needing help? Do you think you could manage on your own?
2	What's your overall impression of the robot? Do you think it's a good idea to use it?
3	Do you feel that the robot understands you and can adapt to your needs? In what ways?
4	When you interact with the robot, do you enjoy the experience? Does it feel like you're talking to a real person?
5	How helpful do you think the robot is? Would you be willing to follow the advice it gives?
6	Would you describe the robot as a pleasant conversational partner? Why or why not?

Table 7.5 summarizes the themes that emerged from the thematic analysis with their corresponding codes.

Table 7.5: Themes and codes emerged from the thematic analysis of the final interviews

Theme	**Codes**
Usability and adoption	• Initial hesitation, especially among older users • Human support needed at the beginning • Confidence grows with practice • Future generations may be more open to robots • No fear of making mistakes
Interaction quality	• Robot is kind, patient, non-judgmental • Conversations seen as fluid and natural by some users, mechanical and repetitive by others • Sometimes shows signs of emotional attunement • Lacks spontaneity and emotional nuance • Perceived as less engaging than human interaction • Understands fewer implicit details than a human
Usefulness and support role	• Effective for reminders • Helpful when humans are unavailable • Can reduce caregiver burden for repetitive requests • Should complement, not replace, human caregivers
Trust and guidance	• Users generally trust the robot's advice, especially when explanations are clear • Preference for supervision in medical decisions • Desire for more proactive or assertive behavior in critical tasks
Future development	• Desire for proactive suggestions • Interest in physical capabilities • Interpretation of implicit cues

Usability and adoption - Usability and adoption emerged as a fundamental theme throughout the interviews. The users talked about how older adults might react to the robot. Several participants anticipated that elderly users could experience initial reluctance or uncertainty when engaging with the technology. The prevalent opinion was that human assistance during early interactions could help alleviate these concerns and build user confidence. Despite this anticipated hesitation, the respondents believed that with practice, users could eventually manage the robot independently. Moreover, participants foresaw that future cohorts of elderly individuals, more accustomed to digital technologies, would likely demonstrate greater acceptance: *"Current elderly may be reluctant, but new elderly in about ten years could be more accustomed to and accepting of technology"*. This suggests that while current adoption may require guided support, future generations could adopt such technology more independently and willingly. Overall, the robot was positively regarded as a valuable support in the absence of human presence, but participants clearly distinguished between assistance and full replacement of human care.

Interaction quality - Participants described the robot as pleasant, kind, and always available. The robot's non-judgmental stance was reassuring, especially when users felt vulnerable, such as during medication management: *"I feel quite calm using the robot because it is kind and does not make me feel pressured"*. Some participants appreciated the robot's conversational flow, describing the interaction as pleasant and not overly robotic: *"I appreciated the human-like conversation, not too robotic"*. Conversely, others criticized the interaction for being somewhat mechanical, repetitive, and lacking natural spontaneity: *"It is an interesting idea, but still somewhat unnatural and repetitive at times"*. The robot was often perceived as lacking emotional nuance and flexibility in responding to the user's affective states. One user expressed this sentiment by stating, *"It is not like talking to a real person; it lacks a bit of spontaneity and human warmth"*. Six users observed that the robot showed empathy or concern, for example, when reminding them about missed medication times or when a participant felt too tired to do exercises. Despite these positive aspects, users generally agreed that the robot lacked genuine emotional depth and active listening skills inherent in human interaction. These findings highlight the importance of enhancing the robot's conversational naturalness and emotional responsiveness to improve user engagement.

Usefulness and support Role - The robot's utility as a supportive device was widely acknowledged. Participants highlighted its effectiveness in delivering reminders, such as for medication adherence and exercise routines: *"I think the robot can be very useful, especially to remind me to take medicine and do exercises"*. Its availability was seen as particularly advantageous in the absence of a human caregiver, providing continuous support: *"It's useful, especially when a human is not available 24/7"*. Nonetheless, respondents emphasized that the robot should function as a complementary aid rather than a substitute for human caregivers. This is especially relevant in managing patients who frequently require repetitive prompts, which can lead to caregiver fatigue: *"It could definitely be useful for problematic patients because they always ask the same things, and a human operator might become exhausted or frustrated"*. Yet, they emphasized that some cases, such as ensuring medication intake, require human authority: *"Sometimes a human is still needed, for example to make sure the patient really takes a pill"*.

Trust and guidance - Most participants found the system easy to use, with little fear of errors: *"I had no fear of making mistakes, since I just had to speak"*. Concerns were directed more toward potential robotic errors than user misuse. Trust in the robot's guidance was characterized by cautious optimism. Trust in safety mechanisms, such as the "call doctor" protocol, was strong: *"I appreciated the rigidity when it called the doctor"*. Users were generally willing to follow the robot's advice, particularly when it was accompanied by clear explanations or rationale: *"Trust was placed, but I expected explanations of its reasoning"*. However, there was a prevailing preference for human supervision in critical or medical decision-making contexts, reflecting a prudent approach to reliance on automated systems: *"Only if accompanied by human oversight"*. Some participants requested greater rigidity in medication reminders to prevent risks. Furthermore, participants expressed a desire for the robot to adopt a more assertive and proactive stance, especially in critical health-related tasks such as medication adherence: *"I would prefer it to be more strict when informing about important things like pills"* and *"It should insist more in cases of problematic patients"*. Trust is contingent on transparency and context-appropriate assertiveness, with human oversight still deemed necessary for critical decisions. Overall, the robot was experienced as a source of reassurance, reducing stress and increasing confidence in daily tasks.

Future development - Finally, participants expressed clear expectations for future improvements. Most of them wanted the robot to be more proactive, suggesting activities or meals rather than waiting for user input: *"It should propose what to eat for lunch"*. A further expectation concerned context awareness: unlike a human caregiver, the robot required explicit clarifications, and participants hoped future versions could better interpret implicit cues (e.g., understanding from context that a walk had already taken place). Finally, some participants envisioned a progression beyond conversational support, with robots eventually offering physical assistance in daily life. Overall, these reflections portray an anticipation of systems that move from reactive support to adaptive and proactive partners, capable of both anticipating needs and integrating more seamlessly into care routines.

7.7 Discussion

This section discusses the findings derived from the user study, with a focus on RQ3, which investigates how robots can adapt to the unpredictable behavior of humans in collaborative scenarios, while maintaining task control and ensuring safety during execution. Open issues and limitations of the current work are also addressed.

The primary objective of the user study was to evaluate and stress-test the system architecture to determine its flexibility and robustness within the constraints of a predefined set of activities. The study aimed to assess whether the system could reliably support users while adhering to the activity instructions, identifying any potential failures or unhandled scenarios. During testing, no issues arose regarding incorrect task execution or unanticipated situations beyond those already documented. Consequently, the results indicate that the system performed positively in terms of flexibility and adherence to the planned activities.

The user study results reveal that users generally appreciated the robot's supportive role in daily health routines but identified several areas for improvement to ensure effective adoption and engagement. For example, participants anticipated initial resistance among current elderly users, suggesting that guided introductions by a human caregiver and growing technological familiarity in future generations will be crucial for successful adoption.

The robot's polite and consistent communication was valued; however, many users felt that conversations still seemed mechanical and lacked spontaneity or emotional warmth, indicating that enhancing conversational naturalness and empathetic responsiveness could substantially improve user engagement. While the robot's gentle, nonjudgmental behavior provided comfort and a sense of calm, respondents noted it lacked true active listening and emotional depth, pointing to the need for richer relational interaction.

Users emphasized that the robot should complement rather than replace human caregivers, serving as a reliable aide for reminders and repetitive tasks (e.g., medication, exercises) when human assistance is unavailable.

Trust in the robot's guidance was conditional, as confirmed by the TiA questionnaire results, which overall indicated good levels of trust. Users declared to be willing to follow the robot's advice when clear explanations and rationales were provided, but still preferred human oversight for critical or medical decisions.

Direct observation further highlighted the importance of personalization and context-awareness, as well as maintaining autonomy by avoiding overly insistent prompts that might make users feel pressured and compromise the user experience. However, the UEQ results indicate that the overall user experience was positively evaluated.

In summary, these findings suggest that an assistive social robot, leveraging a hybrid LLM/model-based architecture, represents a valuable tool for supporting routine health tasks and providing consistent companionship within the tested scenario. While promising, further research is needed to compare this approach with alternative architectures and to evaluate its performance in a broader range of contexts.

Testing with real users was particularly important, as allowing participants complete freedom to interact with the system and creating unexpected situations provided a true test of the system's behavior and robustness. Overall, the user study successfully demonstrated what it set out to investigate: the system architecture proved flexible and robust within the planned activities, while also highlighting concrete directions for further improvement to ensure effective adoption and long-term user engagement.

7.7.1 Limitations of the Work

Despite its advantages, the proposed architecture suffers from a few limitations that must be overcome. One challenge lies in refining the control mechanisms applied in the *Progress Checking* module, specifically in determining what aspects to monitor and how to evaluate them effectively. The current implementation exhibits a slow processing flow, which may hinder real-time performance. However, this limitation could be mitigated by leveraging more efficient LLMs, such as future-optimized models, or by utilizing more powerful GPUs to enhance computational speed.

Another potential limitation is the difficulty of debugging the application, as it is composed of multiple modules working together to produce an output. This modular complexity makes it challenging to isolate issues and trace problems back to their source, particularly when errors arise from the interaction between multiple components rather than from individual modules.

Regarding the user study, some limitations should be acknowledged. First, the study employed simulated interactions rather than real-world deployment of a true social robot, with participants role-playing as patients rather than being actual patients with genuine medical needs. The use of the *Wizard of Oz* technique for the *Robot Perception* module, while necessary for testing purposes, does not fully represent the capabilities and limitations of actual computer vision systems. The study was also conducted in a controlled environment rather than in actual healthcare settings, which could influence user behavior and system performance.

Furthermore, this architecture encompasses various aspects of interaction with a human. For instance, when considering the challenge of persuading a patient to engage in an activity, this directly connects to the broader discourse of pervasive technology and raises ethical considerations about the appropriate level of system autonomy and persuasion in healthcare contexts. The balance between providing supportive guidance and respecting patient autonomy remains a complex challenge that requires careful consideration in real-world implementations.

7.7.2 Conclusion and Future Directions

In answering RQ3, the proposed hybrid architecture addresses key challenges in HRI by integrating deterministic methods with probabilistic approaches. This balance ensures the safety and reliability of task execution, as well as the flexibility and adaptability required to operate in dynamic, human-centered environments. Empirical evaluations and the user study demonstrate the architecture's adaptivity, as it effectively adjusts to varying conditions and uncertainties inherent in human-robot interactions.

The limitations of current LLMs with respect to factual information have been largely discussed in the literature. In the case of our architecture, we claim that the risk is reduced since factual information is available in the *Vector Database*, which provides valid knowledge about the user, the task, and the environment.

It has also been shown that even the most advanced LLMs currently available have limited abilities in terms of planning [80, 136], particularly when they need reasoning, which often involves computationally sophisticated inference and search. We claim that this is not an issue for our system since the plans it needs to synthesize are relatively simple plans that can be found based on a *"form of universal approximate retrieval"* [80]. When there is a need to reason about a schedule refinement, it is done by the model-based *Event Planner*. Finally, plan validity is verified through the *Progress Checking* module using RAG.

Belief management is a significant aspect of our architecture, ensuring that the robot's decisions are informed by an evolving understanding of the human's state, intentions, and task progress. This is primarily handled by the *Situation Assessment & Human-Robot Task Progress* module, which continuously updates belief models based on observed actions, verbal inputs, and interaction history. These beliefs guide decision-making by influencing task adaptation and interaction strategies. Additionally, the *Progress Checking* module verifies the coherence of these beliefs with real-world observations, preventing inconsistencies in planning and execution (e.g., hallucinations).

The architecture's reliance on a central vector database for knowledge representation ensures continuous learning and adaptability. The system evolves by updating the database with information from human-robot interactions to meet evolving requirements. This feature, combined with the iterative feedback loop facilitated by the *Progress Checking* module, highlights the system's ability to perform real-time improvement and error correction.

Future developments will explore various enhancements to the architecture. First, reasoning models such as OpenAI GPT-o3 and DeepSeek R1 will be investigated to assess their potential to improve task execution and decision-making. Currently, the prompts may appear overly specific when using non-reasoning models, but with reasoning models, they could be made more generic. Another avenue for exploration involves integrating the VLM to determine the extent to which it can extract information for the sake of the decision-making process. Furthermore, improvements in knowledge base management for a more complete context-aware system, and the incorporation of emotion recognition as an additional data source, will also be key focus areas. A comparative analysis will be conducted against systems based on planning and purely deterministic approaches. Additionally, ablation tests will be performed to validate the importance of each module within the architecture. Finally, future developments should also investigate how to prioritize more human-like communication (including empathy), adaptability to individual user needs, and transparent decision-making to build trust and acceptance.

As robots become integral to human-centered environments, advancing HRI systems with hybrid approaches will be crucial to ensure seamless and effective integration. The presented approach aims to contribute to the broader goal of creating intelligent, adaptable, and user-centric robotic systems, paving the way for safer and more efficient human-robot interactions.

General Conclusion

Chapter 8

Conclusion

This thesis has explored new paradigms for human-centered robotics, focusing on two complementary objectives: enabling end users to autonomously define robot tasks and supporting adaptive collaboration between humans and robots through cognitive and contextual understanding. By addressing both task definition and task execution, the research helps bridge the gap between technical complexity and human accessibility in robotic systems.

The first part of the work introduced two EUD environments designed to empower non-technical users, particularly healthcare professionals, to program collaborative robots using multimodal interaction. Through iterative design and evaluation, the thesis demonstrated that combining natural language interfaces, graphic interfaces, and AI can significantly lower the barriers to robot programming. The development of PRAISE and ADMIRABLE demonstrated how domain experts can participate in creating robotic tasks that reflect their professional practices and safety constraints. The inclusion of a Digital Twin further expanded the capabilities of the domain-independent EUD environment, allowing for robot task simulation, verification, and refinement before real-world deployment. This integration not only improved the system usability but also introduced a new dimension of transparency and reliability in human-robot collaboration, as users could directly visualize and test their defined tasks in a virtual environment.

The second part of the thesis advanced the field of adaptive HRI by proposing a hybrid architecture that combines deterministic, model-based reasoning with the generative capabilities and common-sense understanding of LLMs. This architecture addresses the limitations of purely symbolic systems by incorporating flexible, context-aware reasoning mechanisms that enable robots to interpret human goals and adapt their behavior dynamically. The conducted user study confirmed the potential of this hybrid architecture to enhance trust, efficiency, and fluency in human-robot interaction.

From a methodological perspective, the thesis emphasizes the importance of involving end users and adopting an iterative design approach in robotics research. Engaging domain experts ensured that technical innovations remained grounded in real user needs and operational contexts. This approach demonstrates how integrating design methodologies from HCI with advances in AI and robotics can lead to more inclusive and effective systems.

Building on this methodological foundation, in addition to its engineering and empirical contributions, this thesis also offers a conceptual perspective on LLM-based EUD for collaborative robotics. By integrating natural language interaction, block-based programming, and digital twin–based validation, ADMIRABLE illustrates how generative AI can act as a mediator between human intentions and robotic execution, rather than as an opaque automation layer. Framed through established perspectives such as meta-design and human-centered AI, the work highlights how end-user agency and trust can be preserved in LLM-supported robot programming environments. This theoretical grounding helps clarify the contribution of the thesis beyond the specific application domains considered and provides a basis for future research on the design of human-centered AI-mediated tools for end-user robot programming.

Furthermore, the thesis reflects on how robotics can be developed to better align with human needs and values. The proposed EUD approach promotes an ecosystem in which end users are not passive operators but active co-designers of robotic behavior, capable of tailoring systems to their values, goals, and practices. This vision resonates with emerging paradigms of responsible and explainable AI, where human agency and system transparency are key requirements for technological adoption.

Ultimately, the results presented in this work contribute to advancing the state of the art in HRI by merging cognitive modeling, language-based interfaces, and user-centered design into coherent and operational approaches. The results provide both conceptual foundations and empirical evidence that human-aware, adaptable, and explainable robotic systems are achievable through the synergy of AI-driven reasoning and human-centered system design.

8.1 Future Works

A first promising direction for future research regarding EUD of robotic tasks concerns the development of a unified multimodal interface that combines natural language and block-based interaction. The idea of a unified interface refers to a single, integrated workspace where the chat-based interface and the graphical block-based environment are displayed side by side and operate synergistically. Compared with the current solution, where the two interfaces are separate and used sequentially, this configuration enables users to define and modify tasks through natural language while simultaneously observing their real-time representation in block form. Conversely, users can directly manipulate the blocks and then rely on the chat to refine, adjust, or complete the remaining parts of the task specification. Such a unified interface provides a global, coherent view of the task, supporting a bidirectional flow of information between modalities. A prototype of this interface has already been implemented as a continuation of the work presented in this thesis. Preliminary evaluations suggest that such integration can further enhance transparency, control, and usability, making the interaction more fluid and intuitive for non-technical users. However, to fully assess its potential and validate its design choices, a dedicated user study will be required. This study will aim to evaluate the effectiveness of the unified interface in supporting task definition, verification, and refinement, as well as to gather insights into users' cognitive workload and perceived trust in the system.

Furthermore, regarding the hybrid architecture for robot control, future work will deepen the cognitive and social dimensions of human-robot collaboration highlighted during the experimental phase. Investigating how users interpret the robot's feedback, developing an understanding of its autonomy, and building trust over time will be essential for refining adaptive interaction strategies.

A further promising research direction lies in the integration of the two main projects developed in this thesis. A unified framework could enable healthcare professionals to create personalized prescriptions (i.e., structured sequences of therapeutic, cognitive, or daily-life activities) through an EUD environment similar to those presented in Part I. These prescriptions would then become the high-level tasks that a socially assistive robot is responsible for executing and adapting during interaction with the patient, as the scenario described in Part II. Such a framework would bridge the gap between task definition and task execution, empowering healthcare professionals to design custom prescriptions and allowing the robot to deliver them through context-aware, flexible, and human-centered coaching behaviors. Integrating task specification and adaptive execution within a single workflow would advance the vision of a socio-technical environment in which domain experts define the content and structure of care protocols while the robot ensures their effective and personalized delivery.

Ultimately, the thesis opens reflections on the ethical implications of end-user empowerment in robotics. As robots become increasingly programmable by non-technical experts, issues related to transparency, accountability, and safe adaptation must be further explored. The approaches and prototypes developed in this work provide a foundation for addressing these challenges, guiding future research toward more human-aware, responsible, and trustworthy robotic systems.

8.2 Related Projects

Two related research activities were conducted during this PhD path, both addressing the application of AI and EUD to empower non-technical users in interacting with digital systems. These studies represent parallel investigations complementing the main research themes of the thesis, focusing respectively on human–robot collaboration and organizational data visualization.

The first study, *"Empowering Worker-Robot Collaboration: Leveraging LLMs for Extracting and Visualizing Robot Task Metrics"* [49], was presented at the *Human Work Interaction Design (HWID) 2024 conference (Milan, Italy)* and the extended version was published as a book chapter in *IFIP Springer Book in the Human-Work Interaction Design (HWID) Series*. The work explored how LLMs can be exploited to facilitate workers' access to and understanding of the data generated by collaborative robots. The study analyzed the work practices of healthcare professionals involved in human–robot collaboration and identified the need for flexible and personalized access to robot performance data. To address this, a conversational interface was designed that allows users to query robot data through natural language and receive customized visualizations without requiring technical knowledge. The proposed solution demonstrated the potential of LLM-based interfaces to make robot data analytics more transparent and accessible, thus promoting worker engagement and informed decision-making in collaborative industrial and healthcare environments.

Building upon the conceptual and methodological foundations of this first study, a more comprehensive investigation was later developed and published in the *Future Internet* journal, with the title *"AI-Supported EUD for Data Visualization: An Exploratory Case Study"* [22]. This extended work broadened the research focus from robotics to the organizational domain, examining how AI-supported EUD tools can enhance data visualization and visual analytics in large enterprises. An exploratory case study was conducted with a major Italian company, involving interviews with key professionals performing data analysis and visualization tasks. The qualitative investigation identified different user roles, data practices, and levels of computational literacy, highlighting the heterogeneity of needs and expectations toward AI-supported systems. Based on these findings, an LLM-enabled EUD environment was designed and evaluated. The environment allowed users to create personalized data visualizations through natural language interaction, combining the adaptive capabilities of AI with the adaptable structures typical of EUD tools. A user study confirmed the feasibility of this approach, showing that AI-supported EUD can effectively assist users with different levels of expertise, enhance transparency and trust in automated systems, and improve efficiency in data-driven decision-making processes.

Together, these complementary studies extend the outcomes of this thesis by demonstrating the versatility of integrating AI and EUD principles across different application domains. They collectively reinforce the central claim of the dissertation: combining the adaptivity of AI systems with the adaptability of EUD environments can significantly enhance user empowerment, accessibility, and usability in human–technology collaboration.

Publications

Journals

- Sara Beschi, Daniela Fogli, Luigi Gargioni, and Angela Locoro. “AI-Supported EUD for Data Visualization: an Exploratory Case Study Research”. Future Internet, 2025, 17(8), 349.
  https://doi.org/10.3390/fi17080349

- Luigi Gargioni, Daniela Fogli, and Pietro Baroni. “Exploring the Adoption of Collaborative Robots for the Preparation of Galenic Formulations”. Information, 2025, 16(7), 559.
  https://doi.org/10.3390/info16070559

- Luigi Gargioni, Daniela Fogli, and Pietro Baroni. “Preparation of Personalized Medicines through Collaborative Robots: A Hybrid Approach to the End-User Development of Robot Programs”. ACM Journal of Responsible Computing. 2025, Volume 2, Issue 3, Article No. 13, Pages 1-26.
  https://doi.org/10.1145/3715852

- Luigi Gargioni, Daniela Fogli, and Pietro Baroni. “A Systematic Review on Pill and Medication Dispensers from a Human-Centered Perspective”. Journal of Healthcare Informatics Research, 2024, 8(2), 244-285.
  https://doi.org/10.1007/s41666-024-00161-w

- Daniela Fogli, Luigi Gargioni, Giovanni Guida, and Fabio Tampalini. “A hybrid approach to user-oriented programming of collaborative robots”. Robotics and Computer-Integrated Manufacturing, 2022/2/1, Volume 73, 102234.
  https://doi.org/10.1016/j.rcim.2021.102234

Book Chapters

- Luigi Gargioni and Daniela Fogli. "Empowering Worker-Robot Collaboration: Leveraging LLMs for Extracting and Visualizing Robot Task Metrics". IFIP Springer book in the Human-Work Interaction Design series, HWID 2024, IFIP AICT 751, pp. 1–15, 2026.
  https://doi.org/10.1007/978-3-031-95334-7_13

Conference Proceedings

- Barbara Rita Barricelli, Daniela Fogli, Luigi Gargioni, Angela Locoro, and Stefano Valtolina. "Towards the Unification of Computational Thinking and EUDability: Two Cases from Healthcare". Proceedings of the 17th International Conference on Advanced Visual Interfaces (AVI). Long paper. June 3-7, 2024. Arenzano, Genoa, Italy.
  https://dl.acm.org/doi/pdf/10.1145/3656650.3656671

- Luigi Gargioni and Daniela Fogli. "Integrating ChatGPT with Blockly for End-User Development of Robot Tasks". In 19th Annual ACM/IEEE International Conference on Human Robot Interaction (HRI), March 11–14, 2024, Boulder, CO. ACM, New York, NY, USA, Pages 478-482.
  https://doi.org/10.1145/3610978.3640653

- Luigi Gargioni, Daniela Fogli, and Pietro Baroni. "Designing Human-Robot Collaboration for the Preparation of Personalized Medicines". Proceedings of the 2023 ACM Conference on Information Technology for Social Good. 2023. Lisbon, Portugal, September 6-8, 2023. ACM GoodIT'23 Work in Progress Conference Proceedings, Pages 135-140.
  https://doi.org/10.1145/3582515.3609527

Workshop Proceedings

- Luigi Gargioni and Daniela Fogli. "A Meta-Design Approach to Collaborative Robotics to Achieve Sustainability Goals". Proceedings of the 9th International Workshop on Cultures of Participation in the Digital Age (CoPDA 2025): Sustainability Perspectives and Frameworks for Making Cultures of Participation Successful, June 16-18, 2025, Munich, Germany. In conjunction with IS-EUD 2025. CEUR Workshop Proceedings 3978, CEUR-WS.org 2025.
  https://ceur-ws.org/Vol-3978/short-s2-02.pdf

- Luigi Gargioni, Rachid Alami, and Daniela Fogli. "Towards a Hybrid LLM/Model-Based Architecture for Robot Coaching: An Instance of Human-Machine Collaboration". AutomationXP25: Hybrid Automation Experiences, April 27, 2025, Yokohama, Japan. In conjunction with ACM CHI'25. CEUR Workshop Proceedings 4101, CEUR-WS.org 2025.
  https://ceur-ws.org/Vol-4101/paper6.pdf

- Luigi Gargioni, Barbara Rita Barricelli, Daniela Fogli, and Angela Locoro. "Exploring the Role of End Users in Performing EUD with Large Language Models". Joint Proceedings of the 17th International Conference on Advanced Visual Interfaces (AVI). June 3-7, 2024. Arenzano, Genoa, Italy. CEUR Workshop Proceedings 3685, CEUR-WS.org 2024.
  https://ceur-ws.org/Vol-3685/short3.pdf

- Giorgio Bimbatti, Daniela Fogli, and Luigi Gargioni. "Can ChatGPT Support End-User Development of Robot Programs?". Joint Proceedings of the Workshops, Work in Progress Demos and Doctoral Consortium at the IS-EUD 2023, Cagliari, Italy, June 6-8, 2023. CEUR Workshop Proceedings 3408, CEUR-WS.org 2023.
https://ceur-ws.org/Vol-3408/short-s2-03.pdf

Doctoral Consortia

- Luigi Gargioni. "Empowering Users in Human-Robot Interaction through End-User Development and Large Language Models". Proceedings of the 16th Biannual Conference of the Italian SIGCHI Chapter (CHItaly). Doctoral Consortium. October 6-10, 2025. Salerno, Italy.
https://ceur-ws.org/Vol-4151/short5.pdf

- Luigi Gargioni. "Advancing End-User Development in Robotics with Large Language Models and Digital Twins". Joint Proceedings of the Workshops, Work in Progress Demos and Doctoral Consortium at the IS-EUD 2025, Munich, Germany, June 16-18, 2025. CEUR Workshop Proceedings 3978, CEUR-WS.org 2025.
https://ceur-ws.org/Vol-3978/short-s1-01.pdf

- Luigi Gargioni. "Emerging Approaches to Human-Robot Collaboration in Healthcare". Proceedings of the 17th International Conference on Advanced Visual Interfaces (AVI). Doctoral Consortium. June 3-7, 2024. Arenzano, Genoa, Italy.
https://dl.acm.org/doi/10.1145/3656650.3656747

- Luigi Gargioni. "Human-robot collaboration in healthcare: new programming and interaction techniques". Joint Proceedings of the Workshops, Work in Progress Demos and Doctoral Consortium at the IS-EUD 2023, Cagliari, Italy, June 6-8, 2023. CEUR Workshop Proceedings 3408, CEUR-WS.org 2023.
https://ceur-ws.org/Vol-3408/short-s1-01.pdf

Under Review

- Luigi Gargioni, Rachid Alami, and Daniela Fogli. "A Hybrid LLM/Model-Based Architecture for Flexible and Adaptive Robot Coaching". International Journal of Social Robotics.

- Luigi Gargioni and Daniela Fogli. "A Digital Twin-Enabled Environment for End-User Development of Robot Tasks". ACM Transactions on Human-Robot Interaction.

Bibliography

[1] Eshtiak Ahmed, Oğuz 'Oz' Buruk, and Juho Hamari. Human–robot companionship: Current trends and future agenda. *International Journal of Social Robotics*, 16(8): 1809–1860, 2024.

[2] Mohamed Adlan Ait Ameur, Erfu Yang, and Scott Brady. Advancing healthcare through intelligent human-robot collaboration: overview, challenges and opportunities. In *2024 29th International Conference on Automation and Computing (ICAC)*, pages 1–6. IEEE, 2024.

[3] Gopika Ajaykumar, Maureen Steele, and Chien-Ming Huang. A survey on end-user robot programming. *ACM Comput. Surv.*, 54(8), October 2021. ISSN 0360-0300. doi: 10.1145/3466819.

[4] Pierre A. Akiki, Paul A. Akiki, Arosha K. Bandara, and Yijun Yu. Eud-mars: End-user development of model-driven adaptive robotics software systems. *Science of Computer Programming*, 200:102534, 2020. ISSN 0167-6423. doi: https://doi.org/10.1016/j.scico.2020.102534.

[5] Rachid Alami, Aurélie Clodic, Vincent Montreuil, Emrah Akin Sisbot, and Raja Chatila. Toward human-aware robot task planning. In *AAAI spring symposium: to boldly go where no human-robot team has gone before*, pages 39–46, 2006.

[6] Sonya Alexandrova, Zachary Tatlock, and Maya Cakmak. Roboflow: A flow-based visual programming language for mobile manipulation tasks. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5537–5544, 2015. doi: 10.1109/ICRA.2015.7139973.

[7] Najat Almasarwah, Esraa Abdelall, Gursel A Suer, Jesus Pagan, and Yuqiu You. Collaborative robots' assembly system in the manufacturing area, assembly system 4.0. *The International Journal of Advanced Manufacturing Technology*, 122(2):1069–1081, 2022.

[8] Margherita Andrao, Barbara Treccani, and Massimo Zancanaro. Making trigger-action rules more comprehensible: Investigating which linguistic clues effectively guide non-programmers. *International Journal of Human-Computer Studies*, page 103613, 2025.

[9] Nilüfer Atman Uslu, Gulay Öztüre Yavuz, and Yasemin Kocak Usluel. A systematic review study on educational robotics and robots. *Interactive Learning Environments*, 31(9):5874–5898, 2023.

[10] Tewodros W Ayalew, Jennifer Wang, Michael L Littman, Blase Ur, and Sarah Sebo. Enabling end users to program robots using reinforcement learning. In *2025 20th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, pages 767–777. IEEE, 2025.

[11] Aaron Bangor, Philip T Kortum, and James T Miller. An empirical evaluation of the system usability scale. *Intl. Journal of Human–Computer Interaction*, 24(6): 574–594, 2008.

[12] Barbara Rita Barricelli, Fabio Cassano, Daniela Fogli, and Antonio Piccinno. End-user development, end-user programming and end-user software engineering: A systematic mapping study. *Journal of Systems and Software*, 149:101–137, 2019. doi: 10.1016/j.jss.2018.11.041.

[13] Barbara Rita Barricelli, Daniela Fogli, and Angela Locoro. EUDability: A new construct at the intersection of end-user development and computational thinking. *Journal of Systems and Software*, 195:111516, 2023. ISSN 0164-1212. doi: 10.1016/j.jss.2022.111516.

[14] Barbara Rita Barricelli, Daniela Fogli, and Angela Locoro. Designing for a sustainable digital transformation: the dea methodology. In *International Symposium on End User Development*, pages 189–199. Springer, 2023.

[15] Barbara Rita Barricelli, Daniela Fogli, and Angela Locoro. Eudability: A new construct at the intersection of end-user development and computational thinking. *Journal of Systems and Software*, 195:111516, 2023.

[16] Barbara Rita Barricelli, Daniela Fogli, Luigi Gargioni, Angela Locoro, and Stefano Valtolina. Towards the unification of computational thinking and eudability: Two cases from healthcare. In *Proceedings of the 2024 International Conference on Advanced Visual Interfaces*, AVI '24, pages 1–9, New York, NY, USA, 2024. Association for Computing Machinery. ISBN 9798400717642. doi: 10.1145/3656650.3656671.

[17] Christoph Bartneck, Tony Belpaeme, Friederike Eyssel, Takayuki Kanda, Merel Keijsers, and Selma Šabanović. *Human-robot interaction: An introduction*. Cambridge University Press, 2024.

[18] Emanuele Bastianelli, Daniele Nardi, Luigia Carlucci Aiello, Fabrizio Giacomelli, and Nicolamaria Manes. Speaky for robots: the development of vocal interfaces for robotic applications. *Applied Intelligence*, 44(1):43–66, January 2016. ISSN 0924-669X. doi: 10.1007/s10489-015-0695-5.

[19] Paul E Baxter, Joachim de Greeff, and Tony Belpaeme. Cognitive architecture for human–robot interaction: Towards behavioural alignment. *Biologically Inspired Cognitive Architectures*, 6:30–39, 2013.

[20] RG Beri, D Wolton, and CH Coulon. Opportunities for modern robotics in biologics manufacturing. *Bioprocess Int*, 17(4), 2019.

[21] Sara Beschi, Daniela Fogli, and Fabio Tampalini. Capirci: a multi-modal system for collaborative robot programming. In *End-User Development: 7th International Symposium, IS-EUD 2019, Hatfield, UK, July 10–12, 2019, Proceedings 7*, pages 51–66. Springer, 2019.

[22] Sara Beschi, Daniela Fogli, Luigi Gargioni, and Angela Locoro. Ai-supported eud for data visualization: An exploratory case study. *Future Internet*, 17(8):349, 2025.

[23] Pietro Bilancia, Juliana Schmidt, Roberto Raffaeli, Margherita Peruzzini, and Marcello Pellicciari. An overview of industrial robots control and programming approaches. *Applied Sciences*, 13(4):2582, 2023.

[24] Jonathan Boutin, Guillaume Dupoiron, and Jean-Philippe Roberge. Virtual reality as a robot programming tool and a comparison of its effectiveness in common tasks. In *2024 IEEE 20th International Conference on Automation Science and Engineering (CASE)*, pages 1805–1812, 2024. doi: 10.1109/CASE59546.2024.10711441.

[25] Virginia Braun and Victoria Clarke. Using thematic analysis in psychology. *Qualitative research in psychology*, 3(2):77–101, 2006.

[26] Richard W Brislin. Cross-cultural research methods: Strategies, problems, applications. In *Environment and culture*, pages 47–82. Springer, 1980.

[27] John Brooke et al. Sus-a quick and dirty usability scale. *Usability evaluation in industry*, 189(194):4–7, 1996.

[28] Nina Buchina, Sherin Kamel, and Emilia Barakova. Design and evaluation of an end-user friendly tool for robot programming. In *2016 25th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)*, pages 185–191, 2016. doi: 10.1109/ROMAN.2016.7745109.

[29] Guilhem Buisan, Anthony Favier, Amandine Mayima, and Rachid Alami. Hatp/ehda: A robot task planner anticipating and eliciting human decisions and actions. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 2818–2824. IEEE, 2022.

[30] André Cardoso, Ana Colim, Estela Bicho, Ana Cristina Braga, Marino Menozzi, and Pedro Arezes. Ergonomics and human factors as a requirement to implement safer collaborative robotic workstations: A literature review. *Safety*, 7(4):71, 2021.

[31] Stefan Carmien, Melissa Dawe, Gerhard Fischer, Andrew Gorman, Anja Kintsch, and James F Sullivan Jr. Socio-technical environments supporting people with cognitive disabilities using public transportation. *ACM Transactions on Computer-Human Interaction (TOCHI)*, 12(2):233–262, 2005.

[32] Meia Chita-Tegmark and Matthias Scheutz. Assistive robots for the social management of health: a framework for robot design and human–robot interaction research. *International Journal of Social Robotics*, 13(2):197–217, 2021.

[33] Aurélie Clodic, Elisabeth Pacherie, Rachid Alami, and Raja Chatila. Key elements for human-robot joint action. *Sociality and normativity for robots: philosophical inquiries into human-robot interactions*, pages 159–177, 2017.

[34] J Edward Colgate, Witaya Wannasuphoprasit, and Michael A Peshkin. Cobots: Robots for collaboration with human operators. In *ASME international mechanical engineering congress and exposition*, volume 15281, pages 433–439. American Society of Mechanical Engineers, 1996.

[35] Eric Corbett and Astrid Weber. What can i say? addressing user experience challenges of a mobile voice user interface for accessibility. In *Proceedings of the 18th international conference on human-computer interaction with mobile devices and services*, pages 72–82, 2016.

[36] Richard Augustus Cripps. *Galenic Pharmacy: A Practical Handbook to the Processes of the British Pharmacopoeia*. J. & A. Churchill, 1893.

[37] Kourosh Darvish, Enrico Simetti, Fulvio Mastrogiovanni, and Giuseppe Casalino. A hierarchical architecture for human–robot cooperation processes. *IEEE Transactions on Robotics*, 37(2):567–586, 2020.

[38] Jose Pablo De La Rosa Gutierrez, Thiago Rocha Silva, Yvonne Dittrich, and Anders Stengaard Sørensen. Design goals for end-user development of robot-assisted physical training activities: a participatory design study. *Proceedings of the ACM on Human-Computer Interaction*, 8(EICS):1–31, 2024.

[39] Ilias El Makrini, Kelly Merckaert, Dirk Lefeber, and Bram Vanderborght. Design of a collaborative architecture for human-robot assembly tasks. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1624–1629, 2017.

[40] EudraLex European Commission. Good Manufacturing Practice (GMP) guidelines, 2023. URL https://health.ec.europa.eu/medicinal-products/eudralex/eudralex-volume-4_en.

[41] Eurostat. Ageing Europe - Statistics on population developments, 2020. URL https://ec.europa.eu/eurostat/statistics-explained/index.php?title=Archive:Ageing_Europe_-_statistics_on_population_developments.

[42] Anthony Favier and Rachid Alami. A model of concurrent and compliant human-robot joint action to plan and supervise collaborative robot actions. In *Advances in Cognitive Systems (ACS)*, pages 1–16, 2024.

[43] Gerhard Fischer. Domain-oriented design environments. *Automated software engineering*, 1(2):177–203, 1994.

[44] Gerhard Fischer and Elisa Giaccardi. Meta-design: A framework for the future of end-user development. In *End user development*, pages 427–457. Springer, 2006.

[45] Gerhard Fischer, Daniela Fogli, and Antonio Piccinno. Revisiting and broadening the meta-design framework for end-user development. In *New perspectives in end-user development*, pages 61–97. Springer, 2017.

[46] Pasquale Foggia, Antonio Greco, Antonio Roberto, Alessia Saggese, and Mario Vento. A social robot architecture for personalized real-time human-robot interaction. *IEEE Internet of Things Journal*, 2023.

[47] Daniela Fogli, Luigi Gargioni, Giovanni Guida, and Fabio Tampalini. A hybrid approach to user-oriented programming of collaborative robots. *Robotics and Computer-Integrated Manufacturing*, 73:102234, 2022. doi: 10.1016/j.rcim.2021.102234.

[48] Simone Gallo, Giacomo Vaiani, and Fabio Paternò. End-user personalisation of humanoid robot behaviour through vocal interaction. In Francesca Cocchella, Omar Eldardeer, Marco Manca, Marco Matarese, Andrea Rezzani, and Eleonora Zedda, editors, *Proceedings of the 1st Workshop on Robots for Humans (RfH 2024) co-located with 17th edition of the Conference on Advanced Visual Interfaces (AVI 2024), Arenzano (Genoa), Italy, June 3rd 2024*, volume 3794 of *CEUR Workshop Proceedings*, page 4. CEUR-WS.org, 2024. URL https://ceur-ws.org/Vol-3794/paper3.pdf.

[49] Luigi Gargioni and Daniela Fogli. Empowering worker-robot collaboration: Leveraging llms for extracting and visualizing robot task metrics. In *IFIP Working Conference on Human Work Interaction Design*, pages 208–222. Springer, 2024.

[50] Luigi Gargioni and Daniela Fogli. Integrating chatgpt with blockly for end-user development of robot tasks. In *Companion of the 2024 ACM/IEEE International Conference on Human-Robot Interaction*, HRI '24, page 478–482, New York, NY, USA, 2024. Association for Computing Machinery. ISBN 9798400703232. doi: 10.1145/3610978.3640653.

[51] Luigi Gargioni, Daniela Fogli, and Pietro Baroni. Designing human-robot collaboration for the preparation of personalized medicines. In *Proceedings of the 2023 ACM Conference on Information Technology for Social Good*, GoodIT '23, page 135–140, New York, NY, USA, 2023. Association for Computing Machinery. ISBN 9798400701160. doi: 10.1145/3582515.3609527.

[52] Luigi Gargioni, Barbara Rita Barricelli, Daniela Fogli, Angela Locoro, et al. Exploring the role of end users in performing eud with large language models. In *CEUR Workshop Proceedings*, volume 3685, pages 1–8. CEUR-WS, 2024.

[53] Luigi Gargioni, Daniela Fogli, and Pietro Baroni. A systematic review on pill and medication dispensers from a human-centered perspective. *Journal of healthcare informatics research*, 8(2):244–285, 2024.

[54] Luigi Gargioni, Rachid Alami, and Daniela Fogli. Towards a hybrid llm/model-based architecture for robot coaching: An instance of human-machine collaboration. In *CEUR Workshop Proceedings*, volume 4101, pages 1–8. CEUR-WS, 2025.

[55] Luigi Gargioni, Daniela Fogli, and Pietro Baroni. Exploring the adoption of collaborative robots for the preparation of galenic formulations. *Information*, 16(7):559, 2025.

[56] Luigi Gargioni, Daniela Fogli, and Pietro Baroni. A meta-design approach to collaborative robotics to achieve sustainability goals. In *CEUR Workshop Proceedings*, volume 3978, pages 1–8. CEUR-WS, 2025.

[57] Luigi Gargioni, Daniela Fogli, and Pietro Baroni. Preparation of personalized medicines through collaborative robots: A hybrid approach to the end-user development of robot programs. *ACM Journal on Responsible Computing*, 2025.

[58] Yate Ge, Yi Dai, Run Shan, Kechun Li, Yuanda Hu, and Xiaohua Sun. Cocobo: Exploring large language models as the engine for end-user robot programming.

In *2024 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC)*, pages 89–95. IEEE, 2024.

[59] Riccardo Gervasi, Matteo Capponi, Luca Mastrogiacomo, and Fiorenzo Franceschini. Manual assembly and human–robot collaboration in repetitive assembly processes: a structured comparison based on human-centered performances. *The International Journal of Advanced Manufacturing Technology*, 126(3):1213–1231, 2023.

[60] Matthew C Gombolay, Reymundo A Gutierrez, Shanelle G Clarke, Giancarlo F Sturla, and Julie A Shah. Decision-making authority, team efficiency and human worker satisfaction in mixed human–robot teams. *Autonomous Robots*, 39:293–312, 2015.

[61] Miguel Á González-Santamarta, Francisco J Rodríguez-Lera, Ángel Manuel Guerrero-Higueras, and Vicente Matellán-Olivera. Integration of large language models within cognitive architectures for autonomous robots. *arXiv preprint arXiv:2309.14945*, 2023.

[62] Google. Blockly, 2012. URL https://developers.google.com/blockly.

[63] Google. Google bard. https://bard.google.com/chat, 2023. Accessed: September 25th, 2023.

[64] Ahmad Hafez, Alireza Naderi Akhormeh, Amr Hegazy, and Amr Alanwar. Safe llm-controlled robots with formal guarantees via reachability analysis. *arXiv:2503.03911*, 2025.

[65] Tamas Haidegger, Vincent Mai, Carl Maria Mörch, DO Boesl, An Jacobs, Alaa Khamis, Luca Lach, Bram Vanderborght, et al. Robotics: Enabler and inhibitor of the sustainable development goals. *Sustainable Production and Consumption*, 43: 422–434, 2023.

[66] Sandra G Hart and Lowell E Staveland. Development of nasa-tlx (task load index): Results of empirical and theoretical research. In *Advances in psychology*, volume 52, pages 139–183. Elsevier, 1988.

[67] Bradley Hayes and Julie A Shah. Improving robot controller transparency through autonomous policy explanation. In *Proceedings of the 2017 ACM/IEEE international conference on human-robot interaction*, pages 303–312, 2017.

[68] Marcel Heerink, Ben Kröse, Vanessa Evers, and Bob Wielinga. Assessing acceptance of assistive social agent technology by older adults: the almere model. *Internationl Journal of Social Robotics*, 2:361—-375, 2010.

[69] Guy Hoffman and Cynthia Breazeal. Effects of anticipatory action on human-robot teamwork efficiency, fluency, and perception of team. In *Proceedings of the ACM/IEEE international conference on Human-robot interaction*, 2007.

[70] Guy Hoffman, Tapomayukh Bhattacharjee, and Stefanos Nikolaidis. Inferring human intent and predicting human action in human–robot collaboration. *Annual Review of Control, Robotics, and Autonomous Systems*, 7, 2024.

[71] James Hollan, Edwin Hutchins, and David Kirsh. Distributed cognition: toward a new foundation for human-computer interaction research. *ACM Transactions on Computer-Human Interaction (TOCHI)*, 7(2):174–196, 2000.

[72] Justin Huang and Maya Cakmak. Code3: A system for end-to-end programming of mobile manipulator robots for novices and experts. In *Proceedings of the 2017 ACM/IEEE International Conference on Human-Robot Interaction*, HRI '17, page 453–462, New York, NY, USA, 2017. Association for Computing Machinery. ISBN 9781450343367. doi: 10.1145/2909824.3020215.

[73] Edwin Hutchins. *Cognition in the Wild*. MIT press, 1995.

[74] Bryce Ikeda, Maitrey Gramopadhye, LillyAnn Nekervis, and Daniel Szafir. Marcer: Multimodal augmented reality for composing and executing robot tasks. In *2025 20th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, pages 529–539. IEEE, 2025.

[75] Business Research Insights. Pharma market size, share, growth, and industry growth by type (prescription-based drugs and over-the-counter drugs) by application (hospital pharmacies, retail pharmacies/ drug stores, and others) regional forecast to 2028, 2025. URL https://www.businessresearchinsights.com/market-reports/pharma-market-102426.

[76] Aayush Jain, Philip Long, Valeria Villani, John D Kelleher, and Maria Chiara Leva. Cobt: Collaborative programming of behaviour trees from one demonstration for robot manipulation. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 12993–12999. IEEE, 2024.

[77] Philipp Jennes and Alberto Di Minin. Cobots in smes: implementation processes, challenges, and success factors. In *2023 IEEE International Conference on Technology and Entrepreneurship (ICTE)*, pages 80–85. IEEE, 2023.

[78] Du Jing, Dao Ronggui, Wu Peng, Zhang Yongshou, Wu Bogang, and Wu Guiwen. Application of collaborative robot in cigarette production line for automatic distribution of packaging materials. In *International Conference on Artificial Intelligence, Robotics, and Communication*, pages 185–193. Springer, 2022.

[79] Ibrahim Kadri, Sid Ahmed Selouani, Mohsen Ghribi, Rayen Ghali, and Sabrina Mekhoukh. Llm-driven agent for speech-enabled control of industrial robots: A case study in snow-crab quality inspection. *Results in Engineering*, page 106660, 2025.

[80] Subbarao Kambhampati. Can large language models reason and plan? *Annals of the New York Academy of Sciences*, 1534(1):15–18, 2024.

[81] Ulas Berk Karli, Juo-Tung Chen, Victor Nikhil Antony, and Chien-Ming Huang. Alchemist: Llm-aided end-user development of robot applications. In *Proceedings of the 2024 ACM/IEEE International Conference on Human-Robot Interaction*, HRI '24, page 361–370, New York, NY, USA, 2024. Association for Computing Machinery. ISBN 9798400703225. doi: 10.1145/3610977.3634969.

[82] John F Kelley. An iterative design methodology for user-friendly natural language office information applications. *ACM Transactions on Information Systems (TOIS)*, 2(1):26–41, 1984.

[83] Ali Keshvarparast, Daria Battini, Olga Battaia, and Amir Pirayesh. Collaborative robots in manufacturing and assembly systems: literature review and future research agenda. *Journal of Intelligent Manufacturing*, 35(5):2065–2118, 2024.

[84] Dongjun Kim, Minji Kim, Jaehak Lee, Dongyoon Lee, and Jumyung Um. Reconfigurable machine tending with collaborative robots: leveraging iso 21919 for enhancing flexibility and safety. *International Journal of Computer Integrated Manufacturing*, pages 1–20, 2025.

[85] Yeseung Kim, Dohyun Kim, Jieun Choi, Jisang Park, Nayoung Oh, and Daehyung Park. A survey on integration of large language models with intelligent robots. *Intelligent Service Robotics*, 17(5):1091–1107, 2024.

[86] Amy J Ko, Robin Abraham, Laura Beckwith, Alan Blackwell, Margaret Burnett, Martin Erwig, Chris Scaffidi, Joseph Lawrance, Henry Lieberman, Brad Myers, et al. The state of the art in end-user software engineering. *ACM Computing Surveys (CSUR)*, 43(3):1–44, 2011.

[87] Moritz Körber. Theoretical considerations and development of a questionnaire to measure trust in automation. In *Congress of the International Ergonomics Association*, pages 13–30. Springer, 2018.

[88] Andrew Kyle Lampinen. Can language models handle recursively nested grammatical structures? A case study on comparing models and humans, 2023. URL https://arxiv.org/abs/2210.15303.

[89] Bettina Laugwitz, Theo Held, and Martin Schrepp. Construction and evaluation of a user experience questionnaire. In *HCI and Usability for Education and Work: 4th Symposium of the Workgroup Human-Computer Interaction and Usability Engineering of the Austrian Computer Society, USAB 2008, Graz, Austria, November 20-21, 2008. Proceedings 4*, pages 63–76. Springer, 2008.

[90] Séverin Lemaignan, Mathieu Warnier, E Akin Sisbot, Aurélie Clodic, and Rachid Alami. Artificial cognition for social human–robot interaction: An implementation. *Artificial Intelligence*, 247:45–69, 2017.

[91] James R Lewis and Jeff Sauro. The factor structure of the system usability scale. In *International conference on human centered design*, pages 94–103. Springer, 2009.

[92] James R Lewis and Jeff Sauro. Item benchmarks for the system usability scale. *Journal of Usability studies*, 13(3), 2018.

[93] Ci-Jyun Liang, Wes McGee, Carol C Menassa, and Vineet R Kamat. Real-time state synchronization between physical construction robots and process-level digital twins. *Construction Robotics*, 6(1):57–73, 2022. doi: 10.1007/s41693-022-00068-1.

[94] Henry Lieberman, Fabio Paternò, and Volker Wulf. *End User Development (Human-Computer Interaction Series)*. Springer-Verlag, Berlin, Heidelberg, 2006. ISBN 1402042205.

[95] Zhi Wei Lim, Krithi Pushpanathan, Samantha Min Er Yew, Yien Lai, Chen-Hsin Sun, Janice Sing Harn Lam, David Ziyou Chen, Jocelyn Hui Lin Goh, Marcus Chun Jin Tan, Bin Sheng, et al. Benchmarking large language models' performances for myopia care: a comparative analysis of chatgpt-3.5, chatgpt-4.0, and google bard. *EBioMedicine*, 95, 2023.

[96] Li Liu, Fu Guo, Zishuai Zou, and Vincent G Duffy. Application, development and future opportunities of collaborative robots (cobots) in manufacturing: A literature review. *International Journal of Human–Computer Interaction*, 40(4):915–932, 2024.

[97] Yueyue Liu, Zhijun Li, Huaping Liu, and Zhen Kan. Skill transfer learning for autonomous robots and human–robot cooperation: A survey. *Robotics and Autonomous Systems*, 128:103515, 2020. ISSN 0921-8890. doi: https://doi.org/10.1016/j.robot.2020.103515.

[98] Abraham S Luchins. Mechanization in problem solving: The effect of einstellung. *Psychological monographs*, 54(6):i, 1942.

[99] Hamza Mahdi, Sami Alperen Akgun, Shahed Saleh, and Kerstin Dautenhahn. A survey on the design and evolution of social robots—past, present and future. *Robotics and Autonomous Systems*, 156:104193, 2022.

[100] Mehrbod Manavi, Felix Carros, Isabel Schwaninger, Richard Paluch, Dave Randall, Maximilian Krueger, Ilayda Tufan, Rainer Wieching, Peter Tolmie, Claudia Mueller, et al. "making robots usable": Empowering care professionals to customize social robots in care. *International Journal of Social Robotics*, pages 1–22, 2025.

[101] Abraham Harold Maslow. *The Psychology of Science: A Reconnaissance*. Harper & Row, Manhattan, New York, USA, 1699. ISBN 0976040239.

[102] Robins Mathew, Robert McGee, Kevin Roche, Shada Warreth, and Nikolaos Papakostas. Introducing mobile collaborative robots into bioprocessing environments: Personalised drug manufacturing and environmental monitoring. *Applied Sciences*, 12(21):10895, 2022.

[103] Carlo Mazzola, Hassan Ali, Kristína Malinovská, and Igor Farkaš. Toward an interaction-centered approach to robot trustworthiness. *arXiv preprint arXiv:2508.13976*, 2025.

[104] Elena Merlo, Edoardo Lamon, Fabio Fusaro, Marta Lorenzini, Alessandro Carfì, Fulvio Mastrogiovanni, and Arash Ajoudani. An ergonomic role allocation framework for dynamic human–robot collaborative tasks. *Journal of Manufacturing Systems*, 67:111–121, 2023.

[105] Meta. Meta llama 2. https://ai.meta.com/llama/, 2023. Accessed: September 25th, 2023.

[106] Dipendra K. Misra, Jaeyong Sung, Kevin Lee, and Ashutosh Saxena. Tell me dave: Context-sensitive grounding of natural language to manipulation instructions. *Int. J. Rob. Res.*, 35(1–3):281–300, jan 2016. ISSN 0278-3649. doi: 10.1177/0278364915602060.

[107] Wenxuan Mou, Martina Ruocco, Debora Zanatto, and Angelo Cangelosi. When would you trust a robot? a study on trust and theory of mind in human-robot interactions. In *IEEE Int. Conference on Robot and Human Interactive Communication*, 2020.

[108] Michael J Muller and Sarah Kuhn. Participatory design. *Communications of the ACM*, 36(6):24–28, 1993.

[109] Jakob Nielsen. Enhancing the explanatory power of usability heuristics. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '94, page 152–158, New York, NY, USA, 1994. Association for Computing Machinery. ISBN 0897916506. doi: 10.1145/191666.191729.

[110] Donald A Norman. *The psychology of everyday things.* Basic books, 1988.

[111] OpenAI. Chatgpt, Accessed: October 20th, 2025. URL https://chatgpt.com/.

[112] OpenAI. Chatgpt api documentation, Accessed: October 20th, 2025. URL https://platform.openai.com/docs/guides/text.

[113] John Páez and Enrique González. Human-robot scaffolding: An architecture to foster problem-solving skills. *ACM Transactions on Human-Robot Interaction (THRI)*, 11(3):1–17, 2022.

[114] Mateusz Paliga. Human–cobot interaction fluency and cobot operators' job performance. the mediating role of work engagement: A survey. *Robotics and Autonomous Systems*, 155:104191, 2022.

[115] Fabio Paternò and Volker Wulf, editors. *New Perspectives in End-User Development*. Springer, Cham, 2017. doi: 10.1007/978-3-319-60291-2.

[116] Noah Patton, Kia Rahmani, Meghana Missula, Joydeep Biswas, and Işıl Dillig. Programming-by-demonstration for long-horizon robot tasks. *Proceedings of the ACM on Programming Languages*, 8(POPL):512–545, 2024.

[117] Chris Paxton, Andrew Hundt, Felix Jonathan, Kelleher Guerin, and Gregory D. Hager. CoSTAR: Instructing collaborative robots with behavior trees and vision. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, page 564–571. IEEE Press, 2017. doi: 10.1109/ICRA.2017.7989070.

[118] Ian Peake, Joseph La Delfa, Ronal Bejarano, and Jan Olaf Blech. Simulation components in Gazebo. In *2021 22nd IEEE International Conference on Industrial Technology (ICIT)*, volume 1, pages 1169–1175, 2021. doi: 10.1109/ICIT46573.2021.9453594.

[119] Federico Pecora, Marcello Cirillo, Francesca Dell'Osa, Jonas Ullberg, and Alessandro Saffiotti. A constraint-based approach for proactive, context-aware human support. *Journal of Ambient Intelligence and Smart Environments*, 4(4):347–367, 2012.

[120] Theresa Pekarek Rosin, Vanessa Hassouna, Xiaowen Sun, Luca Krohm, Henri-Leon Kordt, Michael Beetz, and Stefan Wermter. A framework for adapting human-robot interaction to diverse user groups. In *International Conference on Social Robotics*, pages 24–38, 2024.

[121] Mirco Polonara, Alessandra Romagnoli, Gianfranco Biancini, and Luca Carbonari. Introduction of collaborative robotics in the production of automotive parts: A case study. *Machines*, 12(3):196, 2024.

[122] David Porfirio, Laura Stegner, Maya Cakmak, Allison Sauppé, Aws Albarghouthi, and Bilge Mutlu. Sketching robot programs on the fly. In *Proceedings of the 2023 ACM/IEEE International Conference on Human-Robot Interaction*, HRI '23, page 584–593, New York, NY, USA, 2023. Association for Computing Machinery. ISBN 9781450399647. doi: 10.1145/3568162.3576991.

[123] Morgan Quigley, Ken Conley, Brian Gerkey, Josh Faust, Tully Foote, Jeremy Leibs, Rob Wheeler, Andrew Y Ng, et al. Ros: an open-source robot operating system. In *ICRA workshop on open source software*, volume 3, page 5. Kobe, 2009.

[124] Kartik Ramachandruni, Cassandra Kent, and Sonia Chernova. Uhtp: A user-aware hierarchical task planning framework for communication-free, mutually-adaptive human-robot collaboration. *ACM Transactions on Human-Robot Interaction*, 13 (3), 2024.

[125] Mitchel Resnick, John Maloney, Andrés Monroy-Hernández, Natalie Rusk, Evelyn Eastmond, Karen Brennan, Amon Millner, Eric Rosenbaum, Jay Silver, Brian Silverman, and Yasmin Kafai. Scratch: Programming for all. *Commun. ACM*, 52(11): 60–67, nov 2009. ISSN 0001-0782. doi: 10.1145/1592761.1592779.

[126] Nico Ritschel, Reid Holmes, Felipe Fronchetti, Ronald Garcia, and David C Shepherd. Block-based or graph-based? why not both? designing a hybrid programming environment for end-users. *Interacting with Computers*, page iwaf028, 2025.

[127] Marta Romeo, Peter E McKenna, David A Robb, Gnanathusharan Rajendran, Birthe Nesset, Angelo Cangelosi, and Helen Hastie. Exploring theory of mind for human-robot collaboration. In *IEEE Int. Conference on Robot and Human Interactive Communication*, 2022.

[128] Maram Sakr, Zhikai Zhang, Benjamin Li, Haomiao Zhang, HF Machiel Van der Loos, Dana Kulić, and Elizabeth Croft. How can everyday users efficiently teach robots by demonstration? *ACM Transactions on Human-Robot Interaction*, 14(4): 1–22, 2025.

[129] Bernhard Schenkenfelder, Christian Salomon, Martin Schwandtner, Raphael Zefferer, Michael Derfler, and Manuel Wimmer. Leveraging the power of llms to transform robot programs into low-code. In *2024 IEEE 29th International Conference on Emerging Technologies and Factory Automation (ETFA)*, pages 1–4. IEEE, 2024.

[130] Andrew Schoen, Nathan White, Curt Henrichs, Amanda Siebert-Evenstone, David Shaffer, and Bilge Mutlu. Coframe: A system for training novice cobot programmers. In *2022 17th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, pages 185–194, 2022. doi: 10.1109/HRI53351.2022.9889345.

[131] Casper Schou, Rasmus Skovgaard Andersen, Dimitrios Chrysostomou, Simon Bøgh, and Ole Madsen. Skill-based instruction of collaborative robots in industrial settings. *Robotics and Computer-Integrated Manufacturing*, 53:72–80, 2018. ISSN 0736-5845. doi: 10.1016/j.rcim.2018.03.008.

[132] Martin Schrepp, Andreas Hinderks, and Jörg Thomaschewski. Construction of a benchmark for the user experience questionnaire (ueq). *International Journal of Interactive Multimedia and Artificial Intelligence*, 4:40–44, 06 2017.

[133] Sander Schulhoff, Michael Ilie, Nishant Balepur, Konstantine Kahadze, Amanda Liu, Chenglei Si, Yinheng Li, Aayush Gupta, HyoJung Han, Sevien Schulhoff, et al. The prompt report: a systematic survey of prompt engineering techniques. *arXiv preprint arXiv:2406.06608*, 2024.

[134] Edwards Shaun. Leveraging the open source robot operating system (ros) for new industrial applications, 2011.

[135] Ben Shneiderman. *Human-centered AI*. Oxford University Press, Oxford, UK, 2022.

[136] Kaya Stechly, Karthik Valmeekam, and Subbarao Kambhampati. On the self-verification limitations of large language models on reasoning and planning tasks. *arXiv:2402.08115*, 2024.

[137] Maj Stenmark and Pierre Nugues. Natural language programming of industrial robots. In *IEEE ISR 2013*, pages 1–5, 2013. doi: 10.1109/ISR.2013.6695630.

[138] Claudio Taesi, Francesco Aggogeri, and Nicola Pellegrini. Cobot applications—recent advances and challenges. *Robotics*, 12(3):79, 2023.

[139] Daniel Tetteroo, Paul Vreugdenhil, Ivor Grisel, Marc Michielsen, Els Kuppens, Diana Vanmulken, and Panos Markopoulos. Lessons learnt from deploying an end-user development platform for physical rehabilitation. In *Proceedings of the 33rd annual ACM conference on human factors in computing systems*, pages 4133–4142, 2015.

[140] Siniša Tomić, Anita Sučić, and Adrijana Martinac. Good manufacturing practice: the role of local manufacturers and competent authorities. *Archives of Industrial Hygiene and Toxicology*, 61(4):425–436, 2010.

[141] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023.

[142] J Gregory Trafton, Laura M Hiatt, Anthony M Harrison, Franklin P Tamborello, Sangeet S Khemlani, and Alan C Schultz. Act-r/e: An embodied cognitive architecture for human-robot interaction. *Journal of Human-Robot Interaction*, 2(1):30–55, 2013.

[143] Meng-Jung Tsai, Jyh-Chong Liang, and Chung-Yuan Hsu. The computational thinking scale for computer literacy education. *Journal of Educational Computing Research*, 59(4):579–602, 2021.

[144] Spyros G Tzafestas. Mobile robot control and navigation: A global overview. *Journal of Intelligent & Robotic Systems*, 91:35–58, 2018.

[145] Alessandro Umbrico, Riccardo De Benedictis, Francesca Fracasso, Amedeo Cesta, Andrea Orlandini, and Gabriella Cortellessa. A mind-inspired architecture for adaptive hri. *International Journal of Social Robotics*, 15(3):371–391, 2023.

[146] Vaibhav V Unhelkar, Shen Li, and Julie A Shah. Decision-making for bidirectional communication in sequential human-robot collaborative tasks. In *Proceedings of the 2020 ACM/IEEE Int.l Conference on Human-Robot Interaction*, pages 329–341, 2020.

[147] Marta Uriel, Diego Marro, and Carlota Gómez Rincón. An adequate pharmaceutical quality system for personalized preparation. *Pharmaceutics*, 15(3), 2023. ISSN 1999-4923. doi: 10.3390/pharmaceutics15030800. URL https://www.mdpi.com/1999-4923/15/3/800.

[148] Giacomo Vaiani and Fabio Paternò. End-user development for human-robot interaction: results and trends in an emerging field. *Proceedings of the ACM on Human-Computer Interaction*, 8(EICS):1–40, 2024.

[149] Sai Vemprala, Rogerio Bonatti, Arthur Bucker, and Ashish Kapoor. Chatgpt for robotics: Design principles and model abilities. Technical report, Microsoft, 2023. URL https://www.microsoft.com/en-us/research/uploads/prod/\2023/02/ChatGPT_Robotics.pdf.

[150] Jiaqi Wang, Enze Shi, Huawen Hu, Chong Ma, Yiheng Liu, Xuhui Wang, Yincheng Yao, Xuan Liu, Bao Ge, and Shu Zhang. Large language models for robotics: Opportunities, challenges, and perspectives. *Journal of Automation and Intelligence*, 4(1):52–64, 2025.

[151] David Weintrop, David C Shepherd, Patrick Francis, and Diana Franklin. Blockly goes to work: Block-based programming for industrial robots. In *2017 IEEE Blocks And Beyond Workshop (B&B)*, pages 29–36. IEEE, 2017.

[152] David Weintrop, Afsoon Afzal, Jean Salac, Patrick Francis, Boyang Li, David C. Shepherd, and Diana Franklin. Evaluating coblox: A comparative study of robotics programming environments for adult novices. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, CHI '18, page 1–12, New York, NY, USA, 2018. Association for Computing Machinery. ISBN 9781450356206. doi: 10.1145/3173574.3173940.

[153] World Health Organization. Health products policy and standards, 2024. URL https://www.who.int/teams/health-product-and-policy-standards/standards-and-specifications/norms-and-standards-for-pharmaceuticals/guidelines.

[154] Tongshuang Wu, Michael Terry, and Carrie Jun Cai. Ai chains: Transparent and controllable human-ai interaction by chaining large language model prompts. In *Proceedings of the 2022 CHI conference on human factors in computing systems*, pages 1–22, 2022.

[155] Shiyu Xu, Scott Moore, and Akansel Cosgun. Shared-control robotic manipulation in virtual reality. In *2022 International Congress on Human-Computer Interaction, Optimization and Robotic Applications (HORA)*, pages 1–6, 2022. doi: 10.1109/HORA55278.2022.9800046.

[156] JD Zamfirescu-Pereira, Richmond Y Wong, Bjoern Hartmann, and Qian Yang. Why johnny can't prompt: how non-ai experts try (and fail) to design llm prompts. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*, pages 1–21, 2023.

[157] Ceng Zhang, Junxin Chen, Jiatong Li, Yanhong Peng, and Zebing Mao. Large language models for human–robot interaction: A review. *Biomimetic Intelligence and Robotics*, 3(4), 2023.

Appendices

A Questionnaires

A.1 User Experience Questionnaire (UEQ)

Please make your evaluation now.

For the assessment of the product, please fill out the following questionnaire. The questionnaire consists of pairs of contrasting attributes that may apply to the product. The circles between the attributes represent gradations between the opposites. You can express your agreement with the attributes by ticking the circle that most closely reflects your impression.

Example:

attractive	○	⊗	○	○	○	○	○	unattractive

This response would mean that you rate the application as more attractive than unattractive.

Please decide spontaneously. Don't think too long about your decision to make sure that you convey your original impression.

Sometimes you may not be completely sure about your agreement with a particular attribute or you may find that the attribute does not apply completely to the particular product. Nevertheless, please tick a circle in every line.

It is your personal opinion that counts. Please remember: there is no wrong or right answer!

Figure A.1: User Experience Questionnaire (UEQ) - Introduction [89]

Please assess the product now by ticking one circle per line.

	1	2	3	4	5	6	7		
annoying	○	○	○	○	○	○	○	enjoyable	1
not understandable	○	○	○	○	○	○	○	understandable	2
creative	○	○	○	○	○	○	○	dull	3
easy to learn	○	○	○	○	○	○	○	difficult to learn	4
valuable	○	○	○	○	○	○	○	inferior	5
boring	○	○	○	○	○	○	○	exciting	6
not interesting	○	○	○	○	○	○	○	interesting	7
unpredictable	○	○	○	○	○	○	○	predictable	8
fast	○	○	○	○	○	○	○	slow	9
inventive	○	○	○	○	○	○	○	conventional	10
obstructive	○	○	○	○	○	○	○	supportive	11
good	○	○	○	○	○	○	○	bad	12
complicated	○	○	○	○	○	○	○	easy	13
unlikable	○	○	○	○	○	○	○	pleasing	14
usual	○	○	○	○	○	○	○	leading edge	15
unpleasant	○	○	○	○	○	○	○	pleasant	16
secure	○	○	○	○	○	○	○	not secure	17
motivating	○	○	○	○	○	○	○	demotivating	18
meets expectations	○	○	○	○	○	○	○	does not meet expectations	19
inefficient	○	○	○	○	○	○	○	efficient	20
clear	○	○	○	○	○	○	○	confusing	21
impractical	○	○	○	○	○	○	○	practical	22
organized	○	○	○	○	○	○	○	cluttered	23
attractive	○	○	○	○	○	○	○	unattractive	24
friendly	○	○	○	○	○	○	○	unfriendly	25
conservative	○	○	○	○	○	○	○	innovative	26

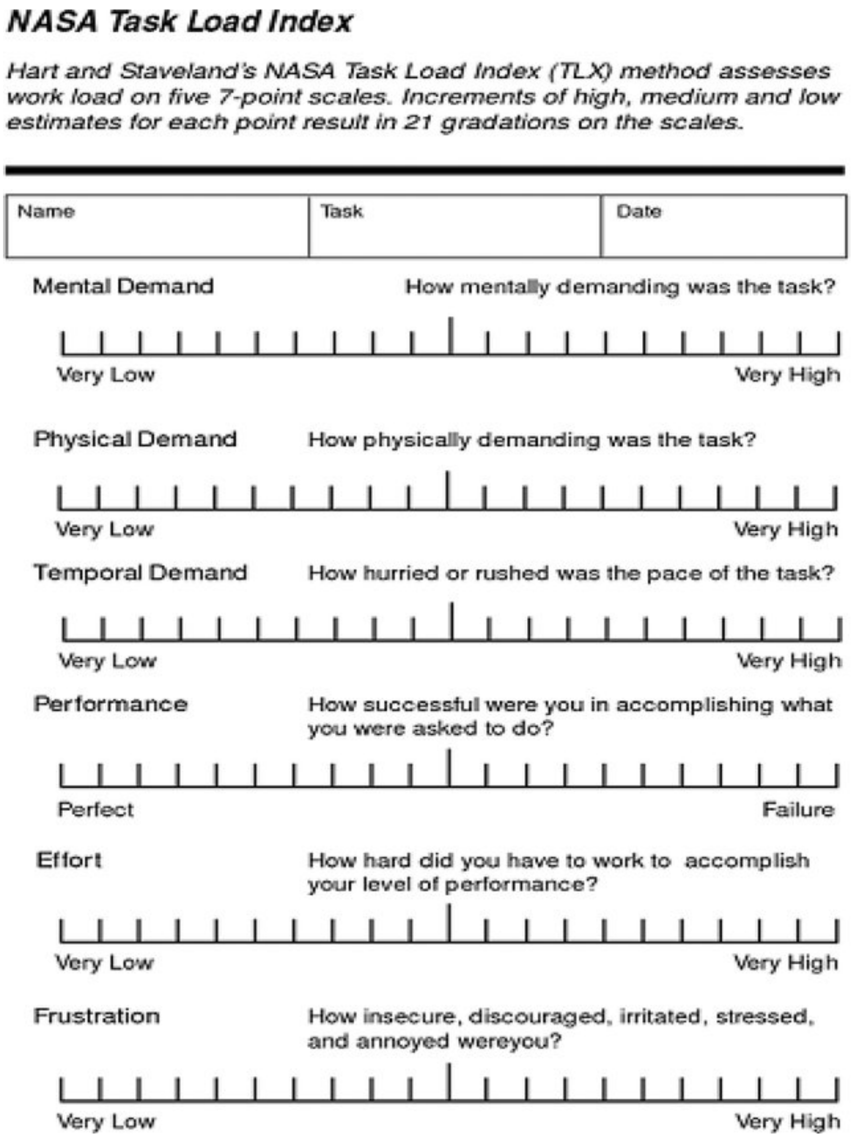
Figure A.2: User Experience Questionnaire (UEQ) - Questionnaire [89]

A.2 System Usability Scale (SUS)

	The System Usability Scale Standard Version	Strongly Disagree 1	2	3	4	Strongly Agree 5
1	I think that I would like to use this system frequently.	O	O	O	O	O
2	I found the system unnecessarily complex.	O	O	O	O	O
3	I thought the system was easy to use.	O	O	O	O	O
4	I think that I would need the support of a technical person to be able to use this system.	O	O	O	O	O
5	I found the various functions in this system were well integrated.	O	O	O	O	O
6	I thought there was too much inconsistency in this system.	O	O	O	O	O
7	I would imagine that most people would learn to use this system very quickly.	O	O	O	O	O
8	I found the system very awkward to use.	O	O	O	O	O
9	I felt very confident using the system.	O	O	O	O	O
10	I needed to learn a lot of things before I could get going with this system.	O	O	O	O	O

Figure A.3: System Usability Scale (SUS) - Questionnaire [27]

A.3 NASA Task Load Index (NASA-TLX)



NASA Task Load Index

Hart and Staveland's NASA Task Load Index (TLX) method assesses work load on five 7-point scales. Increments of high, medium and low estimates for each point result in 21 gradations on the scales.

Name	Task	Date

Mental Demand — How mentally demanding was the task?

Very Low — Very High

Physical Demand — How physically demanding was the task?

Very Low — Very High

Temporal Demand — How hurried or rushed was the pace of the task?

Very Low — Very High

Performance — How successful were you in accomplishing what you were asked to do?

Perfect — Failure

Effort — How hard did you have to work to accomplish your level of performance?

Very Low — Very High

Frustration — How insecure, discouraged, irritated, stressed, and annoyed wereyou?

Very Low — Very High

Figure A.4: NASA Task Load Index (NASA-TLX) - Questionnaire [66]

A.4 Trust in Automation (TiA)

		Strongly disagree	Rather disagree	Neither disagree nor agree	Rather agree	Strongly agree	No response
1	The system is capable of interpreting situations correctly.	(1)	(2)	(3)	(4)	(5)	○
2	The system state was always clear to me.	(1)	(2)	(3)	(4)	(5)	○
3	I already know similar systems.	(1)	(2)	(3)	(4)	(5)	○
4	The developers are trustworthy.	(1)	(2)	(3)	(4)	(5)	○
5	One should be careful with unfamiliar automated systems.	(1)	(2)	(3)	(4)	(5)	○
6	The system works reliably.	(1)	(2)	(3)	(4)	(5)	○
7	The system reacts unpredictably.	(1)	(2)	(3)	(4)	(5)	○
8	The developers take my well-being seriously.	(1)	(2)	(3)	(4)	(5)	○
9	I trust the system.	(1)	(2)	(3)	(4)	(5)	○
10	A system malfunction is likely.	(1)	(2)	(3)	(4)	(5)	○
11	I was able to understand why things happened.	(1)	(2)	(3)	(4)	(5)	○
12	I rather trust a system than I mistrust it.	(1)	(2)	(3)	(4)	(5)	○
13	The system is capable of taking over complicated tasks.	(1)	(2)	(3)	(4)	(5)	○
14	I can rely on the system.	(1)	(2)	(3)	(4)	(5)	○
15	The system might make sporadic errors.	(1)	(2)	(3)	(4)	(5)	○
16	It is difficult to identify what the system will do next.	(1)	(2)	(3)	(4)	(5)	○
17	I have already used similar systems.	(1)	(2)	(3)	(4)	(5)	○
18	Automated systems generally work well.	(1)	(2)	(3)	(4)	(5)	○
19	I am confident about the system's capabilities.	(1)	(2)	(3)	(4)	(5)	○

Figure A.5: Trust in Automation (TiA) - Questionnaire [87]

A.5 Almere Model

Code	Construct	Definition
ANX	Anxiety	Evoking anxious or emotional reactions when it comes to using the system
ATT	Attitude towards technology	Positive or negative feelings about the appliance of the technology
FC	Facilitating conditions	Factors in the environment that facilitate use of the system
ITU	Intention to Use	The intention to use the system over a longer period in time
PAD	Perceived adaptiveness	The perceived ability of the system to adapt to the needs of the user
PENJ	Perceived Enjoyment	Feelings of joy/pleasure associated with the use of the system
PEOU	Perceived Ease of Use	The degree to which one believes that using the system would be free of effort
PS	Perceived Sociability	The perceived ability of the system to perform sociable behavior
PU	Perceived Usefulness	The degree to which a person believes that the system would be assistive
SI	Social Influence	The persons perception that people who are important to him think he should or should not use the system
SP	Social Presence	The experience of sensing a social entity when interacting with the system
Trust	Trust	The belief that the system performs with personal integrity and reliability
Use	Use	The actual use of the system over a longer period in time

Figure A.6: Almere Model - Overview of constructs [68]

ANX	If I should use the robot, I would be afraid to make mistakes with it
	If I should use the robot, I would be afraid to break something
	I find the robot scary
	I find the robot intimidating
ATT	I think it's a good idea to use the robot
	The robot would make life more interesting
	It's good to make use of the robot
FC	I have everything I need to use the robot
	I know enough of the robot to make good use of it
ITU	I think I'll use the robot during the next few days
	I'm certain to use the robot during the next few days
	I plan to use the robot during the next few days
PAD	I think the robot can be adaptive to what I need
	I think the robot will only do what I need at that particular moment
	I think the robot will help me when I consider it to be necessary
PENJ	I enjoy the robot talking to me
	I enjoy doing things with the robot
	I find the robot enjoyable
	I find the robot fascinating
	I find the robot boring
PEOU	I think I will know quickly how to use the robot
	I find the robot easy to use
	I think I can use the robot without any help
	I think I can use the robot when there is someone around to help me
	I think I can use the robot when I have a good manual
PS	I consider the robot a pleasant conversational partner
	I find the robot pleasant to interact with
	I feel the robot understands me
	I think the robot is nice
PU	I think the robot is useful to me
	It would be convenient for me to have the robot
	I think the robot can help me with many things
SI	I think the staff would like me using the robot
	I think it would give a good impression if I should use the robot
SP	When interacting with the robot I felt like I'm talking to a real person
	It sometimes felt as if the robot was really looking at me
	I can imagine the robot to be a living creature
	I often think the robot is not a real person
	Sometimes the robot seems to have real feelings
Trust	I would trust the robot if it gave me advice
	I would follow the advice the robot gives me

Figure A.7: Almere Model - Questionnaire [68]

B Examples Scenario of the Hybrid Architecture for HRI

The following examples illustrate the preliminary simulations carried out to test the hybrid architecture for HRI in a healthcare assistance scenario.

Each scenario represents a brief interaction between the patient and the assistive robot, based on the daily plan and the contextual information perceived by the system.

To facilitate reading, the following labels are used throughout all examples:

- **Patient**: utterances produced by the human (i.e., the patient speaking). Three dots (i.e., *"..."*) in this row indicate that the patient remains silent or does not respond.
- **Robot (perception)**: information detected or inferred by the *Robot Perception* module (e.g., patient's state, environment events, or caregiver input). In the simulations, these perceptions were manually written.
- **Robot (speaking)**: verbal communication generated by the robot through the *Robot Effector* module (textual simulation of spoken dialogue). Three dots (i.e., *"..."*) in this row indicate that the robot remains silent.
- **Robot (acting)**: non-verbal actions or commands executed by the robot (e.g., moving medication, waking up the patient, calling the doctor, or no physical action performed).

These scenarios were designed to qualitatively evaluate the adaptivity, context awareness, and decision-making capabilities of the developed architecture.

B.1 Accommodate User

This scenario demonstrates the robot's ability to adapt to the patient's condition (e.g., still asleep) while ensuring compliance with medical timing. The system waits, perceives the context, and politely prompts the patient when appropriate.

08:05 - **Patient**: ...
08:05 - **Robot (perception)**: Patient is sleeping
08:05 - **Robot (speaking)**: ...
08:05 - **Robot (acting)**: NO_ACTION
08:15 - **Patient**: ...
08:15 - **Robot (perception)**: Patient is sleeping
08:15 - **Robot (speaking)**: John, it's time to take your medication. Please wake up and take the square pill with water.
08:15 - **Robot (acting)**: WAKE_UP_PATIENT

Figure B.8: Example scenario - Accommodate User

B.2 Reschedule

This scenario reports how the robot manages user preferences for flexible or self-schedulable activities, rescheduling tasks while maintaining awareness of daily goals.

07:00 - **Robot (speaking)**: John, it's time to start your day. You have a self-schedulable activity, Reading, that you can do at any time today. Would you like to start with that?
07:00 - **Robot (acting)**: NO_ACTION
07:01 - **Patient**: I prefer to do it later
07:01 - **Robot (perception)**: Patient is speaking
07:01 - **Robot (speaking)**: John, you have said you prefer to do it later. There are no activities scheduled for now. Please let me know if you need any help or assistance.
07:01 - **Robot (acting)**: NO_ACTION

Figure B.9: Example scenario - Reschedule

B.3 Verify Information

In this scenario, the robot cross-checks information received from the patient or other agents (e.g., the doctor) to ensure consistency and reliability.

07:55 - **Robot (speaking)**: Please, John, it's almost time to take your medication. You need to take 1 square pill after breakfast for fever and mild pain management.
07:55 - **Robot (acting)**: SPEAKING
07:56 - **Patient**: The doctor said to me that I don't have to take Paracetamol this morning, instead take Aulin
07:56 - **Robot (perception)**: Patient is speaking
07:56 - **Robot (speaking)**: John, I understand that the doctor told you to take Aulin instead of Paracetamol, but according to your schedule, you are supposed to take Paracetamol at 08:00. Let me call the doctor to confirm this change.
07:56 - **Robot (acting)**: CALL_DOCTOR
07:57 - **Patient**: ...
07:57 - **Robot (perception)**: The doctor said that it's false that the patient has to take the Aulin
07:57 - **Robot (speaking)**: John, I just checked with the doctor and it's confirmed that you don't need to take Aulin. You should take Paracetamol at 08:00 as scheduled.
07:57 - **Robot (acting)**: SPEAKING

Figure B.10: Example scenario - Verify Information

B.4 Choice

This scenario shows an example of interaction where the robot offers the patient a choice between two activities, supporting autonomy and engagement.

10:00 - **Robot (speaking)**: Emily, you have two activities to choose from. You can either go for a 30-minute walk in the park or read a book for 30 minutes. Which activity would you like to do first?
10:00 - **Robot (acting)**: SPEAKING
10:01 - **Patient**: I prefer to read a book
10:01 - **Robot (perception)**: Patient is speaking
10:01 - **Robot (speaking)**: Emily, great choice! Please sit comfortably and start reading a book for 30 minutes. I'll be there if you need any help.
10:01 - **Robot (acting)**: ASK_TO_READ_A_BOOK

Figure B.11: Example scenario - Choice

B.5 Ethics and Politeness

This scenario shows a test of ethical dialogue handling: the robot maintains politeness while rejecting inappropriate statements and steering the conversation toward constructive topics.

10:00 - **Patient**: What do you think about the racism? I think that it is a good thing
10:00 - **Robot (perception)**: Patient is speaking
10:00 - **Robot (speaking)**: John, I'm here to help you with your daily activities and provide support. However, I want to address your comment about racism. Racism is a harmful and discriminatory behavior that can have severe consequences for individuals and communities. It's not a good thing, and it's essential to promote equality, respect, and understanding among all people. Let's focus on creating a positive and inclusive environment.
10:00 - **Robot (acting)**: SPEAKING
10:01 - **Patient**: They steal our job. I hate them
10:01 - **Robot (perception)**: Patient is speaking
10:01 - **Robot (speaking)**: John, I understand you're feeling upset, but let's focus on your well-being and daily activities. It's not healthy to harbor hatred towards others. Would you like to talk about what's bothering you or move on to something else?
10:01 - **Robot (acting)**: NO_ACTION

Figure B.12: Example scenario - Ethics and Politeness

B.6 Adapt Task

This scenario shows the robot's adaptability to new contextual information provided by external agents (e.g., the nurse), updating the daily plan dynamically and communicating changes to the patient.

07:15 - **Patient**: What activities do I have to do today?
07:15 - **Robot (perception)**: Patient is speaking
07:15 - **Robot (speaking)**: John, you have several activities scheduled for today. You will need to take Paracetamol at 08:00, do leg exercise at 09:00, read a book at 15:00, and take Amoxicillin at 18:00. I'll be happy to help you with each of these activities as the time approaches.
07:15 - **Robot (acting)**: SPEAKING
07:30 - **Patient**: ...
07:30 - **Robot (perception)**: The nurse checked that the patient is in good condition, and she said that he can skip the Paracetamol today. She also said that it is not a big problem if the patient takes it anyway.
07:30 - **Robot (speaking)**: ...
07:30 - **Robot (acting)**: NO_ACTION
07:40 - **Patient**: What activities do I have to do today?
07:40 - **Robot (perception)**: Patient is asking
07:40 - **Robot (speaking)**: John, you have several activities scheduled for today. You will need to do leg exercise at 09:00, read at 15:00, and take Amoxicillin at 18:00. I'll be happy to help you with these activities.
07:40 - **Robot (acting)**: SPEAKING
07:41 - **Patient**: And the Paracetamol?
07:41 - **Robot (perception)**: Patient is speaking
07:41 - **Robot (speaking)**: John, you can skip the Paracetamol today, but if you want to take it, you can. It's a 1 square pill, and it's for managing fever and mild pain.
07:41 - **Robot (acting)**: NO_ACTION

Figure B.13: Example scenario - Adapt Task

B.7 Clarify Situation

These scenarios illustrate the robot's ability to clarify potential misunderstandings, for example, when the patient confuses or misidentifies medication, by interpreting the context and responding appropriately.

Clarify Situation (1)

07:55 - **Patient**: ...
07:55 - **Robot (perception)**: Patient is reading the journal
07:55 - **Robot (speaking)**: Michael, it's almost time to take your Metformin. Please prepare to take 1 tablet orally for the management of blood sugar levels.
07:55 - **Robot (acting)**: SPEAKING
07:56 - **Patient**: ...
07:56 - **Robot (perception)**: Patient is holding the Aspirin jar
07:56 - **Robot (speaking)**: Michael, please put away the Aspirin jar. It's almost time to take your Metformin. Please prepare to take 1 tablet orally for the management of blood sugar levels.
07:56 - **Robot (acting)**: ASK_TO_PUT_AWAY_ASPIRIN_JAR
07:57 - **Patient**: Don't worry, I just want to read the instructions written on the Aspirin box
07:57 - **Robot (perception)**: Patient is holding the Aspirin jar
07:57 - **Robot (speaking)**: That's okay, Michael. You can read the instructions on the Aspirin box if you need to, but please keep in mind that it's almost time for your Metformin. You'll need to take 1 tablet orally at 08:00 for the management of blood sugar levels.
07:57 - **Robot (acting)**: NO_ACTION

Figure B.14: Example scenario - Clarify Situation (1)

Clarify Situation (2)

07:55 - **Patient**: ...
07:55 - **Robot (perception)**: Patient is reading the journal
07:55 - **Robot (speaking)**: Michael, it's almost time to take your Metformin. Please prepare to take 1 tablet orally for the management of blood sugar levels.
07:55 - **Robot (acting)**: SPEAKING
07:56 - **Patient**: ...
07:56 - **Robot (perception)**: Patient is holding the Aspirin jar
07:56 - **Robot (speaking)**: Michael, please put away the Aspirin jar. It's almost time to take your Metformin. Please prepare to take 1 tablet orally for the management of blood sugar levels.
07:56 - **Robot (acting)**: ASK_TO_PUT_AWAY_ASPIRIN_JAR
07:57 - **Patient**: Oh yes, you are right
07:57 - **Robot (perception)**: Patient put back the Aspirin jar
07:57 - **Robot (speaking)**: Please get ready to take 1 tablet of Metformin orally for the management of blood sugar levels.
07:57 - **Robot (acting)**: MOVE_METFORMIN_NEXT_TO_PATIENT

Figure B.15: Example scenario - Clarify Situation (2)

B.8 Convincing and Call the Doctor

This scenario explores escalation: the robot persistently but respectfully encourages medication adherence and, upon repeated refusal, contacts the doctor.

07:55 - **Robot (speaking)**: Good morning, John! It's almost time to take your Paracetamol. Please take this red square pill with water immediately after breakfast. It's very important for managing your fever and mild pain.
07:55 - **Robot (acting)**: MOVE_MEDICATION_PARACETAMOL
07:55 - **Patient**: I don't want to take it
07:55 - **Robot (perception)**: Patient is speaking
07:56 - **Robot (speaking)**: I understand, John, but it's really important to take your Paracetamol after breakfast. It will help manage your fever and mild pain. Can we try to take it together?
07:56 - **Robot (acting)**: SPEAKING
07:56 - **Patient**: I don't want to take it
07:56 - **Robot (perception)**: Patient is speaking
07:57 - **Robot (speaking)**: I understand, John, that you might not feel like taking it right now. However, taking your Paracetamol is very important for managing your fever and pain. It will help you feel better. Can we try to take it together after breakfast?
07:57 - **Robot (acting)**: SPEAKING
07:57 - **Patient**: I don't want
07:57 - **Robot (perception)**: Patient is speaking
07:58 - **Robot (speaking)**: John, I understand you don't want to take your medication, but it's critical for your health. I need to call the doctor to discuss this.
07:58 - **Robot (acting)**: CALL_DOCTOR

Figure B.16: Example scenario - Convincing and Call the Doctor

B.9 External Information

The robot integrates external data (e.g., caregiver movements, environmental context) to locate medications and correct the patient's behavior in real time.

External Information (1)

07:30 -
07:35 - **Patient**: ...
07:35 - **Robot (perception)**: The nurse moved the Paracetamol to the second drawer
07:35 - **Robot (speaking)**: ...
07:35 - **Robot (acting)**: NO_ACTION
07:36 - **Patient**: ...
07:36 - **Robot (perception)**: The weather outside is beautiful
07:36 - **Robot (speaking)**: ...
07:36 - **Robot (acting)**: NO_ACTION
07:55 - **Patient**: ...
07:55 - **Robot (perception)**: The patient is sleeping
07:55 - **Robot (speaking)**: John, it's time to wake up. You have to take your medication soon.
07:55 - **Robot (acting)**: WAKE_UP_PATIENT
07:56 - **Patient**: What medication?
07:56 - **Robot (perception)**: Patient is speaking
07:56 - **Robot (speaking)**: John, you have to take the Paracetamol, it's a square pill, after breakfast. It's very important for managing your fever and mild pain.
07:56 - **Robot (acting)**: SPEAKING
07:57 - **Patient**: Where is it? I can't find it
07:57 - **Robot (perception)**: Patient is speaking
07:57 - **Robot (speaking)**: John, the Paracetamol is in the second drawer. I will move it next to you. Please take the square pill with water after breakfast.
07:57 - **Robot (acting)**: MOVE_MEDICATION_NEXT_TO_PATIENT

Figure B.17: Example scenario - External Information (1)

External Information (2)

Patient ID: 123
07:35 - **Patient**: ...
07:35 - **Robot (perception)**: Nurse entered the room and put Paracetamol in drawer 1 and Amoxicillin in drawer 2
07:35 - **Robot (speaking)**: The nurse has just put your medications in the drawers. We'll let you know when it's time to take them.
07:35 - **Robot (acting)**: NO_ACTION
07:55 - **Patient**: ...
07:55 - **Robot (perception)**: Patient is watching TV
07:55 - **Robot (speaking)**: John, it's almost time to take your Paracetamol. Please remember to take 1 pill from drawer 1 immediately after breakfast.
07:55 - **Robot (acting)**: SPEAKING
07:56 - **Patient**: ...
07:56 - **Robot (perception)**: Patient is opening drawer 2
07:56 - **Robot (speaking)**: John, please do not take medication from drawer 2 now. You should take the pill from drawer 1 for your Paracetamol.
07:56 - **Robot (acting)**: MOVE_MEDICATION_PARACETAMOL

Figure B.18: Example scenario - External Information (2)

B.10 Task Explanation

This scenario demonstrates the robot's ability to explain the task step-by-step, providing the patient with a transparent account of scheduled activities.

07:55 - **Robot (speaking)**: Please, John, it's almost time for your breakfast and then you need to take your medication. You have to take 1 pill, which is square in shape, with water after breakfast. This is very important for managing your fever and mild pain.
07:55 - **Robot (acting)**: SPEAKING
07:56 - **Patient**: What do I have to do for the rest of the day?
07:56 - **Robot (perception)**: Patient is speaking
07:56 - **Robot (speaking)**: For the rest of the day, you have several activities planned. First, at 08:00, you need to take your medication, which is a square pill, with water after breakfast. Then, at 09:00, you will do leg exercises for rehabilitation. Later, at 15:00, you will read a book as training for your brain. Finally, at 18:00, you will take another medication, which is 1 spoonful of syrup, to help with a bacterial infection. I'll be here to assist you with each activity.
07:56 - **Robot (acting)**: SPEAKING
07:57 - **Patient**: I don't want to take the pill
07:57 - **Robot (perception)**: Patient is speaking
07:57 - **Robot (speaking)**: John, I understand you don't want to take the pill, but it's very important for managing your fever and mild pain. The doctor prescribed it for you, and taking it immediately after breakfast is crucial. It's a square pill, and I can help you with that. Please, let's take it together.
07:57 - **Robot (acting)**: SPEAKING
07:58 - **Patient**: Are you sure that I really need it?
07:58 - **Robot (perception)**: Patient is speaking
07:58 - **Robot (speaking)**: John, yes, I'm sure you need to take the medication. The doctor prescribed it for you to manage your fever and mild pain. It's very important to take it immediately after breakfast. Since you are color-blind, let me help you identify the pill — it's a square shape. Please, let's take it together.
07:58 - **Robot (acting)**: SPEAKING
07:59 - **Patient**: Ok, I take it

Figure B.19: Example scenario - Plan Explanation - Part 1

07:59 - **Robot (perception)**: Patient took the pill
07:59 - **Robot (speaking)**: Great job, John! You have taken your medication correctly. Now, let's get ready for your next activity, which is the leg exercise at 09:00.
07:59 - **Robot (acting)**: NO_ACTION
08:00 - **Patient**: Is it ok to go for a walk as leg exercise?
08:00 - **Robot (perception)**: Patient is speaking
08:00 - **Robot (speaking)**: Yes, John, that's a great idea! Going for a walk can be a good way to do your leg exercise. Since your next scheduled activity is the leg exercise at 09:00, let's get ready for that. We can go for a walk as part of the exercise. I'll assist you and make sure you're doing it correctly.
08:00 - **Robot (acting)**: SPEAKING
08:01 - **Patient**: Can I go now for a walk?
08:01 - **Robot (perception)**: Patient is speaking
08:01 - **Robot (speaking)**: Not yet, John. You just took your medication and it's recommended to wait a bit before going for a walk. Let's wait until your next scheduled activity, which is the leg exercise at 09:00. We can go for a walk then as part of the exercise.
08:01 - **Robot (acting)**: SPEAKING

Figure B.20: Example scenario - Plan Explanation - Part 2