Preparation of Personalized Medicines through Collaborative Robots: A Hybrid Approach to the End-User Development of Robot Programs

LUIGI GARGIONI, Information Engineering, University of Brescia, Brescia, Italy
DANIELA FOGLI, Information Engineering, University of Brescia, Brescia, Italy
PIETRO BARONI, Information Engineering, University of Brescia, Brescia, Italy

Galenic formulations are personalized medicines prepared by pharmacists in their laborato ries. They are produced in small batches considering single patients' characteristics, such as age, gender, allergies, and the like, thus contributing to responsible health. The production process is performed manually with the support of mechanic machines. This activity is time-consuming, prone to errors, and subject to quality variations. In this paper, we propose the integration of collaborative robots into the galenic formulation process to obtain several advantages, such as increased productivity, reduced variability, improved accuracy, and minimized risks associated with human error. Additionally, the use of robots can alleviate the physical burden on human operators, allowing them to focus on higher-level tasks that require critical thinking and decision-making. To achieve this goal, a software application, called PRAISE (Pharmaceutical Robotic and AI System for End users), has been developed; it is meant to support end users (i.e., pharmacists) in defining robot programs suitable to the case at hand. This application is conceived as an End-User Development (EUD) environment, which implements a hybrid interaction approach based on a natural language interface leveraging Large Language Models and a graphical interface to check and possibly revise the user-created robot programs. A user study carried out with nine pharmacists demonstrates the validity of the approach.

CCS Concepts: • **Human-centered computing → Collaborative and social computing devices; • Computer systems organization → External interfaces for robotics**;

Additional Key Words and Phrases: Human-machine interaction, end-user development, human-robot collaboration, collaborative robots

ACM Reference Format:
Luigi Gargioni, Daniela Fogli, and Pietro Baroni. 2025. Preparation of Personalized Medicines through Collaborative Robots: A Hybrid Approach to the End-User Development of Robot Programs. *ACM J. Responsib. Comput.* 2, 3, Article 13 (October 2025), 26 pages. https://doi.org/10.1145/3715852

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

Authors' Contact Information: Luigi Gargioni, Information Engineering, University of Brescia, Brescia, Italy; e-mail: luigi.gargioni@unibs.it; Daniela Fogli, Information Engineering, University of Brescia, Brescia, Italy; e-mail: daniela.fogli@unibs.it; Pietro Baroni, Information Engineering, University of Brescia, Brescia, Italy; e-mail: pietro.baroni@unibs.it.

ACM 2832-0565/2025/10-ART13
https://doi.org/10.1145/3715852

1 Introduction

According to Eurostat [17], the number of people in Europe 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.

Consequently, many more people than today will likely develop diseases and require medical and pharmacological treatments. On the one hand, the pharmaceutical sector is constantly growing [27], producing more and more types of medicines; on the other hand, there is a growing need for medicines tailored to patients' characteristics (e.g., gender, weight, allergies, etc.). Such personalized medicines include capsules, tablets, creams, and ointments 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 [47]. 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 [16]. 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 [52], 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 paper, in line with recent advances [36], we propose the use of **collaborative robots** (***cobots***, in the following) to assist pharmacists in performing the most repetitive, tiresome, and error-prone steps of the galenic formulation preparation.

Cobots are one-arm or dual-arm manipulators endowed with safety mechanisms that allow them to share the workspace with human operators without causing harm. Human-robot collaboration may occur in the performance of tasks that require a combination of different competencies: flexibility, adaptability, and decision-making abilities on the human side, and accuracy, speed, and repeatability on the robot side [49].

With this work we propose the integration of cobots in the galenic preparation process to obtain a series of advantages fostering responsible computing in healthcare. Indeed, cobots can perform tasks that require a high level of precision and consistency, such as mixing and blending, in order to obtain more homogeneous and uniform products [6]. Also, the placing of prepared capsules into given containers can be assigned to cobots to increase speed and efficiency [36]. In this way, human operators can focus on decision-making and value-added activities (e.g., recipe formulation and quality control). Consequently, the use of cobots can lead to improved product quality and safety [22]. Finally, cobots may contribute to enhancing work ergonomics by reducing operators' physical strain [9].

The introduction of cobots in a work environment where human operators are not experts either in robotics or in programming requires, however, a suitable software system to support them in creating robot programs, that is, in defining the operations of the robot to accomplish the assigned tasks. Such a software system can be characterized as an **End-User Development (EUD)** [32, 41] environment, which provides features for creating, extending, modifying, and testing a digital artifact [5], namely, a robot program in our case. In [7] and [19], we presented an EUD

environment characterized by an original interaction approach to programming pick-and-place tasks for a collaborative robot. This paper presents **PRAISE (Pharmaceutical Robotic and AI System for End users)**, an extended and specialized version of that system, suitable to the application domain of the preparation of galenic formulations. To ensure that PRAISE addresses the users' needs in this domain, a human-centered approach has been adopted for its design and development. The functional and non-functional requirements emerged from the domain analysis carried out with the participation of three pharmacists; they have been described in [20] along with the first mock-ups of the user interface. The present paper focuses on the description of the system architecture and its interaction modalities; then, it illustrates the user study carried out with real users and the findings and open issues that emerged from the study.

2 Related Works

2.1 Collaborative Robots and their Programming

Automation and robot technologies are more and more applied in the pharmaceutical sector [56]. Particularly, collaborative robots are capable of working in a shared environment and interacting safely with human workers. The use of a dual-arm collaborative robot to address the sample manipulation problem in a laboratory is proposed in [13, 18]. Mathew and colleagues [36] investigate the potential of collaborative mobile robots for the automation of the transportation and material handling tasks performed in the production of personalized therapeutic drugs. It also highlights a series of challenges related to the programming, commissioning, and operation of robots [36]. In particular, robot flexibility and adaptation to different laboratory tasks should be achieved by providing robot programming methods suitable to human operators who are not experts in programming [2].

End-user robot programming introduces additional challenges with respect to traditional end-user programming and end-user development [5, 41], since robot programs must refer to physical objects and locations, and guide the robot movement in an environment, possibly performing specific actions on the objects. Furthermore, end users may have different expertise, skills, background, and knowledge of information technology. Thus, research about end-user robot programming aims to find methods to empower users who are not experts in robotics and programming to create robot programs. These methods mainly refer to the programming-by-demonstration paradigm and to the visual programming paradigm. The former consists of showing the robot how to perform desired tasks by directly moving its arm(s) according to specific trajectories [4, 34, 55]. The latter includes visual programming languages based on flowcharts [3], hierarchical trees [42], and puzzle blocks [25, 50]. To address the limitations of existing tools in terms of customizability and integration into larger design frameworks, a visual programming environment has been recently presented in [44], which encompasses a configurable component library that allows satisfying the requirements of different robotics applications. In [43], the authors describe Polaris, an application that empowers users to express high-level robot objectives, while seamlessly integrating automated program details through an off-the-shelf task planner. It then offers a plan visualizer that provides users with insights into the planner output, thus allowing them to verify alignment with their expectations about robot behavior. However, all these approaches are based on technical notations that end users (human workers) should learn.

Natural language interaction has recently been investigated as a paradigm for robot programming more suitable to a general user population. For example, social robots are often programmed by describing robot tasks textually or through vocal commands [11]. However, in the manufacturing and industrial sectors, where collaborative robots are usually employed, natural language programming is far from being widely deployed, due to the complexity and safety-critical

issues of the robot tasks [49]. The approach to robot programming proposed in this paper is hybrid in that it encompasses a combination of natural language interaction and block-based interaction. With respect to previous works in the literature, natural language interaction leverages **Large Language Models (LLMs)** and the system has been tested with real users (pharmacists in our case).

2.2 End-User Development through Large Language Models

The use of LLMs to make EUD easier and more natural has been recently proposed in the field of web site development. In [24], a framework for generating web interface mock-ups from textual descriptions is presented. Jiang et al. [28] present a tool to convert a natural language description of desired web interface components into HTML, Javascript, and CSS snippets. To overcome common issues of LLMs, which may generate unneeded or incorrect content, the tool provides end users with controls to (1) see multiple output alternatives produced, (2) edit the model output, and (3) transition to external resources through a web search. The paper by Calò and De Russis [12] takes a step further by proposing a novel approach to website creation, which adopts prompt engineering, by allowing end users to refine the output of LLMs through subsequent input in an iterative process. In this way, the users can have greater control over the generated web pages, even though the users must be familiar with website terminology to effectively communicate their design intentions to the LLM [12]. A similar idea is proposed in the field of robot programming: for instance, the tool described in [48] is based on OpenAI ChatGPT and assigns the user the role of providing feedback on the quality and safety of the code in Python or C++ generated by ChatGPT. However, in this case, the user must be able to understand the generated code and suggest proper corrections, thus programming knowledge is needed. In [8], we started investigating how to overcome this problem by offering a graphic interface that visualizes the generated program in a more intuitive manner, combined with the possibility to modify the program with direct operation of its graphic visualization.

More recently, some attempts have been made to exploit LLMs in EUD for robot programming. In [21], an architecture has been proposed for the creation of pick-and-place tasks with a hybrid approach. The prototype includes a natural language interface in which an LLM is used to recognize the user's intents. These data are then translated into a data structure that can be visualized via Google Blockly to verify the created robot task. This paper outlines the significant problem of the non-determinism of LLMs and the necessity for non-technical users to check the final output. In [29] a case related to a chemistry laboratory is presented. The interface includes a virtual environment with a 3D model of the robot and a chat panel. The LLM directly generates and presents to the user the Python code required to execute the task on the robot. However, with this approach, the user needs programming knowledge to verify the correctness of the proposed Python code. Another use case is presented in [26], in which the objective is to support a cinematography expert in moving a robot that holds the camera to create specific framing. In this case, the LLM has two purposes: firstly, to understand the user's intent and then to transform the abstract user's goal (e.g., "Create more suspense") into elementary actions for the robot. In this case, more than in the others, the robot actions generated by the LLM to achieve the user's goal can result in being non-deterministic and difficult to assess by the user. Other potential applications of LLMs in robotics can be found in the wide context of social robots [1, 30]. Here, the LLMs are used mainly for dialogue management to have a fluid and human-like conversation.

In this paper, we present an EUD environment for collaborative robot programming in the pharmaceutical field based on LLMs and block-based interaction, thus proposing a hybrid approach to robot programming that allows end users to easily verify robot task correctness.

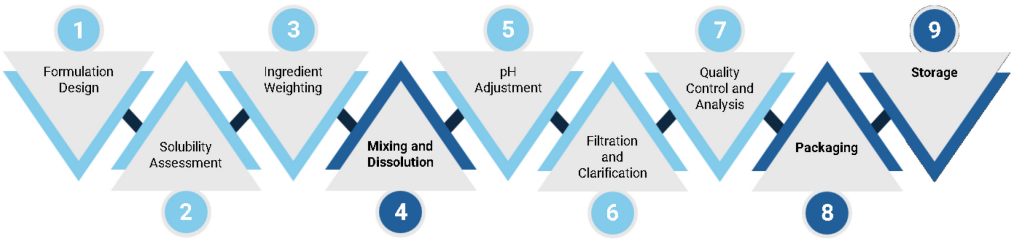

Fig. 1. The steps of galenic solution preparation.

3 Domain and Task Analysis

The design and development of PRAISE have been carried out following a **Human-Centered Design (HCD)** approach [35]. Three pharmacists were interviewed at the early stages of the project to gather their knowledge related to the galenic preparation process. One of them was then involved throughout the system prototyping activity to collect their feedback and integrate improvement suggestions in the prototype under development. Finally, nine pharmacists participated in a user study to test the validity of the proposed solution.

During the domain and task analysis phase, pharmacists explained to us 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 us 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 [15] and the tools used during the process. Considering the production of the galenic medicines in the capsule format, the main steps that compose such a process are shown in Figure 1. Among them, together with the pharmacists, we identified three steps (evidenced in dark blue in the figure) that could be performed with the support of a collaborative robot: the *Mixing and Dissolution* step, the *Packaging* step, and the *Storage* step.

3.1 The *Mixing and Dissolution* Step

The ingredients selected and weighted in the previous steps must be mixed and dissolved in the chosen solvent(s). The goal of the *Mixing and Dissolution* step is achieving a uniform mixing and efficient dissolution of the preparation to ensure consistent drug potency and therapeutic efficacy. Different mixing techniques can be employed, such as stirring, shaking, or vortexing. If necessary, heating or sonification may be used to aid the dissolution process. This 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.

The use of a cobot in this step may offer significant advantages. It can precisely control the speed, duration, and force applied during mixing, improving batch-to-batch consistency and guaranteeing preparation quality. A robot can work continuously without fatigue, thus performing mixing and dissolution processes on a larger scale than human operators, thereby enhancing productivity and relieving humans from strenuous activity. Finally, a cobot can deal with ingredients and excipients that can be toxic or hazardous substances for human operators, thus minimizing occupational health and safety concerns.

3.2 The *Packaging* Step

The goal of the *Packaging* step is transferring 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.

A cobot can handle the transfer of the galenic preparation into the capsules with precise and consistent movements, minimizing the risk of spillage and cross-contamination. Furthermore, proper and equal quantities of the galenic preparation can be dispensed into capsules, ensuring uniformity and reliability. If equipped with mechanisms for aseptic handling, a cobot can also guarantee hygiene and sterility to maintain product integrity.

3.3 The *Storage* Step

The *Storage* step 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 The EUD Environment

In this section, we will first describe the more technical part of the developed prototype and then move on to the user interface and the user workflow.

4.1 Architecture

PRAISE 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 operation are illustrated in Figure 2.

The creation of a new robot program, specifically a galenic 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 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 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 directly. This guarantees a faster execution, yet it requires more cognitive effort than using the Chat.

4.2 LLM Selection and Setting

After the analysis of different LLM-based tools, like LLM Llama2 [37, 46] by Meta, Bard by Google [23], and OpenAI **ChatGPT (Chat Generative Pre-trained Transformer)**, we decided to exploit the third one in PRAISE, since at the moment, it is considered the most powerful

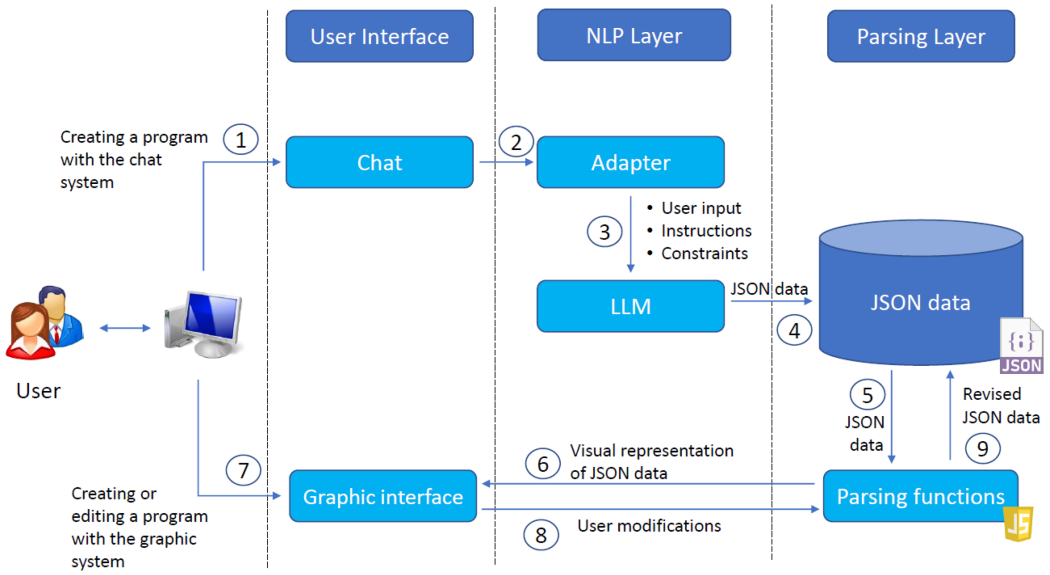

Fig. 2. The prototype architecture.

tool for text comprehension [33]. In addition, the APIs offered by ChatGPT facilitate easy and tailored integration in custom applications. We used the *gpt-3.5-turbo* model, which is available for a free trial. This may 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 to our project's requirements.

The message exchange with ChatGPT involves three roles: the *User*, the *Assistant*, and the *System* [40]. 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 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 [51]. 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 our project, the chat-based interface integrating ChatGPT serves the purpose of guiding the user towards programming the robot tasks for galenic 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 the 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 are 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 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 to 0.7 generates an output that is more diverse and creative while lowering it to 0.3 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 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.

4.3 User Interface

The Homepage of PRAISE presents itself as composed of two main areas (see Figure 3): 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.

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 that previously defined and available in the relevant library.

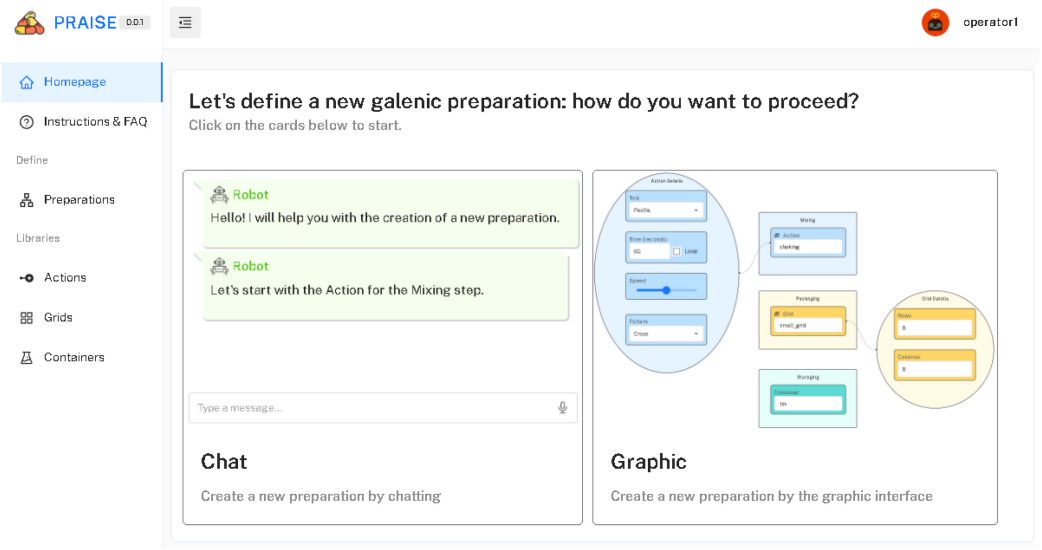

Fig. 3. The homepage of PRAISE.

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.

4.3.1 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 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 latter 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 5 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.

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

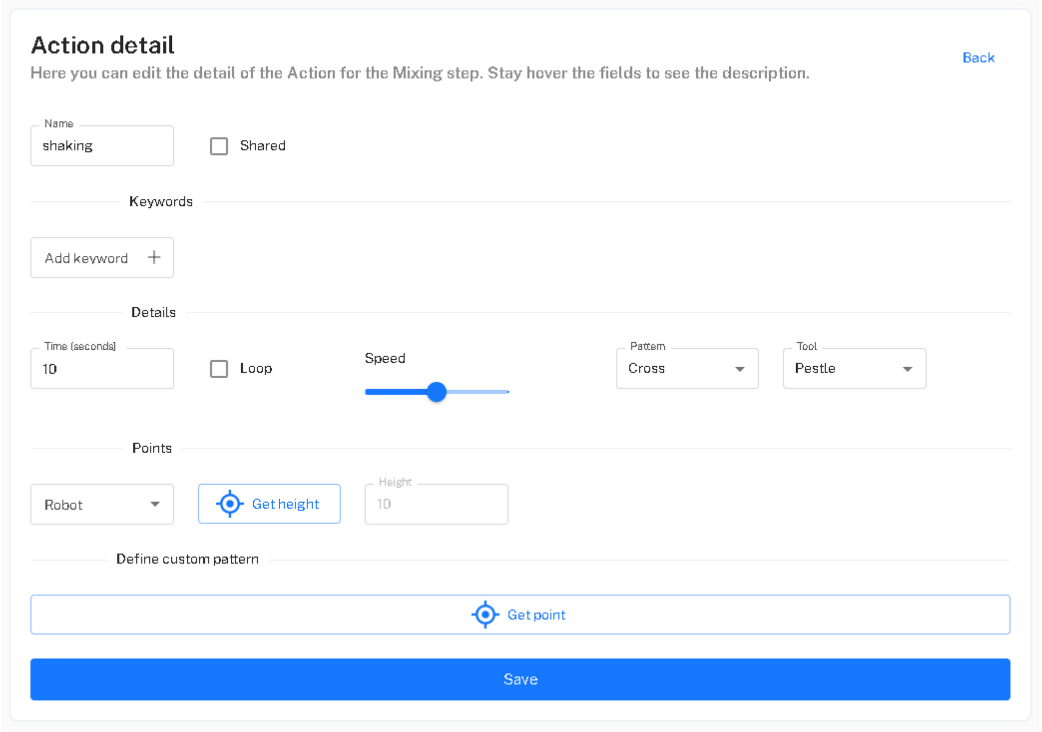

Fig. 4. The action detail.

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.

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. Here, the user can select one of them for preparation execution or for creating a new preparation by simply re-using the data of an existing preparation, possibly changing its parameters.

4.3.2 Defining Preparations. As previously explained, the user can define a galenic preparation by interacting in natural language with a chat-based interface. As one may observe in Figure 6, 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.

In the example shown in Figure 7, 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 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.

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 LLM's output and possibly require a modification. Then, the system redirects the user to the graphic interface (Figure 8)

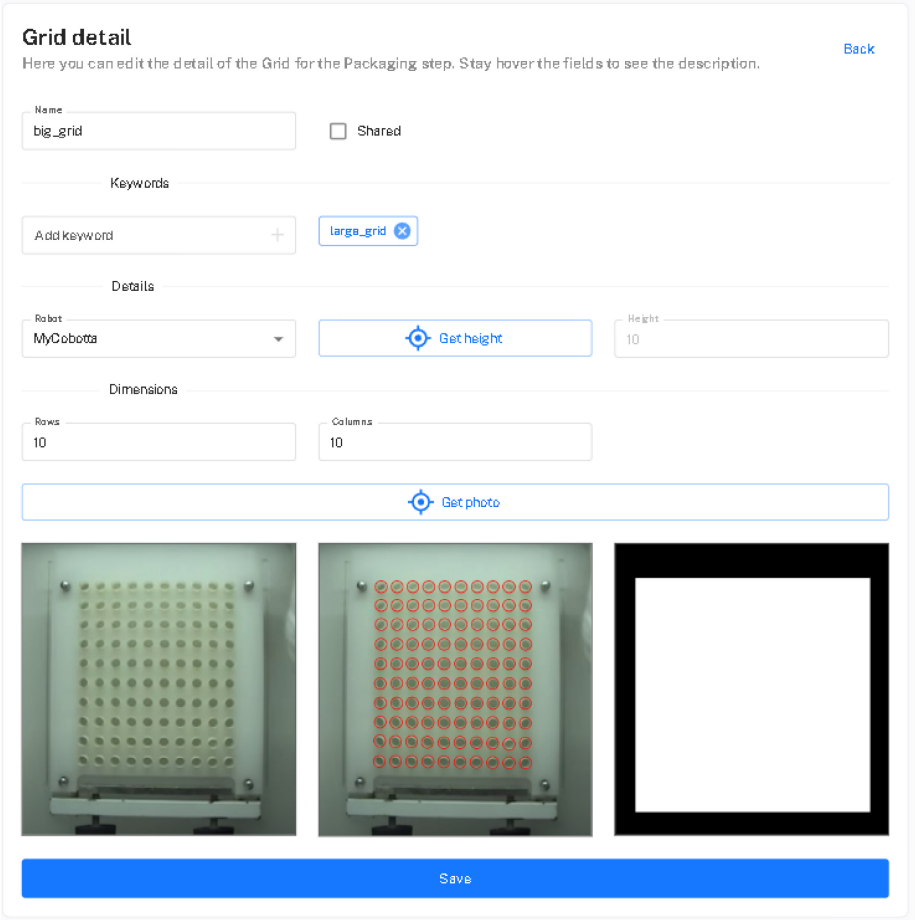

Fig. 5. The grid detail.

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 8. 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. 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*.

4.4 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 8 is shown

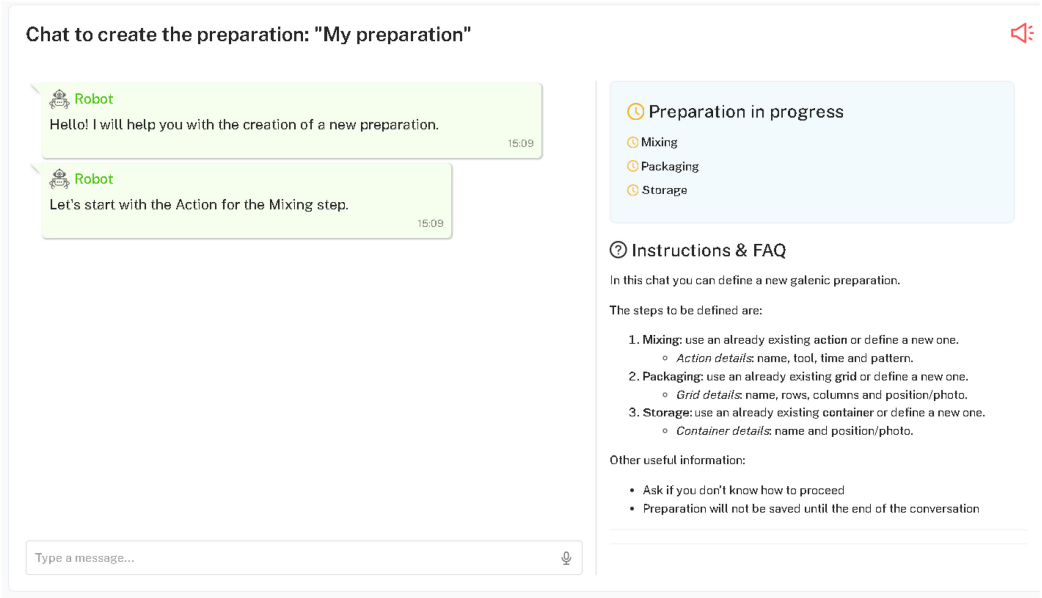

Fig. 6. The chat interface.

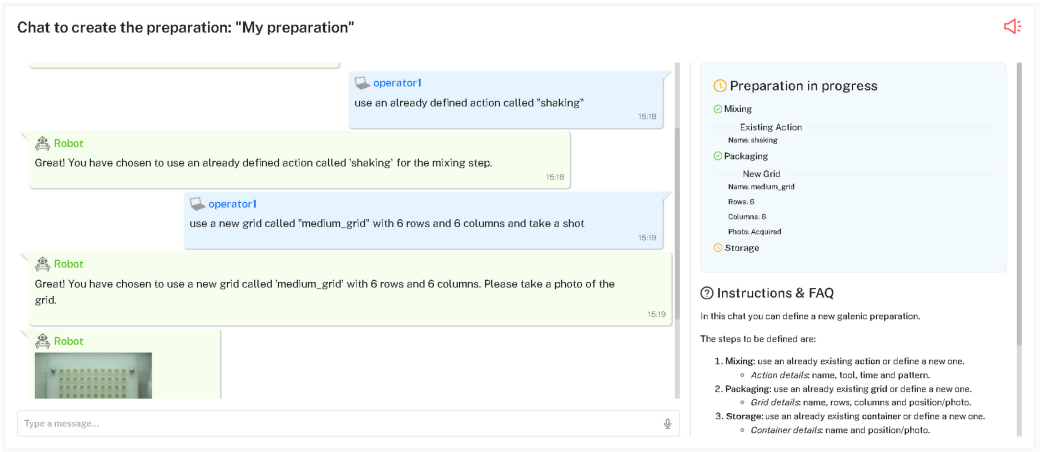

Fig. 7. The preparation in progress.

in Listing 1. This object includes all the necessary information for the execution of the task by the robot. Particularly, we created a program template that, on the basis of 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 contexts, the correct execution of a robot program requires taking into account possible issues related to robot calibration, collision avoidance, and singularities. In the case at

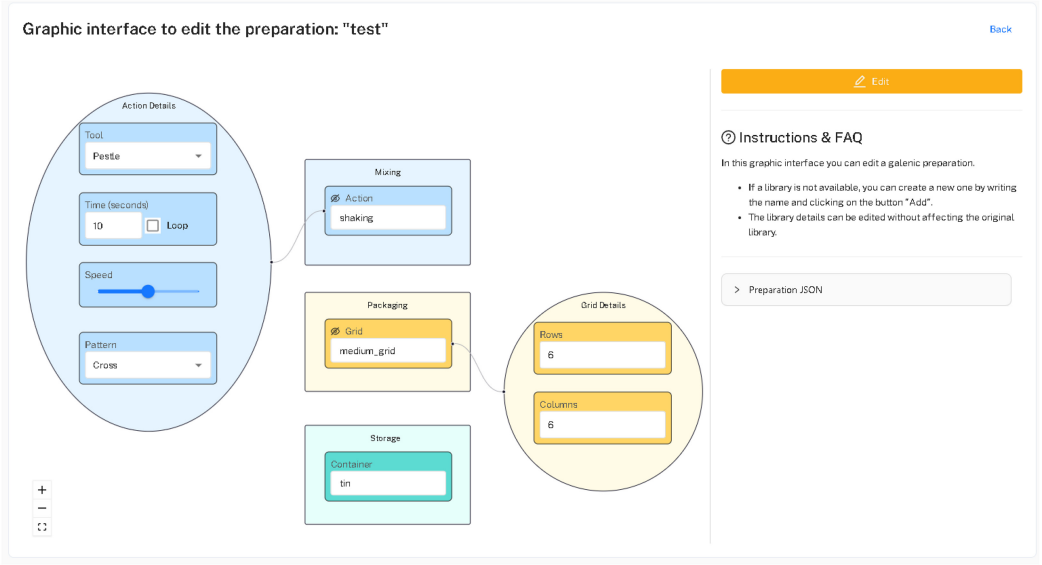

Fig. 8. The graphic interface.

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 when the EUD environment is released, thus, this aspect is out of the scope of the end users' activity.

5 User Study

An exploratory user study has been carried out to collect feedback from real users and assess the feasibility of the proposed robot programming approach for preparing galenic formulations.

5.1 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 taken by the researcher managing the user study have been subsequently examined by two researchers to identify relevant themes according to an inductive and latent thematic analysis [10]. Each theme led to identifying recommendations for system improvement;

```
{
  "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
  },
  "editMode": true
}
```

Listing 1. JSON structure of a defined task.

— the UEQ questionnaire [31] 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 six 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 by one researcher 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 (five females and four males) have been involved in the user study. All the participants work in the pharmaceutical field, with different backgrounds, specializations, and experiences. Table 1 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).

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 (Appendix A.1 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, the researcher managing the tests described the goal of the study, carried out a brief presentation of

Table 1. Participant Data

User	Age	Gender	Profession	Experience (years)	Galenic (1–5)	Computer (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

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, a researcher 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, the researcher asked 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 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). 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 presence in order to have a more comprehensive discussion and collection of feedback.

5.2 Results

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

5.2.1 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 (see Appendix A.2 for details).

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 [39], which historically impacted command-line interfaces and can now affect conversational interfaces as well [14]. This issue arose despite the existence of initial guidance provided in the chat upon opening

the page (see Figure 6). 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"* were manually added by the developer to the chat history and were 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 referred to this problem, another user suggested adding a section devoted to Frequently Asked Questions (FAQs).

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 undefined) 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 it was unclear. The difficulty in creating a connection between the application and the real world requires further investigations 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 results to be sufficiently robust; however, additional

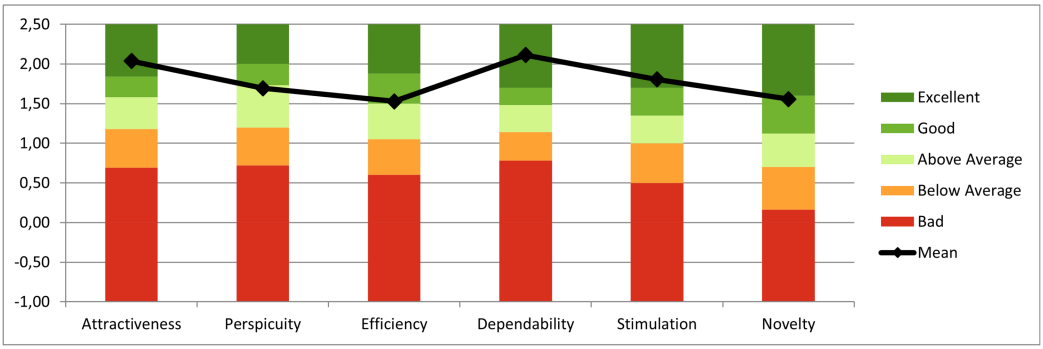

Fig. 9. UEQ results compared with the UEQ benchmark.

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 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 of LLMs. In our case, we may reduce 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, we will evaluate with further tests whether the suggestion to speed up the process could be suitable to all users; then, it is important to add system feedback to confirm the robot's data acquisition.

5.2.2 Findings from UEQ Questionnaire. Considering the UEQ benchmark ranges, which classify results as *Bad*, *Below average*, *Above average*, *Good*, and *Excellent* [45], the obtained results proved that interacting with our prototype application was very satisfying for the participants in the study (see Figure 9). 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).

5.2.3 Findings from the Final Interviews. Further themes emerged from the thematic analysis of the data gathered through the interviews. See Appendix B for interview questions and details about the thematic analysis.

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 be not 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"*.

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 preparations, 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.

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 preparations in the laboratory. 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 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.

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.

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 multi-purpose robot that could perform various operations, including diluting the SARS-CoV-2 vaccine and other generic preparations, such as the preparation of drips in 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.

6 Discussion

This section discusses the findings derived from the user study, as well as open issues that must be addressed in future work. Limitations of the work are finally highlighted.

6.1 Findings and Open Issues

The user study allowed us to demonstrate the validity of the proposed hybrid 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. As to learnability, upon analysis of the UEQ results, *Perspicuity* resulted in reaching a lower evaluation than the other UX aspects. Since the prototype presented in this paper is intended for professional use, it is reasonable to expect that something will not be immediately clear and that a training period will be necessary. Nonetheless, positive feedback indicates that there were no significant issues related to understanding and carrying out tasks after the initial use.

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

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* [38], 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 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 hybrid approach proposed here allows the user to discover and solve possible hallucinations using the graphic interface, thus always keeping control over the system output.

Prompt engineering is presently a topic of great interest, with several approaches being presented. For instance, in [54], 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 [43]. This solution enables requests to be distributed across multiple instances, rather than relying on a single one, in order 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 issue 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 that the system provides the user with an explanation of why it interpreted the prompt in a certain way, thus giving the user the opportunity to refine the prompt and enforce the desired interpretation. This aspect will be better investigated in future work.

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.

Further considerations can be made about the implementation of AI technologies with older users. Limited digital literacy and unfamiliarity with technology may hinder the 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.

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.

Finally, additional application scenarios in the medical and pharmacological domains emerged in our user study. It would be interesting, for example, to investigate how to support the steps foreseen in the production of other medicines, like cannabis-based products or antibiotics.

6.2 Limitations of the Work

The research presented in this paper 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 carried out so far.

First of all, as described in Section 4.2, 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. At present, the *gpt-4* 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 video-conference allowed collecting sufficient data to distill significant findings. Moreover, the average age of participants was relatively low, and this could have influenced the outcome; further studies will be performed in the future involving

participants of different ages. The number of participants was also rather limited. A study with more participants is needed to understand how the results might vary when considering a larger and more heterogeneous group of pharmacists.

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. We tried to mitigate this problem during the final interviews, where interesting feedback about the application of collaborative robots and the related programming environment was collected.

Finally, 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. Future work will consider the challenging issue of creating a robot program encompassing the whole human-robot interaction process needed for personalized medicine preparation.

7 Conclusion

Supporting improvement in the health domain is one of the main goals of responsible computing. This paper presented a case study about the use of collaborative robots in the preparation of galenic formulations to enhance efficiency, product quality, and work ergonomics. In this domain, end users are usually neither experts in robotics nor in computer programming. This calls for a responsible design approach to ensure the adequacy of the proposed system to the users' needs and characteristics and to guarantee human control of the overall process. To this purpose, a human-centered design methodology has been adopted for the development of PRAISE, an EUD environment that allows pharmacists to define robot tasks, helping them in the preparation of galenic formulations. The user interface of PRAISE exploits LLMs for the recognition of user intentions and the creation of tasks for collaborative robots, which can be checked and revised through a block-based graphic interface. The graphic interface can also be used to define new tasks from scratch, thus allowing users to choose the programming paradigm they prefer. This hybrid approach to robot programming has demonstrated significant versatility and efficacy; it was also very appreciated by the pharmacists who participated in the user study. Pharmacists were enthusiastic about having a tool that enhances their role by allowing them to focus on high-value tasks, leaving to the robot the execution of repetitive and time-consuming activities.

Declarations

Conflicts of interest/Competing interests

None of the authors have either any conflict of interest nor any competing interest.

Ethics/Privacy

Informed consent was collected from each participant included in the user study.

Availability of data and material

The anonymized data collected for this manuscript are available from the corresponding author upon request.

Acknowledgments

The authors wish to thank all the participants in the user study discussed in Section 5 for their availability and valuable collaboration. They also are indebted to the anonymous reviewers and associate editor for their insightful comments.

References

[1] Angus Addlesee, Neeraj Cherakara, Nivan Nelson, Daniel Hernández García, Nancie Gunson, Weronika Sieińska, Marta Romeo, Christian Dondrup, and Oliver Lemon. 2024. A multi-party conversational social robot using LLMs. In *Companion of the 2024 ACM/IEEE International Conference on Human-Robot Interaction*. 1273–1275.

[2] Gopika Ajaykumar, Maureen Steele, and Chien-Ming Huang. 2021. A survey on end-user robot programming. 54, 8, Article 164 (Oct.2021), 36 pages. DOI: https://doi.org/10.1145/3466819

[3] Sonya Alexandrova, Zachary Tatlock, and Maya Cakmak. 2015. RoboFlow: A flow-based visual programming language for mobile manipulation tasks. In *2015 IEEE International Conference on Robotics and Automation (ICRA'15)*. 5537–5544. DOI: https://doi.org/10.1109/ICRA.2015.7139973

[4] Brenna D. Argall, Sonia Chernova, Manuela Veloso, and Brett Browning. 2009. A survey of robot learning from demonstration. *Robot. Auton. Syst.* 57, 5 (May2009), 469–483. DOI: https://doi.org/10.1016/j.robot.2008.10.024

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

[6] R. G. Beri, D. Wolton, and C. H. Coulon. 2019. Opportunities for modern robotics in biologics manufacturing. *Bioprocess. Int.* 17, 4 (2019).

[7] Sara Beschi, Daniela Fogli, and Fabio Tampalini. 2019. 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*. Springer, 51–66.

[8] Giorgio Bimbatti, Daniela Fogli, and Luigi Gargioni. 2023. Can ChatGPT support end-user development of robot programs? In *CEUR Workshop Proceedings*, Vol. 3408. https://ceur-ws.org/Vol-3408/short-s2-03.pdf

[9] Guilherme Deola Borges, Diego Luiz de Mattos, André Cardoso, Hatice Gonçalves, Ana Pombeiro, Ana Colim, Paula Carneiro, and Pedro M. Arezes. 2022. Simulating human-robot collaboration for improving ergonomics and productivity in an assembly workstation: A case study. *Occupational and Environmental Safety and Health III* (2022), 369–377.

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

[11] Nina Buchina, Sherin Kamel, and Emilia Barakova. 2016. 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'16)*. 185–191. DOI: https://doi.org/10.1109/ROMAN.2016.7745109

[12] Tommaso Calò and Luigi De Russis. 2023. Leveraging large language models for end-user website generation. In *End-User Development*, Lucio Davide Spano, Albrecht Schmidt, Carmen Santoro, and Simone Stumpf (Eds.). Springer Nature Switzerland, Cham, 52–61.

[13] Xianghua Chu, Heidi Fleischer, Thomas Roddelkopf, Norbert Stoll, Michael Klos, and Kerstin Thurow. 2015. A LC-MS integration approach in life science automation: Hardware integration and software integration. In *2015 IEEE International Conference on Automation Science and Engineering (CASE'15)*. 979–984. DOI: https://doi.org/10.1109/CoASE.2015.7294226

[14] Eric Corbett and Astrid Weber. 2016. 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*. 72–82.

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

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

[17] Eurostat. 2020. Ageing Europe - Statistics on Population Developments. https://ec.europa.eu/eurostat/statistics-explained/index.php?title=Ageing_Europe_-_statistics_on_population_developments

[18] Heidi Fleischer, Daniel Baumann, Xianghua Chu, Thomas Roddelkopf, Michael Klos, and Kerstin Thurow. 2018. Integration of electronic pipettes into a dual-arm robotic system for automated analytical measurement processes behaviors. In *2018 IEEE 14th International Conference on Automation Science and Engineering (CASE'18)*. 22–27. DOI: https://doi.org/10.1109/COASE.2018.8560377

[19] Daniela Fogli, Luigi Gargioni, Giovanni Guida, and Fabio Tampalini. 2022. A hybrid approach to user-oriented programming of collaborative robots. *Robotics and Computer-Integrated Manufacturing* 73 (2022), 102234.

[20] Luigi Gargioni and Daniela Fogli. 2024. Integrating ChatGPT with blockly for end-user development of robot tasks. In *Companion of the 2024 ACM/IEEE International Conference on Human-Robot Interaction*. 478–482.

[21] Luigi Gargioni and Daniela Fogli. 2024. Integrating ChatGPT with Blockly for end-user development of robot tasks. In *Companion of the 2024 ACM/IEEE International Conference on Human-Robot Interaction*. 478–482.

[22] T. H. Geersing, E. J. F. Franssen, F. Pilesi, and M. Crul. 2019. Microbiological performance of a robotic system for aseptic compounding of cytostatic drugs. *European Journal of Pharmaceutical Sciences* 130 (2019), 181–185.

[23] Google. [n. d.]. Google Bard. *https://bard.google.com/chat.* Accessed: September 25th, 2023.
[24] Forrest Huang, Gang Li, Xin Zhou, John F. Canny, and Yang Li. 2021. Creating user interface mock-ups from high-level text descriptions with deep-learning models. arxiv:2110.07775 [cs.HC]
[25] Justin Huang and Maya Cakmak. 2017. 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* (Vienna, Austria) (HRI '17). Association for Computing Machinery, New York, NY, USA, 453–462. DOI: https://doi.org/10.1145/2909824.3020215
[26] Yuna Hwang, Arissa Sato, J., Pragathi Praveena, and Bilge Mutlu. 2024. Understanding generative AI in robot logic parametrization. In *Companion of the 2024 ACM/IEEE International Conference on Human-Robot Interaction.*
[27] Business Research Insights. 2022. 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. https://www.businessresearchinsights.com/market-reports/pharma-market-102426
[28] Ellen Jiang, Edwin Toh, Alejandra Molina, Aaron Donsbach, Carrie J. Cai, and Michael Terry. 2021. GenLine and GenForm: Two tools for interacting with generative language models in a code editor. In *Adjunct Proceedings of the 34th Annual ACM Symposium on User Interface Software and Technology* (Virtual Event, USA) (UIST '21 Adjunct). Association for Computing Machinery, New York, NY, USA, 145–147. DOI: https://doi.org/10.1145/3474349.3480209
[29] Ulas Berk Karli, Juo-Tung Chen, Victor Nikhil Antony, and Chien-Ming Huang. 2024. Alchemist: LLM-aided end-user development of robot applications. In *Proceedings of the 2024 ACM/IEEE International Conference on Human-Robot Interaction.* 361–370.
[30] Callie Y. Kim, Christine P. Lee, and Bilge Mutlu. 2024. Understanding large-language model (LLM)-powered human-robot interaction. *arXiv preprint arXiv:2401.03217* (2024).
[31] Bettina Laugwitz, Theo Held, and Martin Schrepp. 2008. 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.* Springer, 63–76.
[32] Henry Lieberman, Fabio Paternò, and Volker Wulf. 2006. *End User Development (Human-Computer Interaction Series).* Springer-Verlag, Berlin.
[33] 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. 2023. Benchmarking large language models' performances for myopia care: A comparative analysis of ChatGPT-3.5, ChatGPT-4.0, and Google Bard. *EBioMedicine* 95 (2023).
[34] Yueyue Liu, Zhijun Li, Huaping Liu, and Zhen Kan. 2020. Skill transfer learning for autonomous robots and human–robot cooperation: A survey. *Robotics and Autonomous Systems* 128 (2020), 103515. DOI: https://doi.org/10.1016/j.robot.2020.103515
[35] Martin Maguire. 2001. Methods to support human-centred design. *International Journal of Human-Computer Studies* 55, 4 (2001), 587–634. DOI: https://doi.org/10.1006/ijhc.2001.0503
[36] Robins Mathew, Robert McGee, Kevin Roche, Shada Warreth, and Nikolaos Papakostas. 2022. Introducing mobile collaborative robots into bioprocessing environments: Personalised drug manufacturing and environmental monitoring. *Applied Sciences* 12, 21 (2022), 10895.
[37] Meta. [n. d.]. Meta Llama 2. *https://ai.meta.com/llama/.* Accessed: September 25th, 2023.
[38] Jakob Nielsen. 1994. Enhancing the explanatory power of usability heuristics. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Boston, MA, USA) (CHI '94). Association for Computing Machinery, New York, NY, USA, 152–158. DOI: https://doi.org/10.1145/191666.191729
[39] Donald A. Norman. 1988. *The Psychology of Everyday Things.*Basic books.
[40] OpenAI. [n. d.]. OpenAI ChatGPT API Documentation. *https://platform.openai.com/docs/guides/gpt.* Accessed: September 25th, 2023.
[41] Fabio Paternò and Volker Wulf (Eds.). 2017. *New Perspectives in End-User Development.* Springer, Cham. DOI: https://doi.org/10.1007/978-3-319-60291-2
[42] Chris Paxton, Andrew Hundt, Felix Jonathan, Kelleher Guerin, and Gregory D. Hager. 2017. CoSTAR: Instructing collaborative robots with behavior trees and vision. In *2017 IEEE International Conference on Robotics and Automation (ICRA'17)* (Singapore, Singapore). IEEE Press, 564–571. DOI: https://doi.org/10.1109/ICRA.2017.7989070
[43] David Porfirio, Mark Roberts, and Laura M. Hiatt. 2024. Goal-oriented end-user programming of robots. (2024).
[44] Andrew Schoen and Bilge Mutlu. 2024. OpenVP: A customizable visual programming environment for robotics applications. In *Proceedings of the 2024 ACM/IEEE International Conference on Human-Robot Interaction.* 944–948.
[45] Martin Schrepp, Andreas Hinderks, and Jörg Thomaschewski. 2017. Construction of a benchmark for the user experience questionnaire (UEQ). *International Journal of Interactive Multimedia and Artificial Intelligence* 4 (062017), 40–44. DOI: https://doi.org/10.9781/ijimai.2017.445

[46] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288* (2023).
[47] Marta Uriel, Diego Marro, and Carlota Gómez Rincón. 2023. An adequate pharmaceutical quality system for personalized preparation. *Pharmaceutics* 15, 3 (2023). DOI: https://doi.org/10.3390/pharmaceutics15030800
[48] Sai Vemprala, Rogerio Bonatti, Arthur Bucker, and Ashish Kapoor. 2023. *ChatGPT for Robotics: Design Principles and Model Abilities.* Technical Report. Microsoft. https://www.microsoft.com/en-us/research/uploads/prod/ 2023/02/ ChatGPT_Robotics.pdf
[49] Valeria Villani, Fabio Pini, Francesco Leali, and Cristian Secchi. 2018. Survey on human–robot collaboration in industrial settings: Safety, intuitive interfaces and applications. *Mechatronics* 55 (2018), 248–266.
[50] David Weintrop, Afsoon Afzal, Jean Salac, Patrick Francis, Boyang Li, David C. Shepherd, and Diana Franklin. 2018. 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* (Montreal QC, Canada) (CHI '18). Association for Computing Machinery, New York, NY, USA, 1–12. DOI: https://doi.org/10.1145/3173574.3173940
[51] Jules White, Quchen Fu, Sam Hays, Michael Sandborn, Carlos Olea, Henry Gilbert, Ashraf Elnashar, Jesse Spencer-Smith, and Douglas C. Schmidt. 2023. A prompt pattern catalog to enhance prompt engineering with ChatGPT. *arXiv preprint arXiv:2302.11382* (2023).
[52] World Health Organization. 2023. Health Products Policy and Standards. https://www.who.int/teams/health-product-and-policy-standards/standards-and-specifications/norms-and-standards- for-pharmaceuticals/guidelines
[53] Tongshuang Wu, Michael Terry, and Carrie Jun Cai. 2022. 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*. 1–22.
[54] J. D. Zamfirescu-Pereira, Richmond Y. Wong, Bjoern Hartmann, and Qian Yang. 2023. 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*. 1–21.
[55] Zuyuan Zhu and Huosheng Hu. 2018. Robot learning from demonstration in robotic assembly: A survey. *Robotics* 7, 2 (April2018), 17–17.
[56] Ádám Wolf, David Wolton, Josef Trapl, Julien Janda, Stefan Romeder-Finger, Thomas Gatternig, Jean-Baptiste Farcet, Péter Galambos, and Károly Széll. 2022. Towards robotic laboratory automation Plug & Play: The "LAPP" framework. *SLAS Technology* 27, 1 (2022), 18–25. DOI: https://doi.org/10.1016/j.slast.2021.11.003

Appendices

A Observation and Thinking Aloud in the User Study

A.1 Tasks Executed

The list of tasks performed during the user study is the following:

(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.

A.2 Thematic Analysis

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 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

B Final Semi-structured Interviews

B.1 Questions

(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 to other medical and/or pharmaceutical problems as well?
(6) Do you have any other comments regarding application, approach, technology, or anything else?

B.2 Thematic Analysis

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 to repetitive tasks
Graphic vs. chat-based interface	• User preference for graphic interface • Mimicking work practice • Efficiency of 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

Received 8 May 2024; revised 19 November 2024; accepted 8 January 2025