Read, Watch, or Chat: Investigating Learning Barriers and learner-content Interaction in End-user Robot Programming Tasks

Authors: Felipe Fronchetti, Kostadin Damevski, Logan Schorr

This document represents a draft of our research paper to be submitted to the Transactions on Human-Robot Interaction Journal. The information presented in this document is confidential and should not be shared with the public audience at the moment.

A. About the study

Robot programming can be a challenging task for individuals without prior experience in robotics. With the advancements of end-user-friendly technologies in the robot industry, the gap between end-users and robots is coming to an end, increasing the popularity of such machines to a broad audience. Even though advancements in end-user robot programming have been made throughout the years, little is known about what type of information end-users seek when programming a robot. While technical manuals seem like a suitable source of information for engineers, it is not clear if the same holds for end-users. It is also unclear in the literature what kind of learning barriers end-users face while programming a robot for the first time.

In this study, we aim to investigate these two topics: First, we want to investigate potential learning barriers end-users face when programming a robot for the first time. And second, how end-users consume different types of learning resources (e.g., videos, manuals) to overcome their learning barriers.

B. About the experiment

To answer such questions, we propose an experiment involving a robot programming task to be accomplished by undergraduate students from a Computer Science department in the United States. In this experiment, each student will be invited to an individual session to try the robot programming task using a block-based language designed for end-users without experience experience in robotics. This language will be installed on the controller of a collaborative robot that participants will use to complete the task (See Figure 1). In the experimental task, participants will be focused only on the programming environment, and all the necessary resources (e.g., robot positions, and robot preliminary configuration) will be provided in advance by the authors.

The block-based language of choice will be the Wizard Programming Tool developed by ABB, an end-user-friendly language that allows users to "*teach robots within 10 minutes with no prior experience necessary*". Using this language, users will have to implement a pick-and-place solution to sort canned goods among two dispensers placed on the table where the robot is installed (See Figure 1). In the initial state of the task, two types of canned goods (e.g., carrots and green beans) will be randomly placed on one of the dispensers. Participants will have to implement an interface using the block-based language that allows users to either move a can

to the other dispenser or move the current can to the last position of the current dispenser (i.e., place the can on the top part of the dispenser, so it can roll back to the last position). To accomplish these actions, participants will not only use robot movement and gripper instructions to move the cans around the table, but will also use other pre-defined buttons from the Wizard Programming Tool that allow developers to easily build a button-based interface to define the next actions of the robot (e.g., move the current can to the other dispenser, or put it on top of the current one). This task mimics the idea of sorting algorithms from traditional computer science exercises (e.g., sorting queues, lists, and stacks), and should explore most of the features available in the block-based environment made by ABB, including but not limited to conditionals, loops, variables and more.

To start the experiment, a quick introduction to the robot workspace will be given by a proctor to each participant. The robot, the programming language, the teaching pendant, and the task will be introduced to each student in a quick presentation. We will also record training videos that will be presented to each participant before the beginning of the experiment. Participants will have about one hour and twenty minutes to solve the task using the block-based language. The proctors will only assist participants throughout the experiment if the robot stops by a mistake in the programmed solution, but no other information will be given by them to the participants.

Figure 1. The experiment workspace

As our goal with this experiment is to better understand what kinds of questions end-users face when programming a robot for the first time (and the learning materials they like the most when facing a question), next to the robot, participants will also have a computer available, which we will call as the “help desk” in this experiment. At the help desk, a computer program simulating a support system will be available for use by participants facing difficulties in accomplishing the

task. On the first page of this program, participants will be able to request assistance by defining what type of robot programming difficulty they are facing (See Figure 2).

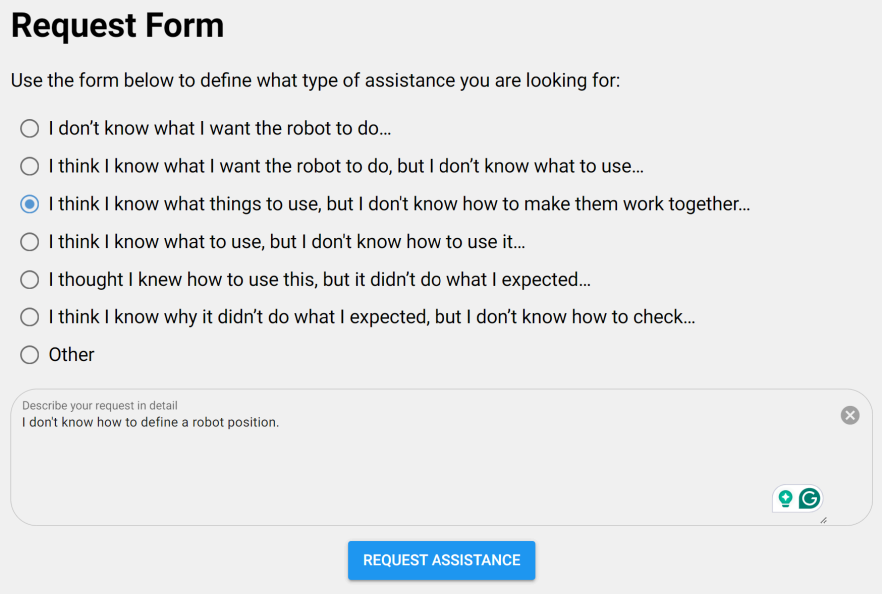

Figure 2. The form used by participants to request assistance.

Once assistance is requested from a participant, a new window will open with three sections of information: read, watch, and chat. In the read section, text-based materials and a wiki system made by the robot manufacturer will be available for use. In the watch section, video tutorials recorded by the robot manufacturer will also be available. Finally, in the chat section, participants will be able to get assistance from a chatbot similar to ChatGPT, trained with the text and video resources available in the other two sections.

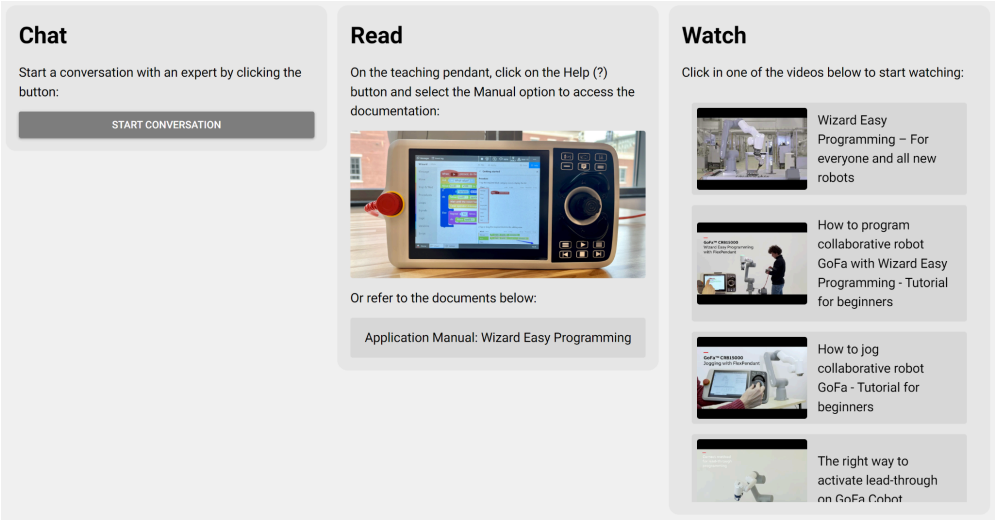

Figure 3. The three sections of information: read, watch, and chat.

Throughout the process, we expect to record every user-content interaction within our help desk, from what learning resources participants are using to how much time they spend on each section. Once a participant gets the answer they are looking for in their question, they will be instructed to close the request using a close button available on the interface. Once a request is closed, the participant will be redirected to a quick questionnaire requesting feedback about their learning experience with the learning materials provided, followed by another redirection to the first page where they can open new requests for the following difficulties.

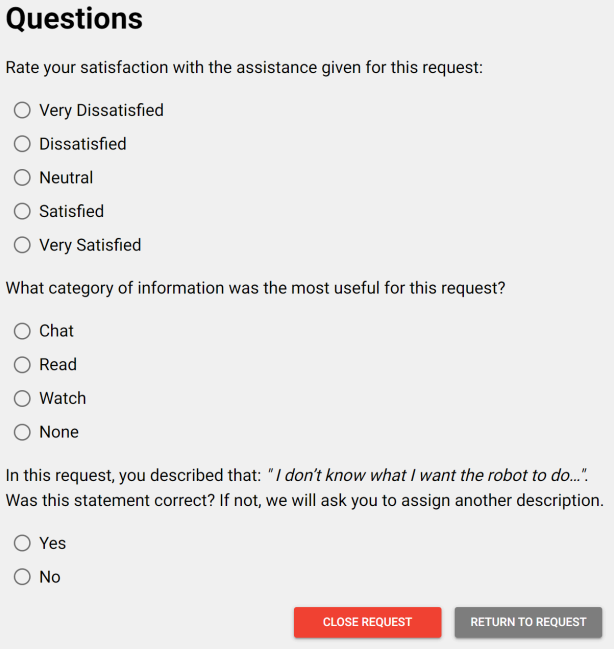

Figure 4. The questionnaire presented to participants when they close a request.

Using this system, we expect to better understand what kinds of difficulties end-users face while programming the robot, and what kind of materials they use the most. After the experiment, we will also run a post-experiment questionnaire with each participant to get their overall opinion about the learning materials provided, the block-based programming language, and the experiment itself.

C. About the learning resources

In this experiment, we want to simulate a real use-case scenario of an end-user buying a collaborative robot to apply robotics to their professional domain. We want to use this opportunity to also investigate the status quo of cobots sold by robot manufacturers (i.e., do robot manufacturers provide the necessary resources end-users need?). Based on this hypothesis, we plan to offer in our help desk only the learning materials provided by our manufacturer of choice (Asea Brown Boveri, also known as ABB) on their webpage. In the following subsections, we list the materials we will be using in this experiment. These materials were extracted from the official page of the robot we are using (the ABB CRB 15000), which are also the ones recommended for use by the manufacturer.

C.1 Text-based materials

- ABB Wizard (Block-based) Programming Tool - Application Manual (PDF, 32 pages)

https://library.e.abb.com/public/d7ee7462d4fd45088bc4d2ee8471ab94/3HAC073766%20AM%20Wizard-en.pdf/

(The manufacturer also provides the manual above through a help button in the block-based environment, and we will instruct participants about its existence)

C.2 Video-based materials

- Wizard Easy Programming – For everyone and all new robots
  https://www.youtube.com/watch?v=Kmv5jUI3WF0

- How to program collaborative robot GoFa with Wizard Easy Programming - Tutorial for beginners
  https://www.youtube.com/watch?v=zPnEOQX4jUA

- Step-by-step guide on pick and place application with Wizard Easy Programming tool
  https://www.youtube.com/watch?v=eUgqXsWMmwI

- Using Wizard to create a PCB assembly application in minutes
  https://www.youtube.com/watch?v=nj0X8fLj1SE

- How to use “Move” blocks in Wizard with your virtual robot
  https://www.youtube.com/watch?v=zizcQSnPyyE

C.3 Chat

The chatbot will be based on a ChatGPT language model. We will instruct the model to act as an assistant in a robot programming task. We will supply the ChatGPT model with the necessary information regarding the robot specification and the block-based language. The chatbot will not contain any sort of advantage over the other learning materials, such as detailed information from the experiment or potential solutions for the task given to participants.

E. About the robot

In this study, we will be using an ABB CRB 15000 (also known as ABB GoFa), a one-armed collaborative robot made by Asea Brown Boveri (ABB). The gripper installed on this robot will be an RG6 finger gripper from OnRobot, which participants will use to pick and place the cans in the workspace. To interact with the robot, participants will use the teaching pendant provided by the robot manufacturer. To program robot instructions, participants will have available the ABB Wizard Easy Programming Tool (Version 1.5.2) installed on the teaching pendant, the latest version of the block-based programming language made by ABB for collaborative robots. The

teaching pendant will be connected to the controller of GoFa, an OmniCore C30 robot controller running RobotWare 7.12.0, the latest operating system made available by ABB in 2024.

In our experiment, we will provide all the preliminary settings necessary for users to accomplish the task using the block-based language. This includes the robot positions necessary to pick and place the cans in the dispensers. Participants will not have to interact with the robot directly, only to control the program execution from the block-based environment. We decided to provide the positions in advance as our goal in this experiment is to evaluate potential difficulties in robot programming, and not with robot positioning (already proven to be troublesome for end-users in previous studies). By limiting the participants to interact only with the block-based environment, we also reduced the time necessary to accomplish the task, limiting the dropout rates of participation in our experiment.