TU WIEN Informatics

Crowdsourcing- basierte Evaluierung von webbasierten Roboter-Programmierumgebungen am Beispiel von 'Assembly'

DIPLOMARBEIT

zur Erlangung des akademischen Grades

Diplom-Ingenieurin

im Rahmen des Studiums

Wirtschaftsinformatik

eingereicht von

Daria Piacun, Bacc.Oec, BSc
Matrikelnummer 01402189

an der Fakultät für Informatik

der Technischen Universität Wien

Betreuung: Univ.-Prof. Dr.-Ing. Dipl.-Ing. Sebastian Schlund
Mitwirkung: Dr.-Ing., M.Sc., M.A. Tudor B. Ionescu

Wien, 16. August 2021

Daria Piacun | Sebastian Schlund

Crowdsourced Evaluation of Web-Based Robot Programming Environments by the Example of ’Assembly’

DIPLOMA THESIS

submitted in partial fulfillment of the requirements for the degree of

Diplom-Ingenieurin

in

Business Informatics

by

Daria Piacun, Bacc.Oec, BSc
Registration Number 01402189

to the Faculty of Informatics

at the TU Wien

Advisor: Univ.-Prof. Dr.-Ing. Dipl.-Ing. Sebastian Schlund
Assistance: Dr.-Ing., M.Sc., M.A. Tudor B. Ionescu

Vienna, 16th August, 2021

Daria Piacun Sebastian Schlund

Erklärung zur Verfassung der Arbeit

Daria Piacun, Bacc.Oec, BSc

Hiermit erkläre ich, dass ich diese Arbeit selbständig verfasst habe, dass ich die verwendeten Quellen und Hilfsmittel vollständig angegeben habe und dass ich die Stellen der Arbeit – einschließlich Tabellen, Karten und Abbildungen –, die anderen Werken oder dem Internet im Wortlaut oder dem Sinn nach entnommen sind, auf jeden Fall unter Angabe der Quelle als Entlehnung kenntlich gemacht habe.

Wien, 16. August 2021

Daria Piacun

Acknowledgements

This thesis took a little more than a year to finish, due to the world and life circumstances. I have been writing it in three countries, starting slowly in Spain during the hard Covid lockdown, continuing barely in my home country Croatia during the final part of my sabbatical year, and somewhat in Austria while working full time. The most part has been written upon my return to Spain, in one specific coffee shop in Barcelona by the sea, where I drank endless coffees, was treated by the waiters as if I was in my own living room and got constant motivation and inspiration. Thank you Farggi crew!
This has definitely not been the work of one individual, which is why I want to express my gratitude to some of them here.
Sincere thanks to my mentor Sebastian Schlund, for his expert and essential feedback and counsel which helped to bring my work to a higher level.
Next, this thesis would have been impossible without the aid and support of Tudor B. Ionescu, who tirelessly and very patiently gave me invaluable help throughout the entire last year. You kept directing me back to the right research path and provided me with the tools that I needed to successfully complete my thesis. Additionally, thank you for all the opportunities and ideas you gave me to advance my research, resulting in my very first scientific research paper.
To my flatmates, who had a lot of understanding about what I was going through and supported and motivated me greatly. To my friends, for canceling some trips and appointments for having a bad consciousness about not writing the thesis (and occasionally ending up not writing it that day anyway). You guys are incredible to that extent, that apart from motivating me greatly, you helped with the translations- Danke Lisa, and even handed in the hard copy of my thesis because I was not in the town on the deadline- Mirza, this one goes to you, hvala!
Finally a big thank you to my family, for supporting my decision to move to Vienna from day one and believing in me even when I didn't.
Now it is time for the next step.

Kurzfassung

Eine neue Welle von kollaborativen Robotern, die für die Zusammenarbeit mit Menschen konzipiert sind, bringt die Automatisierung, die in der Vergangenheit in großindustriellen Umgebungen zu beobachten war, in neue, vielfältige Kontexte[95]. Die Forschung hat jedoch gezeigt, dass sich die Innovationslücke zwischen Roboterhardware und -software nicht zu schließen scheint - die Hardwareentwicklung hat ein extrem hohes Niveau erreicht. Im Gegensatz dazu blieb die Software auf dem Niveau der Programmiersprachen aus den frühen 1990er Jahren, die fast nur unter Ingenieuren bekannt sind, recht rudimentär. Infolgedessen erfahren die neuesten Robotertechnologien keine hohe Akzeptanz oder gar Anerkennung durch den größten Teil des Publikums ohne Ingenieursprofil.
Das Hauptziel dieser Arbeit ist es, die Bewertung der Benutzerfreundlichkeit von Roboterprogrammierumgebungen zu verbessern, indem eine Methodik entwickelt wird, mit der die Erstellung und Durchführung von Crowdsourced-Evaluierungen schnell und mit möglichst geringen Kosten durchgeführt werden kann, was zu einer höheren Akzeptanz der Benutzergruppen insgesamt führt. Mit unserem Ansatz evaluieren wir dann *Assembly*[63], eine webbasierte Roboterprogrammierumgebung, die an der TU Wien als Teil eines öffentlich geförderten Forschungsprojekts namens Cobot Meets Makerspace (COMEMAK)[8] entwickelt wurde, zu dem auch diese Arbeit gehört.

Abstract

A new wave of collaborative robots designed to work alongside humans is bringing the automation historically seen in large-scale industrial settings in new, diverse contexts[95]. However, the research has shown that the innovation gap between robot hardware and software doesn't seem to close- the hardware development reached extremely high levels. In contrast, the software side remained quite rudimentary at the level of programming languages from the early 1990s, known almost exclusively among engineers. Consequently, the newest robotic technologies don't experience high acceptance levels or even recognition from the biggest part of the audience without engineering profiles.
The main goal of this thesis is to improve robot programming environments usability evaluation by developing a methodology such that creating and executing crowdsourced evaluation can be done quickly and with the lowest cost possible, resulting in higher user acceptance overall user groups. With our approach, we then evaluate *Assembly*[63], a web-based robot programming environment developed at TU Wien as a part of a publicly-funded research project called Cobot Meets Makerspace (COMEMAK)[8].

Contents

CHAPTER 1

Introduction

1.1 General introduction to the topic

Currently, we find ourselves in the middle of the fourth industrial revolution, which is efficiently working its way to digitize in the first place, customize in most cases and automatize business processes wherever possible. As recognized by [67], one of the most important aspects of industry 4.0 is the assortment of digitalization strategies in CPPS. Terms such as smart, intelligent or digital factory[16] have been coined to describe an inter-connected and highly flexible manufacturing system utilizing a permanent data stream from connected operations and production systems in order to acquire information and learn to adapt to new demands[86].

In no place is the change due to the fourth industrial revolution bound to be felt more than in the working environment, specifically in the production facilities. Equivalent to the past industrial revolutions, this one is pledged to significantly influence individuals' lives as AI and expanded automation see numerous sorts of employments drastically change or even vanish. Simultaneously, altogether several new kinds of employments are emerging. What is more, this shift should take place a lot earlier in the process, specifically already during the education, adapting both engineering and regular courses to the ongoing revolution[82]. The authors in [81] even define the term Education 4.0 and discuss in their paper the need for transferring the skills such as hands-on experience, robot programming, the deployment, and integration of competencies and experiences from the digital native generations to those less used to these rather unconventional modes of learning or learning in general.

The fourth industrial revolution depicts the softening of limits between the physical, digital, and natural universes. It is a combination of advances in AI, IOT, robotics, 3D printing, genetic engineering, quantum computing, and many other technologies. It's the aggregate power behind numerous items and administrations that are quick in getting imperative to present modern-day life[29]. Finally, one of the inevitable paradigms of

Industry 4.0 implies close human-robot cooperation, which makes extensive research necessary, not only restricted to robotics but also in user-centered effortless programming as a strong foundation for smooth HRI[62].
In Industry 4.0, robots and humans manifest five different combinations of mutual work[41]:

1. they work individually and independently, spatially separated from each other, each working on their task, unaware of the existence of the other
2. they coexist in separated workspace and tasks, but aware of each other
3. they work synchronously in a shared workspace and possibly, but not necessarily, on the same task, while having just one active interaction partner at a time
4. they cooperate, which indicates simultaneous work of the two but on a different task at a time
5. collaborate on interlinking tasks, which demands using smart human-machine interfaces

This thesis will emphasize the last and closest form of collaboration between human and machine workers.
For example, the field of industrial robotics proliferated in the past decade, with the companies looking to embrace this technology in diversified areas as it brings high ROI in no time, increases productivity and velocity, and reduces the risk of errors caused mainly by monotony and repetitiveness of tasks. Robots, in contrast to human workers, would mean a single more considerable initial investment in the hardware, which can later be employed 24 hours/day, keep the entire machinery operating even without the physical presence of human workers, master repetitive tasks quickly, and even specialize in customize-able ones.
Speaking of customization, the ever rising demand for lot size 1 products is challenging this approach mostly from the software point of view, task changes within one industrial robots timeline being seldom[71]. Complexity and high costs of programming and re-programming the robot are one of the main reasons for low flexibility in these changes[70]. This emphasizes the relevance of creating a simple yet powerful and universally applicable robotic programming environment.
Obviously, the HRI is becoming a more common occurrence and therefore an essential field of research, addressing, above all secure, but also seamless and intuitive, user-friendly programming and application interface interaction.
This leads us to the last concept relevant for this thesis, which is called Web 4.0, and as defined by [96] can be seen as the subset of IOT called web of things. Other authors [74] set the foundation of Web 4.0 by defining it's three main pillars:

1. Natural language understanding technique

2. New ways of communication between human-to-machine, machine-to-machine and machine-to-human

3. New interface models
   A nice UI overview was given by [65], where the early interface was called Character user interface since the only interaction element were characters. Next generation interfaces got classified as Graphical user interface, since they included elements such as icons and menus. Nowadays it became common to call the UIs With the rapid development of IT technology, interface was developed to be Natural user interface since it allows for HRI using natural human activities such as voice, motion, gesture, and biological signal recognition.

1.2 Problem definition

What we witnessed during the recent fourth industrial revolution is that while the design and development of the robotic hardware resulted in potent and flexible machinery with an ever-higher set of capabilities, relatively insufficient attention has been put on the accompanying software components[95], such as a visual representation of programming and interaction tools. The research[43] has shown that the most recognized industrial robots can almost exclusively only be introduced and controlled via proprietary, text-based, and predominantly low-level programming languages, derived from Pascal and BASIC and created back in the early 1990s. These programming languages have historically been designed by engineers for fellow engineer colleagues, which inevitably implies that writing or even reading these programs requires corresponding schooling or years of education and training. This furthermore means that only a minority of small and medium-sized enterprises will be able to benefit from robotic automation since they cannot afford that highly educated workers[78]. Finally, as programming becomes ever-present and imperative even for employees with low or no technical background[95], developing simple and user-friendly robot programming interfaces as a way to ease their way into the process has become an essential branch of research. There are already many robot programming environments out there, but none have yet caught on as the indispensable standard.
Furthermore, widely acknowledged and validated methods of objective assessments of UI are both resource and time consuming[25]. As a response to these issues, this thesis offers a cost-effective and reasonably fast solution to the problem in the form of a complete methodological framework for crowdsourced evaluation of web-based robotic programming environments. What is more, the thesis will at the same time prove the effectiveness of the methodology on the example of assessment of *Assembly* robot programming environment over MTURK platform.

1.2.1 Project Cobot Meets Makerspace (COMEMAK)[8]

In this section COMEMAK, the project funded by Austrian Research Promotion Agency FFG is presented. The project's vision is to increase awareness of collaborative robotics technology by simplifying availability and access to public workshops commonly known

as makerspaces[8]. Makerspaces are basically controlled environments with robotic equipment which permits access to ordinary people with little or even without any necessary educational background or training, with the final goal of creating the sociotechnical basis for the seamless introduction of the idea and the concept of COBOTs.
Numerous robotic variations such as industrial, medical, or space are currently employed to aid humans with certain tasks that might be considered demanding for them[50]. The term COBOT already exists since 1999, when it was coined by [80] and defined as a combination of the words "collaborative" and "robotics", describing a robot intended to physically interact with a human worker in order to assist her with a task on hand. In order to ease the HRI, collaborative robotic systems are being popularized under the name COBOT systems. Their correct definition is forming an essential part of COMEMAK project.
There are certain challenges that this project is facing, the most important ones being following[8]:

1. Creating uncomplicated and clean visual programming environment with the aim of detaching complex back end from necessarily simplified front end for the common end-user(drawing on examples such as Blockly[7] and Franka Emika[15]).

2. Investigate, create and document use cases and assembly assistance tools for human-robot collaboration in single-user assembly, where no prior programming nor robotic knowledge is required.

3. Exploring, recording and evaluating new teaching and learning methods, relevant assembly sub-processes as well as new human-robot collaboration forms.

4. Safety concerns during HMI.

This thesis will mainly focus on the second challenge. The findings will then contribute to the first challenge, bearing in mind that *Assembly* is still in its alpha version.

1.2.2 ***Assembly***[63]

Assembly is an easy to use, open source, web- based (Google Chrome optimized), block-based robotic programming environment which is being developed as the answer to the first above challenge. It is being developed as a part of the COMEMAK project.
Assembly UI programming concepts rely of five main building blocks:

1. Actors (Blocks)
   Assembly is developed as a hybrid between text and block based actors. As visible from the following figure 1.1, most of the basic programming concepts already exist: The first row consists of the actors who are meant for precise movement of the entire robot (move to), adjusting the robot's gripper position (dp, dr), opening and closing the gripper, and defining a dynamic action.

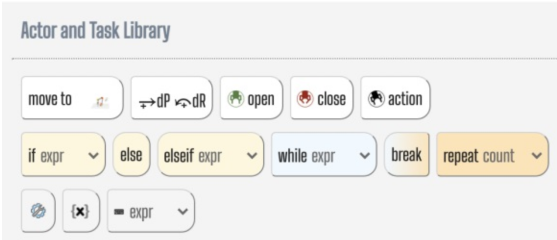


Figure 1.1: *Assembly* actor and task library

The second row consists of basic conditional statements such as if, else, elseif, and also loop enabling ones such as while and repeat. The final row is meant for explicit parameter, variable and condition setting.
Assembly's core idea is to simplify block creation, use and re-use by standardizing them.
Additional help is offered to the end-user via "help option". It can be easily accessed merely by clicking on each actor and looks like this:

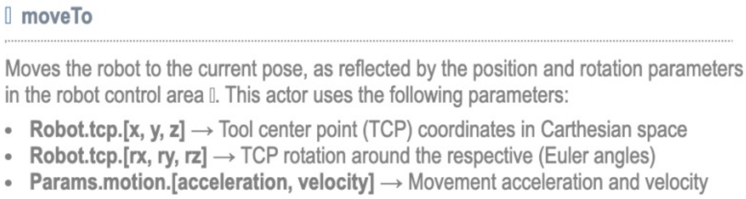


Figure 1.2: *Assembly* help option

2. Blackboard
   The blackboard is a behavioral design pattern used in computer programming, that is used to prioritize the actors of different systems that are bound to work either in parallel or sequentially in certain program[6]. In *Assembly*, this concept is used to ease the creation and management of the actors.
   As visible from the above figure 1.3, the *Assembly* blackboard uses the following

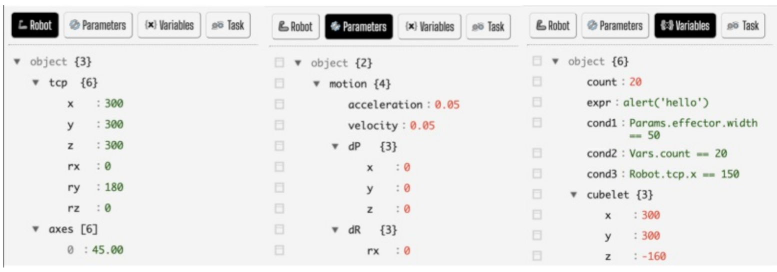


Figure 1.3: *Assembly* blackboard

three objects:

a) "Robot"
A read-only object contains the state of the robot, including the joint angles and the coordinates of the end effector.

b) "Parameters"
This object contains a list of parameters used by the actors from the actor library. Only their values can be modified; however, the object's name or structure no.

c) "Variables"
This object is editable by end-users, and the input is then used for the conditional and loop enabling statements.

The "Parameters" and "Variables" objects can be edited by users using an embedded JSON editor. JSON is a popular ASCII file format, which, over the past 15 years or so, has proven to be versatile and easy to read both by humans and machines[49].

3. Tasks
A workflow can be created by linking a series of actors together. This workflow can then be easily saved via bookmarking option and subsequently reused as a stand alone task or part of the other task. An empty task looks as follows:
Creating a new task is as easy as dragging and dropping necessary actors between

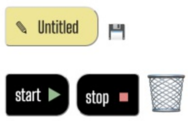


Figure 1.4: *Assembly* empty task

the start and stop buttons, automatically triggering the saving of snapshots of the "Parameters" and "Variables" objects. When loading a task, the content of the global "Parameters" and "Variables" objects is overwritten by the content of the saved snapshots.
When loading a bookmarked task, it will appear as an additional task actor in the actor and task library region. To create compound tasks by reusing, it is essential to load each task individually. The best practice suggests saving an empty task, loading it, and subsequently loading all the other saved tasks within it. This will be the final use case scenario.
A defined task looks as follows: The intention behind task creation is threefold:

a) simplified creation and organization of the robot programs within the Browser

b) reusing tasks via aggregation

c) sharing with other users and exporting tasks effortlessly as Browser bookmarks

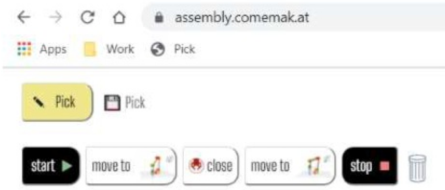


Figure 1.5: *Assembly Pick* task

4. Simulator
   Assembly uses an adapted version of a six Degrees of freedom robot simulator by [42]. The simulator is embedded in the right-hand side of the *Assembly* environment. The "Robot" object contains the axis angles corresponding to a certain TCP pose, to be exact 3 rotational and 3 translational:

   - Cartesian coordinate system (position)
     - X - width
     - Y - height
     - Z - depth
   - Euler rotation angles (orientation)
     - RX - represents a rotation around the X axis
     - RY - represents a rotation around the Y axis
     - RZ - represents a rotation around the Z axis

   The robot can be controlled either using the text input boxes on the upper right side of the page or using the mouse. Tool center point needs to be clicked to move the robot around. This currently only works on Windows systems. The exact position of the robot's TCP is updated in the controls area and the Robot object on the blackboard.
   A snapshot of each robot pose is stored as a thumbnail within the "move to" actor. When used in a workflow, this actor will move the robot to that pose. This way, different poses can be stored by first position the robot to the desired location and then dragging a "move to" actor to the workflow. This corresponds to a teach-in procedure, which is a common way of programming robot behaviors.

5. Robot
   Assembly is set to use both for online and offline robot programming. In the former case, generated *Assembly* code will be sent to a web service which will then trigger low-level robot handling functions. Currently only Universal Robots[35] are supported.
   Connecting to the robot is relatively straightforward and is done by turning the "Connect to robot" switch in the upper part of the simulator. Connecting to an actual robot is out of scope for this thesis.

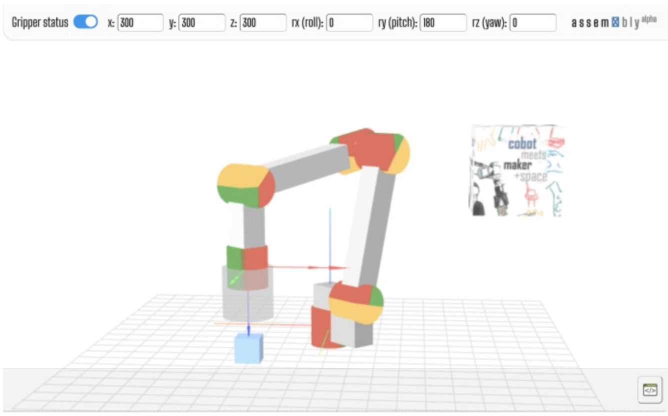


Figure 1.6: *Assembly* robot simulator with connection to the robot

1.2.3 Amazon MTURK[3]

Owned by Amazon and operated under AWS MTURK was launched in 2008 and is currently one of the largest operating micro-task sites[57]. It consists of numerous listed custom webpages nested within the platform, each of them consisting of a unique task. Some tasks may require workers to interact with web pages outside of the MTURK platform. Tasks are called HITs and are being issued by unique users registered as and called requesters. A requester can issue any number of different or same tasks and can set certain filters to only allow suitable workers access to them.
In order to become an official requester, typical account registration steps need to be carried out. The first one needs to get authenticated by providing the full name and additional institution and how often does one plans on using the platform. After that, a credit card needs to be added and subsequently verified by sending and receiving a certain small amount of money.
The entire workflow is depicted on the Figure 1.7:

1. Starting from the lower left circle, marked with "R" for "Requester", the work is published on the crowdsourcing platform in the form of a HIT.
2. MTURK[3] (marked with "MT"), is the crowdsourcing platform of the choice which hosts and shows all available assignments to all potential MTURK workers
3. To complete the first cycle, the MTURK workers (marked with "W"),on one's own initiative, browse and decide to accept the offered assignment. "R" gets notified about every assignment acceptance or rejection.
4. "W" starts solving the assignment immediately or according the given time frame.
5. Subsequently, "W" uploads her results to the platform.
6. "R" gets notified that the solution was uploaded, and she has a limited, previously arranged, time frame to revise the solution.

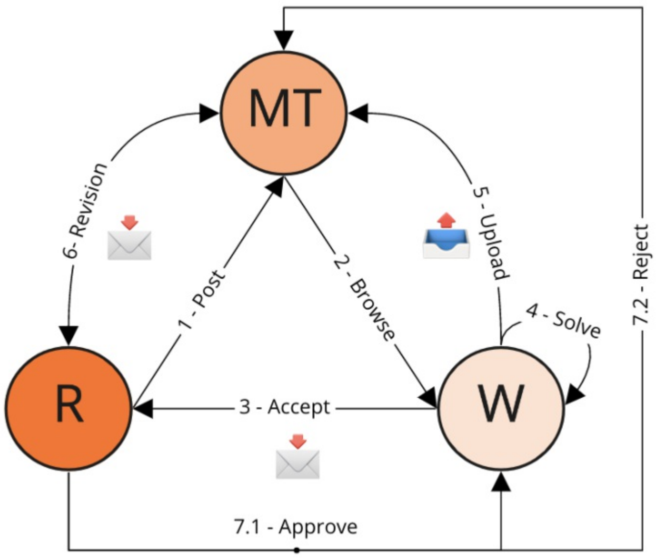


Figure 1.7: MTURK process workflow based on[98]

7. Requester then has two options:

    a) In case of approved assignment, the "R" sends agreed payment to "W", with the non-obligatory possibility of offering and later paying the bonus, and also pays the fee to "MT" for providing the platform.

    b) The other possibility is rejecting the assignment, which can only be authorized by "MT" if a valid argument for doing so was offered.

Next on the list is HIT creation. For this, numerous MTURK templates already exist. However, they are adapted to different use case scenarios (predominantly surveys, image or language recognition, or data collection templates). This meant choosing a customizable template sorted under "Other" and adapt it the best possible to this Thesis needs.

1.2.4 Research questions

It is expected that the findings of this thesis will answer the following research questions:

1. Is the crowdsourcing approach a feasible and repeatable method of evaluation of web-based robotic programming interfaces?

2. To what extent can crowdsourcing support usability evaluation of a web-based robotic programming interface?

3. How diversified is the audience in MTURK both from a demographic and technical expertise point of view?

4. What are the strengths and weaknesses of *Assembly*?

1.2.5 Approach to solving the problem (Definition of work packages)

According to Jakob Nielsen, the two most important techniques for evaluating a UI are heuristic and empirical evaluation[25]. The former represents a theoretical base built upon a predetermined set of rules, which, when followed, results in optimally implemented UI. The latter is more practical and implies testing with real users. This thesis will conduct user studies via crowdsourcing, with the goal of evaluating the usability of *Assembly* programming environment and simultaneously proving that the evaluation indeed can be done effectively using this method.
Nielsen [25] further differentiates between summative and formative usability evaluations. Summative evaluations produce quantitative, predominantly statistical data. Besides backing our findings via comparative analysis or performance measurement tests, quantitative data are useful, e.g., for identifying and rejecting outliers. In the case of crowdsourcing are likely caused by scammers. The results that are more than two standard deviations from the mean are usually excluded in HCI research[66]. On the other hand, the results of formative usability evaluation contribute via qualitative answers on how and why interface users experience certain design or entire or chunk of process flow. For this thesis, both summative and formative usability evaluations are considered equally relevant and will be carried out.
Specifically, six experimental use cases evaluating *Assembly* will be designed and published on MTURK. Drawing on authors in[66], the tasks will progress from elementary mechanical ones, such as moving the robot into any direction, which will demand a small amount of and mental effort and concentration, up to cognitively and timely demanding ones such as creating simple algorithms. Finally, the results will be analyzed and reported in a scientifically sound manner, placing the *Assembly* to the corresponding market position.

1.2.6 Structure of the work (contents of the following chapters)

The thesis is divided into five main chapters, each of them having a different focus. The structure goes as follows:

1. The first chapter will cover the academic literature review of all found relevant papers published in various robotics-related conferences, scientific papers, and journals within the last six years. It begins with the review of robot programming interfaces in general, followed by research on corresponding evaluation methods and practices in this context. Then the review focus will turn to crowdsourcing as a feasible UI evaluation method, finishing with its application in the robotics context.

2. The second chapter is the core of this thesis, delivering a reusable crowdsourced user evaluation methodology framework. All fundamental evaluation steps such

as task specification, validation, execution, evaluation, and discussing results are elaborated and defined in detail.

3. The third chapter follows the exact same methodology and applies it to the actual examples, evaluating *Assembly* over MTURK crowdsourcing platform.

4. The fore-last chapter deals with the analysis of obtained results. Both demographic data and input on *Assembly* UI are analyzed and interpreted using corresponding methods.

5. The last chapter gives a summary of the most critical findings, emphasizing unanswered and questions that were out of scope for this thesis, acknowledges known limitations, and finally offers recommendations for future work and academic research.

CHAPTER 2

Literature review

2.1 Robot programming interface design in general

As explained earlier, collaborative robots offer added supporting value for human workers while working around or directly with them. While being a theoretically valid idea, it doesn't mean that no difficulties occur when designing this interaction, most of the problems arising in previously mentioned robot to human direction. Elementary robot actions come pre-programmed in the same way for all of the robots, obviously without considering the context of use, the industry of application, integration, and synchronization with other tools necessary for a specific process, end-user robotic and/or programming knowledge, and experience. Due to the slow development and acceptance pace, these generic actions were programmed a long time ago by engineers using low-level programming languages, and even the smallest customization requirement depended on domain expert involvement.
There are various proposals to the solution to this problem. For example, the authors in[73] proposed and tested an interaction system that gathers all interaction elements in the shared workspace with the goal of reducing human mental demands caused by distraction occurring due to the need for attention switching in order to perceive and make use of all necessary program components. For this, they developed the interactive Spatial augmented reality, which is a combination of projection and a touch-enabled table, and additionally, they incorporated kin-aesthetic teaching. This allowed the non-expert users to program a robot on rather a high level of abstraction while working within the task context and simultaneously receiving visual feedback and therefore making any additional external devices dispensable.
The concept's potential was tested by conducting user experience tests among regular store workers who were all able to program the robot to prepare parts for a small stool assembly, collaborate with the robot, and adapt the program for an alternative product within a reasonable time despite a number of minor usability issues and system

errors caused by its experimental nature, but without any major issues detected[73]. Study results consisting of the mix of qualitative and quantitative data yielded from the study executed by six regular shop-floor workers of various ages, genders, and technical backgrounds were reported. In our opinion, this is a rather small sample size, which is additionally very uniform due to the fact that all the workers are from the same company. This thesis is offering a lot more diversified palette of experiment participants, which in addition to mentioned demographic factors, presents multicultural, cross-industry, completely scattered work and educational background results from all over the world.
However, there are quite some points of similarity in our approaches, starting from recognizing the problem of non-expert users being in the programmer role, which draws the requirement to keep things on a high level, need for a minimalist, drag and drop, block-based interface to do so and finally the similarity of choice of the use cases (mostly *pick* and *place* action). What is very interesting and different in this paper is the idea of using a mix of the compound and quite complex parametric and non-parametric instructions in order to prevent potential user's overwhelming caused by the need for detailed configuration. For example, the user is abstracted from low-level commands to such a level that even open, and close gripper commands are performed automatically by the robot, as a part of the more complex parametric instructions[73]. Reasonably the question arises how adaptable this concept would be to different industrial environments and how many compound instructions would be needed to be programmed in advance in order to cover all potential use- cases.
The authors in [95] introduced an innovative, cognition based approach to COBOT programming, Spatial augmented reality and multimodal input and output, which they demonstrated on a use case of an imaginary SME. The COBOT had the role of a worker's assistant in preparing assembly parts. The proof of concept system of human interaction with COBOTs was developed with an attempt to simplify the interaction by keeping programming on a high abstraction level and make entire process more fun for the worker, yet maintain it secure and optimize it by keeping both visualization and control elements in one place.
The authors in [69] recognized the omnipresence of the robotic assistants in everyday life, whether at home or in the office; however, they noticed the challenge robots face while attempting to autonomously execute complex tasks in an unstructured environment. For this matter, the authors introduced the idea of involving human operators or, better said, their cognitive skills in the task execution loop. [69] implemented and analyzed four distinct grasping tasks strategies, which stretched from the direct and real-time human operator control of the 6D end-effector gripper pose up to completely autonomous motion and grasp planning from the robot, being simply adjusted or confirmed by the human operator. The results had shown that under the more autonomous assistance the robot was acting, the fewer collisions occurred, and more successful grasp tasks were performed. The framework is known in the literature as the Human-in-the-Loop, and its main goal is reducing the operator's load while increasing the global task efficiency. The idea on how to succeed in this is that instead of letting the robot handle the entire complex task at once, rather divide it into smaller sub-tasks that can either be performed

accurately autonomously or need a relatively effortless human operators contribution. The basic prerequisites for such HITL systems are sub-task identification, the decision between human or robot responsibility, the study of their interplay with, and finally, the analysis of the overall gain inefficiency. The recognition of such sub-tasks is relevant for this thesis, which will identify stand-alone use-cases. Although this thesis employed only the first strategy of the direct human operator control in which the user directly commands the gripper in Cartesian space, the idea to build tasks so that they could be implemented without human assistance in the future remains the same, given that this approach is proven more efficient. The literature review is trying to clear this doubt.
A very interesting and rather simple programming system based on famous Blockly was developed by authors in[61], for mobile service robots, allowing both inexperienced workers (employees at the robotic hardware partner company Savioke[30]) and experienced programmers(external participants with a background in software engineering) to use existing and design new commercially useful and easily deployable use cases. The test groups were separated according to their experience, and the expected outcome was confirmed- users with more experience needed less time to get familiar with the programming interface and create the use cases faster, with a smaller number of minor errors. The error classification is also an interesting thing they did, differing between minor and major ones, the former being easy to spot and fix and the latter representing an obstacle for the correct program execution. Despite the expected outcome of more experienced workers delivering faster and more correct output, the paper succeeded in proving their twofold goals, which are also relevant for this thesis:

1. possible and relatively simple to use for an inexperienced user programming interface
2. complex enough for more advanced users to allow for developing more complex or completely new use cases

One of the interesting takeaways for the future research and something our *Assembly* is already trying to implement is the concept of the "copy and paste literacy" model of end-user programming[79], which would basically allow more experienced users to program templates and/ or more complex cases, which can then be reused and if necessary customized by less experienced ones. The downside of their paper is that CustomPrograms was developed as proprietary software exclusively for one robotic hardware, although the authors[61] claim that it could be easily adaptable. *Assembly* is implemented so that it can connect to all robots.
Building upon the fact that the experts are still necessary to rapidly program robots on the high level of precision and robustness[88] published a study on a prototype of a programming tool on the example of an industrial COBOT in the assembly context. Universally speaking, it could be talked about two essentially different approaches to robot programming:

- Lead- through programming
  In this approach, the programmer owns direct control over robot teaching, conduct-

ing it manually over some kind of joystick, which is most firmly fixed on the robot hardware itself. This way, the programmer can either lead the robot over the path to the desired target or define the desired path by setting consecutive waypoints along the path.[46] The entire process is recorded and can be replayed any time if necessary. However, it is important to understand that the robot can ever only repeat exactly the same task.
Due to the low complexity and repetitive nature of tasks, this approach predominantly found the application in the industrial robotics field.

- Programming by demonstration (PBD) or Learning from demonstration (LFD)
  As the name itself says, this approach is demonstration-based, allowing a user to either manually guide a tool or end effector in the desired direction by guiding an equivalently structured haptic tool or by teleoperating the robot using an array of motion sensors.
  This approach is less apt for industrial robotics, which favors single demonstration, more suitable for complex and more appropriate for customized tasks, and is, therefore, more applied in service robotics.

The authors in [61] proved that Scratch-like[31] block-based visual programming languages significantly reduce the programming effort for the end-user. The trend to provide template skills for generally known applications along with industrial robots seem to be taking off. Pick and place template is already included in Universal robots[35] package, Franka Emika[15] provides download possibility for pre-programmed skills, and Kuka[20] seems to be investing a lot of effort in launching their proprietary software.
The possibility of instant and effortless, out-of-the-box usage of these robots is what makes them interesting for the end-user; however, the issue of developing further, individually customized, and probably more complex reusable skills still remains. The authors in [88] tried to solve that challenge with their proposed high-level simplified interface, but it is still in the prototyping phase.
The authors in [90] introduced a new domain-specific programming language for robot programming on distinct abstraction levels, each of which is adapted to the end-users level of programming knowledge. They recognized the upcoming high demand for highly customized ("different level of detail ") robotics in fields such as lot size one goods and also the difficulty of teaching these low-level programming skills to the end workers who mostly possess little or no programming knowledge. Supporting a generic GUI in Eclipse, this programming language called LightRocks demonstrated high acceptance and success level according to two-run test cases. Nonetheless, no detailed information on tested users was given; merely the claim that the test cases could be run efficiently even by inexperienced users was made.
Previously mentioned robotic programming methods kin-aesthetic teaching and visual programming are often jointly implemented in PBD, which according to [38] acts like a hybrid of the two. Numerous robotic programming interface design opportunities were presented in their paper, which is based on a user study on the example of Universal Robots' Polyscope. Although the exact user study procedure wasn't defined, the partici-

pants' demographic and educational data were as differentiated as in this thesis, with one significant difference, and that is that the majority of them were females. The findings yielded many design opportunities obviously directly related to Polyscope but easily implemented on any other robotic programming interface. We consider their[38] most important findings relevant and noteworthy:

- Need for graphical simulation as early as possible in the programming so that unexpected outcomes such as completely wrong routes or even collisions can be avoided. Kinesthetic teaching is quite a complicated task that puts a high mental load on participants, even more, if requiring complicated mental models instead of presenting the end-users with them.
- The next point is tightly related to the previous one, and it is the idea of real-time visualization (of, for example, the path) of corrections during programming so that the participants can check their work as they advance, and not only the final result. This could save some unnecessary starts from scratch.
- Continuing with a similar issue, the need for an "Undo" button is emphasized. The existence thereof could also help avoid some program restarts and spare the participant of the unavoidable frustration that inevitably comes when losing their entire progress in order to make some small correction.
- This point is about the GUI design and calls for as simple as possible interface. The participants stated that any irrelevant or redundant information should be kept off the screen.
- Another interesting design point is about the need for colors in order to, for example, group the actions or structure the directions, but also to avoid boredom while working with a monochrome interface. However, the color mechanism mustn't be heavily relied upon. Around 8% of all males and around 0.5% of all female population are suffering from some kind of color blindness, with 99% of them all suffering from red-green color blindness[32].

The authors in [64] were the only ones who tackled an interesting topic of connecting and controlling multiple physical robots over one GUI. Unfortunately, just like the majority of other authors dealing with robotic programming, they did a very technical low-level presentation of the UI. No screenshots or word description of it were provided, merely the layered architectural framework behind was modeled. This leads to the conclusion that the common problem occurred again, and that is having a very powerful and useful tool, which in the non-adapted presentation won't be of much use once it gets into the non-expert's or non-programmers hands.
Another example for a different understanding of what is easy to use robotic programming interface was given by[99], where they presented the prototype model of an automated and trainable robotic arm with the memory, which is controlled and managed from MATLAB GUI. The GUI itself is divided into three parts:

- Axes
  - for monitoring the robotic arm movement via cameras
- Training interface
  - the main function is training the robotic arm by adjusting the slider
- Programming interface
  - consisting of 6 main buttons, it is used to control the robotic arm

Although MATLAB is known for its simplicity, ease of data presentation, and for being very tolerant in respect to syntax errors[21], it is still a programming language that is known in rather academic circles and definitely does request some kind of schooling or documentation reading before using it. Therefore it might be the perfect choice for the audience with academic background, most of whom will have probably already heard of or worked with it, but it cannot be used immediately by an average shop floor worker. What is more, the authors stated that the app was easy to use but gave no description of the target audience.
Another proof for this was given by authors in [59], who created an open-source library called ros4mat, which serves as an interface between ROS and MATLAB[21]. The idea was to connect the best of the two worlds, the former currently maturing into the standard for open-source robotics platform, and the latter already being renowned as the standard programming platform designed specifically for engineers and scientists to analyze and design systems and products that transform our world[21]. The motivation behind it was an important one, relieving the user from the burden of complex robotic hardware integration into some software control environment. This interface was tested by students in a computer science course and resulted in positive feedback[59]. This still doesn't solve our issue of creating an environment for a broader specter of users, not only the ones with an academic background in computer science.
The authors in [54] tried to address that problem by proposing a user-friendly mobile collaborative educational platform using visual programming language adapted for both beginner and advanced users with a focus on robotics. The GUI was done fairly simply via scratch, drag, and drop basedVisual programming language, which automatically translates to JSON, allowing the end-user more time to concentrate on programming logic instead of syntax[54]. This platform allows students to implement various projects, and although no official assessment was done and there is a lot of future work to be done, it received positive feedback.
The authors in [100] tackled the known problem of numerous manufacturer-specific robotic programming environments and proposed the universally accessible ready to use Information control systems programming and simulation solution with modular architecture and non-need for specific software. The former characteristic supports scaling, unlike most of the proprietary software programs, and the latter one allows for seamless integration, regardless of certain software or hardware. This framework has been developed but not assessed, leaving the known problematic HCI substantial.
This is where [102] presented an architecture and implementation of the non-proprietary,

scalable via plug-and-play technology robotic GUI especially developed bearing non-programming individuals in mind. Their solution combines once again ROS and Snap (JavaScript-based version of scratch) and is proved on two use cases, one medical and one industrial. The medical one focused on autism therapy, and its GUI was modified after a series of need-finding studies with therapists. What is interesting is that the GUI indeed was a result of modified existing versions, adding missing features in an intuitive way, multiple device support (with easy implementation) from hardware and software point of view, and finally active community development functionality. Their solution is in Polish language and very niche-specific but shows how easily it can be done when existing issues are tackled directly with field experts, independently of their IT literacy. The room for future work definitely exists in the context of creating a larger set of robot behaviors, where the challenge remains of how to cover all the possible use cases and still be left with the intuitive and user-friendly interface. Since covering all of the use cases is virtually impossible, this challenge is reformulated in providing an intuitive, user-friendly interface for simple programming of any additionally necessary reusable skills.

2.2 Crowdsourcing as a powerful evaluation tool and its application in robotics

As of lately and strongly reinforced by Internet usage, the meaning behind the term of the "power of the crowd"[10] has demonstrated its potential over and over again, with different systems being utilized to support cooperative knowledge in undertakings extending from innovation over customer satisfaction questionnaires to image recognition and much more. The term crowdsourcing was coined quite far back in 2006, with Jeff Howes' official definition of it being "the act of a company or institution taking a function once performed by employees and outsourcing it to an undefined (and generally large) network of people in the form of an open call"[60]. Obviously, this network of people is quite distinct to the actual internal employees of a given company since it is composed of literally any individual with access to the internet, who fits the requirements if there are some, working under their own conditions, customarily alone and not knowing the rest of the crowd, receiving typically small remuneration for their work, if it gets accepted. This way, the companies can outsource any task they don't have internal resources for, need a broader opinion on, or even don't have the expertise in. Based on that emerging phenomenon, many web-based platforms and applications were created to host the solution, and at that speed that the research about their potential and limitations is still lacking.
Talking about UX evaluation we can distinguish between:

- Traditional user studies
  These studies are being conducted on the filed or in the lab, requiring both study participants and evaluators to come to a certain location. This obviously takes more time to perform and means higher costs, since the required effort is higher, having lower number of participants as a consequence. Applying this in the context

of evaluation of the robotic programming interfaces, additional limitations in terms of finding knowledgeable or voluntary study participants, robot access, safety and platform availability emerge.
As [91] correctly defined, the typical research cycle passes through four main stages, starting with the method formulation, method implementation, conducting user study, and finally presentation of the results. The last step is where the next delay due to lack of time mostly happens, because neither presented results nor lessons learned don't really mean much if not implemented into the final product[91].

- Crowdsourcing user studies
  Following[91], in order to progress in robotics, it is indispensable to build professional-grade robots and create user evaluation research appropriate for and available to the masses. Conducting web evaluations seem like obvious answer to this issue on hand, benefiting even more using crowdsourcing platforms.

The authors in [98] did an experimental investigation of two distinct crowdsourced design evaluation approaches, one being the usual free evaluation in the crowdsourcing environment and the other using specially developed CDEC. The outcomes of both approaches were finally benchmarked against a standard expert evaluation with a board of experienced designers. It is interesting that the outcome suggests that their CDEC approach produces design rankings with a strong correlation to those of the expert board. [98] created an assessment methodology which proves the effectiveness of crowdsourcing in the context of generation and evaluation of new design solutions. Drawing upon and adapting their methodology to the UI evaluation instead of design, the methodology for this thesis was set.
The feasibility of UI evaluation via crowdsourcing platforms was discussed in[66]. Three distinguished, cognitively non-demanding, predominantly mechanical interface design tasks were chosen:

1. Bubble Cursor[55]
   This design offered an enhancement of standard area cursors by dynamically resizing its activation area, relying upon the vicinity of encompassing targets, to such an extent that just one objective is selectable whenever. Bubble cursor task was set as a baseline because of its simplicity.

2. Split Menus[83]
   Implemented back in 1994, split menus is basically a term for placing a certain number of items with higher usage frequency at the top of the menu, instead of arranging the menu alphabetically as was the case before. This proved to raise efficiency in browsing time by up to 58%.

3. Split Interfaces[53]
   This term was coined when the authors in[53] offered an analysis of two experiments comparing three adaptive user interfaces to a non-adaptive baseline. The main

idea of the paper was to find the underlying reasons which make some adaptive interfaces effective and easier to use than others.

After one lab and two crowdsourcing performance experiments, the result analysis did not prove any considerable differences.[66] analyzed parameters such as the raw task completion times, error rates or consistency. Herewith they made a strong implication that MTURK could offer an efficient and valuable alternative for conducting performance evaluations of UIs.

There is increasing evidence in behavioral economics that MTURK participants perform just like or show no significant difference to lab-based participants.[66]
The crowdsourcing phenomenon is nothing new in the context of robotics. Various authors already addressed numerous and distinctive issues where this concept can be very convenient. We will try to give an overview of some of them.
In [48] work, a web interface was created and used to invite crowdsourcing workers to get a mobile robot out of a maze and simultaneously test different robot teleoperation and training methods. The obtained data were used to test if users performed better teleoperating and training the robot with:

- an unfiltered live camera feed from the robot
- filtered image data typically used in computer vision algorithms

Results proved that the filtered image data ware sufficient to train the robot.
The authors in [87] presented a framework for robot supervision through MTURK. The goal of the work was to design a model of the autonomous systems building upon asynchronous human computation through crowdsourcing, for which image labeling, object clustering, and the final model selection tasks were created[87]. All of the tasks recognized the importance of substantial human feedback as a strong foundation and dispute creating a completely automatic system. The authors[87] as well recognized various challenges in the context of algorithm creation, UI design, QA policies, and teaching. This thesis will particularly focus on the UI design creation challenge.
Another great crowdsourcing-based yet somewhat distinct idea of collecting substantial human input over already existing online communities was described by [51]. The author put immense online social network community to use and challenged the Twitter community to categorize and define actions performed by humans in a typical HRI interaction. This has a twofold advantage over MTURK, the first one being the fact that it doesn't cost anything. The second advantage is that the Twitter community can be targeted more straightforward because the followers of certain Tweet accounts about robotics customarily have some kind of knowledge or relation to it, therefore making their input more reliable than the one from random MTURK workers. At first glance, the goal of the paper seems very high-reaching, intending to teach robots to learn and perform without direct instructions from a dedicated user. This would allow daily HRI

interaction to be far more enjoyable for the human user because they won't have to spend time on teaching their robot everything it knows, but rather reap the benefits of preconfigured capabilities and therefore be reminded why they got the robot in the first place. To try and contribute to this high goal, the author[51] developed quite a simple framework with a basic yet necessary and moreover logical sequence of events:

1. First, the robot will observe a human performing simple physical actions "by quantitatively recording the X-Y coordinates of the designated extremities using skeletal tracking data acquired from the sensor "[51].
2. Next, the ahead gathered data will be used to create a video of the performed action, which will be uploaded to the dedicated robot's Twitter account.
3. Then, the account followers will be asked for a description of the actions they see in the video and have 24 hours to do so before the response analysis starts.
4. Finally, recorded parameters of every action will be used as input into a multi-class support vector machine classification algorithm, which is supposed to enable the robot to recognize the action at a future point in time.

The [51] managed to present the potential of social media as a source of data collection, knowledge-base, and information gathering, all of them contributing to faster robot learning opportunities via crowdsourcing. Nonetheless, more complex actions will demand more sophisticated machine learning algorithms. Additionally, for this method to make a difference, an immense number of tasks need to be revised so that robot has a higher probability of finding itself in a known environment. The final concern is the statistical significance of Twitter users' input since it is on a free basis, and additionally, it is impossible to analyze the crowd impact- how and if seeing other followers' responses impact each user's individual response? None of the two concerns are relevant while using MTURK.
Taking into account the fact that the most prevalent and, for that matter, natural interaction with robots happens via verbal commands (think of the typical home assistant systems such as Alexa from Amazon[2], Siri on Apple devices[5] or Google assistant[17] on respective devices), the authors in [89] took research to another level, using crowdsourcing to collect the significantly large amounts of training data needed to build and train speech models. They asked the workers crowd to watch a short video of an assembly robot, more specifically a forklift executing an action. The workers were subsequently pleaded to type a one-sentence command that described the robot's action.
Contrary to this thesis, which uses crowdsourcing for fairly simple and mentally non-demanding tasks, there are authors such as [94], who relied on this technique to provide solutions to computationally difficult problems and train algorithms. What is interesting in her work, however, is that she proposed taking advantage of anyhow existing large internet gamers community and channeling their time and mental effort toward different kinds of games, called Games with a purpose. [94] recommended general design principles

for the development and evaluation of such a class of games where players would perform tasks hard or impossible for a computer to perform, casually while gaming. The set of guidelines for GWAP creation serve as the first known general method for seamlessly integrating computation and game-play[94], setting a ground for an extensive future research in this field.
The authors in[45] took GWAP to an even higher level by combining virtual and physical approaches and what is more in two essentially distinct domains. The former scenario was realized over a two-human-player online game called "Mars escape" and used to gather a large-scale set of unstructured HRI data, which then served as input to train a physical robot in a real-life museum scenario. The objective was to answer the question of the records of human-to-human interaction gathered in a virtual world can be used to design natural and robust human to robot interactive behavior in distinctive real-world environments[45], while performing similar tasks. The "power of masses" delivered by crowdsourcing was an ideal approach here so as to produce less biased and more diversified inputs from distinct users, just like in this thesis. The results demonstrated similar behavioral patterns in both online and offline scenarios, therefore underlining once again the importance of the crowdsourcing approach in robotics.
The authors in [47] dealt with the question of a new goal-based imitation learning framework which utilizes both physical user input and crowdsourcing as a major source of demonstration input data, on a simple scenario where the robot learns to build 2D object models on a table from basic building blocks. As already discussed in [93], goal-based imitation learning has the advantage over other imitation learning approaches because it focuses on achieving the final task goal instead of solely imitating the human trajectories. This is important because robots obviously have end actuators distinct to the human ones and therefore face different obstacles, which call for different actions to the human ones. Crowdsourcing, on the other hand, has a clear advantage over traditional imitation learning approaches in the context of data and its availability by overcoming the issue of sparse, expensive, and time-consuming human demonstrations. The object models to be built were selected rather simple (car, person, house, flower, fish, snake, turtle, and chick). The eight classes were created to avoid the danger of making goal inference insignificant, although it was proven that a low task difficulty does not necessarily lead to a model that is desirable or is similar to the user's demonstration[47]. The robot was given 24 hours for data collection and learning time. The results exhibit two proofs, firstly that the robot indeed can use crowdsourcing data to grasp simple 2D models, and secondly that the learning effects are better than the ones emerging from completely local user-provided object models. This paper did leave some open issues, such as crowdsourcing quality control, the question if the framework is applicable to more challenging models (e.g.more dimensions), more dangerous tasks such as assembly and tool use, the possibility of self-learning and autonomous decision-taking in judgments such as whether to crowdsource or ask the physical user for support.
As this thesis aims to develop a framework for usability evaluation of web-based robotic programming environments, [92] designed an entire web-based framework called Robot management system (RMS) built in a robot-, laboratory-, and interface-independent

manner, which was created to greatly reduce the overhead and costs of running remote user studies, which also involve remote control of the robot[91].
In their work, they analyzed metrics across three user study conditions:

1. Co-present group
   This group is the traditional user study group, in which the participants are invited to directly to the laboratory, where they are asked to control the physical robot over a standing computer placed in the same area controlled the robot using a desktop computer's on-screen interface located in the same room as the robot itself. The biggest alleged advantage these participants had over the rest is that they could directly and in real-time witness the actions of the robot.

2. In-lab group
   The participants of this group were given a mix of conditions between the present and remote groups. Like the former, they were invited to the laboratory under the same condition apart from the room where they were situated in the one different from the physical robot.

3. Remote group
   As the name itself says, this group consisted of remote participants only, working from their own devices, from their homes. The idea with this group was to test how completely anonymous users interacted with the system. This is the equivalent of crowdsourcing workers.

Their findings show non-significant statistical differences between distinctive usability patterns across the three groups, clearly favoring the web-based crowdsourcing techniques for the considerable fragment of HRI evaluation studies. The most interesting finding is the one which showed no clear advantage of the co-present group, although it was inherent to expect this. Nonetheless, in addition to the disclaimer that the results require quite attentive analysis, the author [92] discloses the shortcomings of this method by identifying the context in which RMS wouldn't be as useful, which is most of the techniques and studies requiring either physical interaction (e.g., kinesthetic teaching and haptic input addressed in[68] which is rather interesting but quite out of scope of this thesis), sensing or observing of the user, and also studies that have strict time requirements, be it real-time or parallel study execution. Another downside of this system is that it requires supervision, whereas "pure" crowdsourcing doesn't. In spite of all of that, this work [92] presents a very important contribution in HRI research, which offers a tool, a RMS system, which is intended to grant researchers with the opportunity for quick and feasible parameter tuning, substantial training data sets gain by keeping a large, consistent stream of users, leading to rapid algorithm development and testing, all of which were previously unimaginable, especially at the same time.

CHAPTER 3

Crowdsourced evaluation methodology framework

The basis for our framework is the iterative User centered design process for interactive systems described by the ISO standard 9241- 210[36]:

As visible from figure 3.1, the following steps can be adhered to and re-iterated if

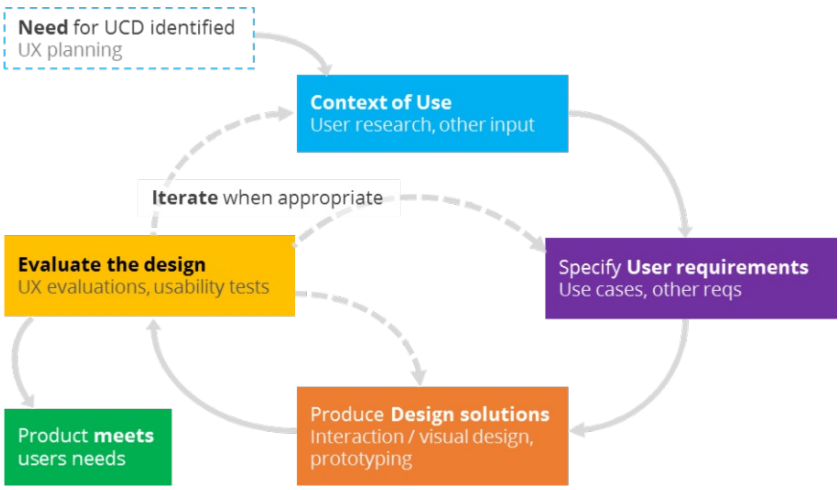


Figure 3.1: UCD process according to the ISO 9241-210 standard[36]

necessary[36]:

1. Firstly, the need for UCD task is identified through input and requirements from various stakeholders and parallel to that all work on UX such as resources and time are planned. This thesis had a clear UCD task based on *Assembly*, the robotic

programming environment developed at the University of Technology in Vienna, as a part of COMEMAK project funded by FFG.

2. Secondly, the context of use is specified based on user research or other input. Since the main goal of the COMEMAK project is ensuring all interested stakeholders access to COBOTs in makerspaces and additionally addressing other issues regarding robots, the context of the use of *Assembly* here is clearly defined.

3. The third step handles user requirements specification. It is suggested to define and document relevant use cases in order to model the specifications. The use cases should start simple to allow the user to get to know and use the program and then gradually upgrading the complexity. Business relevance must not be lost out of sight here.
   One idea would be to use simple tasks as a part of the assessment test to get only qualified users to participate in the entire crowdsourcing project and later raise the difficulty bar. If every participant can solve all the tasks, then the use cases should be made more difficult; otherwise, the results will not be within the scale.

4. The fourth step delivers design solutions, such as UX concepts, tutorials, interaction evaluations, and prototyping.

5. Finally, the UX design is evaluated by testers or real users. This thesis will focus only on crowdsourcing results. Normally, if the users' design or functional needs are met, the entire UCD process is done, else ways one or more new iterations to any of the previous steps are necessary.

6. The end result should be ISO 9241-210 standard[36] approved final product.

ISO standard 9241- 210 was implemented as a strong foundation and enriched by the framework drawing on authors in [98]. The combination of the two generated a specific set of steps customized to the task at hand. The main goal was to provide a conclusive model and a generic framework for the entire process of crowdsourcing evaluation of web-based programming environments. As a result, the methodological approach for this thesis was created.
Minding the color code, each extension of ISO process is marked at the spot, while the own model continues according to its own flow. Based on the top down approach, our methodology is divided into 4 main stages, which are further refined into more detailed sub-stages where necessary. Following sections will be dedicated to each stage in detail.

3.1 Task specification

3.1.1 Evaluation platform selection

The very first task to do is choosing the correct platform depending on the nature of the task. The concept of crowdsourcing is much more common in the context of

idea development or especially design (website, graphic, product). The pages such as DesignHill[12], Designcrowd[13], 99Designs[1], or Guerra Creativa[18] are examples of forerunners in this categories.
For the less specific tasks or simply because they cover a broader range of services, other pages are definitely a better choice. In this section pages such as microworkers[22], Crowdspring[11] or MTURK are the most famous examples.

3.1.2 Submission tool selection

In case the assignment demands any results uploading, if not already included in the chosen platform's service, an external web page with any cloud storage system will be necessary. MTURK claimed to offer the possibility of file uploading, but it will be reviewed later, this turned out to be a piece of outdated information.

3.1.3 Methodology selection

According to [98], there are two main generic crowd design methodologies that need to be decided upon at the high-level stage of creating the task:

1. Iterative improvements
   Especially suitable for the design assignments, these tasks are designed as competitions so that the workers are building upon each other's solutions, where they are anonymously shown previous results from other workers and are basically asked to improve or build upon them.

2. Non-iterative (linear) competition
   On the other hand, this kind of competition can be performed as a single or multistage task, which either compensates workers with rank payments or the best-rated result takes the biggest financial award.

3.1.4 Crowd selection

The participants' profile can but does not have to be defined, allowing every individual with internet access to participate in the task and consequently making each participant's evaluation uniquely diversified.
In case that the requester decides to define the participants' profile, some of the typical filters that she could use are participants' location, age, gender, education level, experience with the matter in question, user rating on the selected platform (e.g., number of previous successfully solved tasks or even the quality of the solution).
Additionally, being intentional and inclusive with participant samples generally has positive effects on products or designs since the input comes from diverse participants profiles who are likely to see and do things differently [85]. This is one of the major arguments in favor of leveraging crowdsourcing platforms for user studies. Nevertheless, even if user diversity is desirable, crowd selection needs to be performed in order to obtain statistically significant results.

3.2 Validation

This stage is dealing with the validation of the task specification, its relevance, and finally, the affirmation of the test run. During this phase, it can happen that the requester realizes that for as relevant as their task initially seemed, there currently exist neither applicable use cases nor appropriate evaluation methods via crowdsourcing.
Another realization could happen a little later in the process, more specifically after having designed the use cases. In case they turn out to be very rudimentary and seemingly too easy to solve, the task definition should be questioned so as not to waste funds. However, if the use cases turn out too complex even for the designer, the task should be rethought and possibly broken down into simpler tasks.

3.2.1 Test run

Ahead of uploading an assignment to the crowdsourcing platform, it is recommendable to run a small pilot test with someone inside of the organization, preferably familiar with the topic yet not too involved with the assignment on the hand. This gives an opportunity to identify the aspects of the design or study that are obvious to the requester yet may be missing from study experience[85].
The evaluation basis is creating robust test cases based on which initial test runs will be carried out. The results of this run are the best indicator of what is missing, is possibly miss-formulated, redundant, or irrelevant. One or more test runs can be undertaken; the decision on that can be based on the relevance of the participants' input. For example, should it start being too repetitive or irrelevant, it is time to finish the test run phase. It is suggested that these are carried out on smaller participants samples, offering less remuneration and possibly even explaining in the instructions that this is the test run, since this has a psychological effect on the participants, encouraging them with their remarks since this is not the final state of the things.
It is noteworthy that crowdsourcing kind of usability testing is not less work than in-person testing; quite on the contrary, it actually requires more planning and more communication in order to make sure the tools and devices that will be worked with are working as expected[85].

3.2.2 Use cases definition

Use case diagrams are officially defined by [34] as behavior diagrams which are used to describe a specific set of actions that one or more systems should or can perform in collaboration when triggered by actors, which are external users of that system. Each use case is meant to offer some perceptible and beneficial results to any system's stakeholder. In this particular context of robotics, it is important to understand that the evaluation participants will most probably be common people with limited or no programming or robotic background and even more restricted attention spans since they are working on their own behind their displays. Out of [87] experience, simplifying complex and difficult to describe problems into smaller, simpler, and easier to describe sub-problems is the key

to lower the rate of unusable participants output.
It is therefore important to define them reasonably so as to save both money and time. The idea is to try and cover the essence of the certain functionality or flow in as short and simple as a possible individual use case. Ideally, the use cases should iteratively grow upon each other, starting with rather simple ones to get participants familiar with the system and get comfortable with using it so that the probability of solving more complex use cases gets higher.
Ongoing research in the area investigates how to define tasks in a way that enables the crowd to accomplish complex and expert-level work. Various approaches and workflows can be used to break a complex piece of work into approachable parts and can also use the crowd to check the quality of its own work[101].

3.2.3 Instructions writing

This point is crucial in remote testing methods because they are mostly not moderated, so it is highly important to write the instructions as clear and straightforward as possible so as to avoid misunderstandings, raising questions, or double meaning.
Basic best practices here are quite self-explanatory. It is important to keep instructions simple, using basic wording or exclusively technical vocabulary, the latter only if the participants are going to be filtered accordingly. Any ambiguity has to be dropped. Long and complex sentences should be avoided because they provide more opportunities to make mistakes, impede the understanding due to complexity and tire the participants faster. Concerning the last point, it is really important to try to keep the participants' attention captivated and motivation high. One of the suggestions for doing so is mixing the type of feedback requests between audio or video output and textual, which can vary between open-ended or single, or multiple-choice questions. The same applies to the input from the requesters' side.
As the author in [85] stated, usability test should only be as long as it needs to be to gain confidence in results. In case the requester is uncertain of the test length, she should pilot it and collect feedback. Running multiple smaller separate tests is recommended over trying to get useful feedback from the participants cognitively exhausted from the excessive assignment's length.[85].

3.2.4 Demographic questionnaire

In every study, a demographic questionnaire is a must in order to get a clear picture of the participants, understand their motivation, and finally assign respective relevance to their input better. In crowdsourced evaluation, this point is even more eminent.
Demographic questions are usually put on the beginning or at the very end of the study and ask participants about their background. There is a variety of questions that can be asked, but the idea is to restrict them so as not to overload the participants who are already occupied enough with the actual assignment yet gather useful input.

3.2.5 Scam detection

High assignment availability and even higher reachability of the diverse participant sample in combination with rather low payment logically also have their downsides, which manifest the strongest in terms of scams. The workers are trying to finish as many assignments in the shortest amount of time possible in order to earn as much as possible.

3.2.6 Payment

Crowdsourcing is infamous for rather low payments because it mostly provides a quick method for solving numerous, although rather basic tasks, by as many participants as possible. According to [56], mean hourly wages per worker on MTURK lie between 3.13/h and 3.48/h, while the median wage lies between 1.77/h and 2.11/h. That is below any imaginable standard for Austria.
Rentability and speed are significant factors from the requester's point of view for crowdsourcing to be chosen over other evaluation methods, but it is important to find the right balance here. If paying too little, it will probably happen that most of the participants will have rather low educational background profile, coming mostly from underdeveloped countries, or finally, not enough participants applying at all. On the other hand, paying too much does not guarantee the right crowd. The problems change their nature, such as attracting wrong participants profiles with increased scamming percentage because of the higher payoff or finally no advantage for the requester since the costs do not offer benefit over other more conventional methods. The payment should be sufficient to provide participants with enough motivation for solving the assignment properly, but not too much so that one of the benefits of the crowdsourcing evaluation method goes to waste.
For example, the authors in [97] reported strong correlation between the payment (worker/task) and the quantity of the delivered designs per worker, but a weak one between the payment and the result's quality. This is confirmed by [84], adding that higher payment definitely incentivizes more which results in quicker recruitment and higher willingness to deliver more output. They also state that there is some weak proof that bonuses and respectively penalties do contribute to higher overall output quality.

3.2.7 Test run QA

As mentioned at the beginning of this stage, each and every test run should be evaluated, and only after test runs yield valuable results and potential issues on that model are fixed the real run should be executed. Until that is the case, iterative test re-runs with implemented changes based on participants' input are highly recommended.

3.3 Execution

Generally speaking, usability evaluation task execution can be:

- Moderated
  Meaning that there is a test facilitator, she interacts in real-time with each participant individually or entire group located at a remote location[58]. This is easily realized via any video conferencing or remote application sharing tool such as WebEx[39].

- Non-moderated
  The most obvious advantage of non-moderated usability testing is that no moderator nor location (such as lab spot) are necessary, which automatically implies lower cost and quicker implementation. This implies that participants can complete the assignments at the time and place of their choosing, which is a definition of an uncontrolled environment. This resembles more closely to how a real user would interact with the program in their natural environment and therefore generates more authentic results.
  Like the remoteness factor, non moderated execution contributes to the speed of result collection, merely because it is not necessary to find a moderator or set a location and time for each session.
  Nonetheless, this usability testing execution method comes with disadvantages, in the first place in the form of less control over the entire test environment and evaluation procedure as a whole. In order to minimize the risk of misunderstanding and the need to ask questions during the evaluation itself, a lot of thought and effort needs to be invested in the instructions writing phase. As the author in [85] suggests writing an introductory context and guidelines at the beginning of the test to inform the participant of the session's goal, give them relevant contextual information that clears the most obvious doubts out.
  [58] proved in their work that both study participant groups delivered rather similar arguments, with non- moderating users reporting a higher percentage of high-relevance input. This could be the result of the positive effect of not being monitored and, therefore, more relaxed and creative.

3.4 Evaluation

Establishing appropriate evaluation criteria is a crucial task. If this is not done properly, even with all the previous steps done correctly, it is highly probable to end up with a big amount of pretty useless data. It should be clear for the requester what success looks like for her own product testing and also have explicit ways to identify if and when participants were or were not successful[85].

- Qualitative analysis
  The qualitative data is usually obtained using individual, subjective satisfaction questionnaires[76] or observations and analysis of participants' input, in the form of task solutions or open-ended answers. Based on requestors own expertise in UX and bearing in mind programs known UX difficulties and limitations, acknowledging

and possibly comparing those insights with the participants' output should lead to clear ideas if a certain part of the UI needs to be redesigned[28].

- Quantitative analysis
  The quantitative data includes all numerical data, such as the time necessary to complete a task, amount of usability defects identified[76] or task completion rate. Crowdsourced evaluation works hand in hand with quantitative testing because it automatically measures the performance of the design and tracks usability metrics easily since most of the data is calculated and reported automatically[85]. Additionally, another method could be implemented within the task. There is a large number of quantitative UX research methods available, and the requestor needs to decide on the aptest one for a certain project. Some of the most famous ones according to to[27] are the following:
  - Quantitative Usability Testing (Benchmarking)
  - Web Analytics (or App Analytics)
  - A/B Testing or Multivariate Testing
  - Card Sorting
  - Tree Testing
  - Clustering Qualitative Comments
  - Desirability Studies
  - Eyetracking Testing

  Finally it is also suggested to employ known usability quantification post experiment survey so as to evaluate the data quicker. Two popular examples for this would be:
  1. Post-Study System Usability Questionnaire (PSSUQ)[26]
  2. System Usability Scale (SUS)[44]

3.5 Results

This final step should deliver all smaller change requests or larger redesign requirements. These should ideally be implemented immediately on the system version, which was actually at the moment of testing. Subsequently, the same use cases should be run through by the requester or the same group of people from inside the organization who were gathered for the pilot study during the test run.
At this stage, the UI should definitely show improvements. If that is not the case, either some part of the methodology was not implemented or evaluated correctly in case that the same errors repeat. In case of newly emerged errors, another user study is suggested.

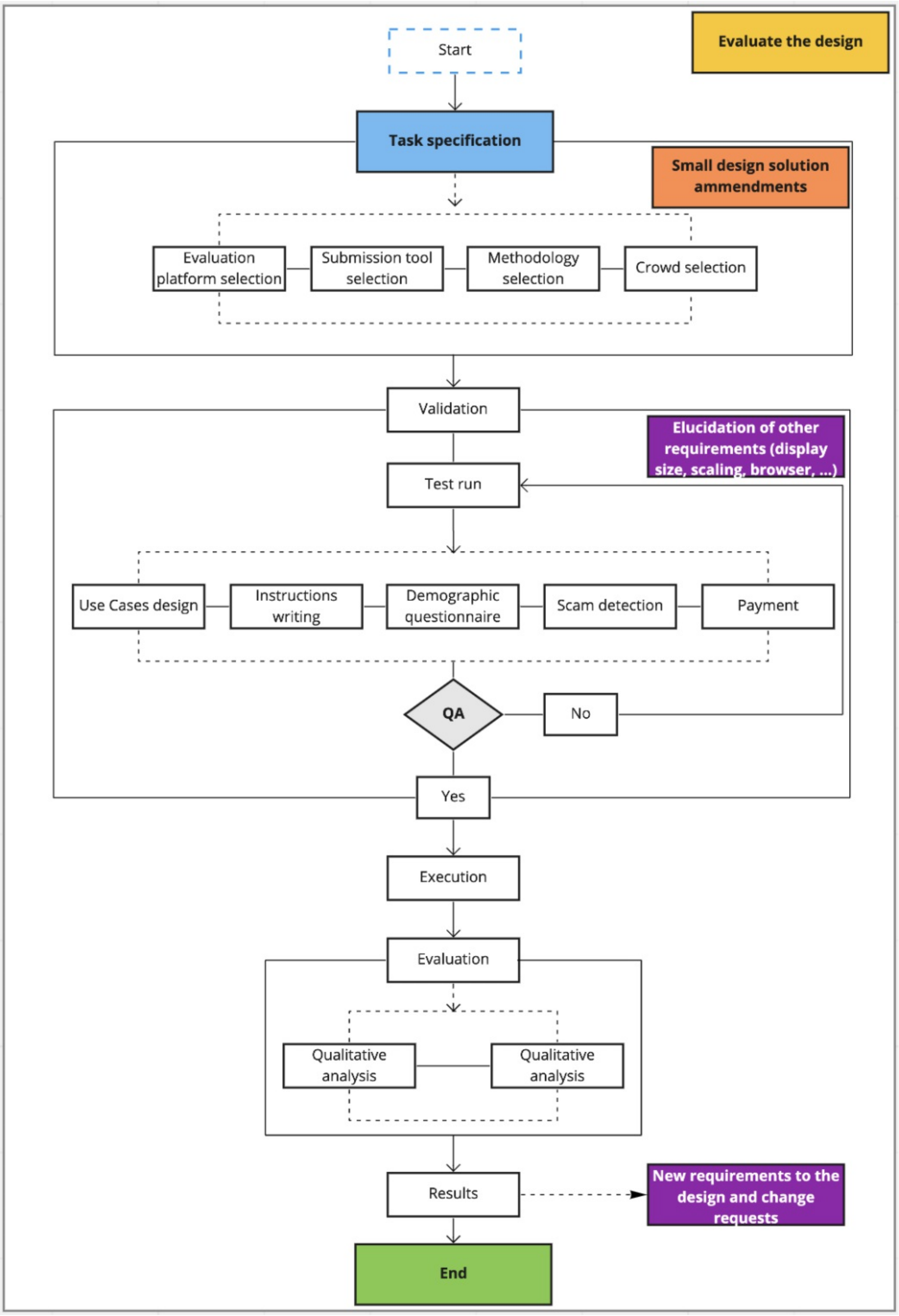


Figure 3.2: Crowdsourced evaluation methodology framework based on[98]

CHAPTER 4

Application of the framework on example of evaluation of *Assembly* on MTURK

In this chapter, the previously defined methodology framework will be applied, following each defined step.

4.1 Task specification: Evaluation of web-based robotic programming interface

Talking about UI evaluation, in general, is a broad subject that would be difficult to tackle as part of one master thesis. This thesis focuses on the robotics context, more specifically the earlier introduced web-based robotic programming environment *Assembly*. This issue has high practical relevance because it deals with the daily business routines of many workers, and its correct application should contribute to more straightforward, more convenient, and adapted to broader audience usage of robotic programming environments.

4.1.1 Evaluation platform selection: Amazon MTURK[3]

As described in the introduction, we chose Amazon MTURK as the crowdsourcing evaluation platform.

4.1.2 Submission tool selection: *ufile.io*[33]

The appropriate methods for proof of work need to be established. They can be as basic as a general questionnaire, more complex by asking specific questions that can only be answered after a certain work step has been done, up to screen recording and many more. For this thesis, we opted for an option with file upload and initially decided to follow the official Amazon tutorial using Amazon S3[4], which finally turned out not to be supported by Amazon anymore.
The first option for the file upload tool was naturally Google Drive with the editable link; however, the following turned out to be a problem:

1. everyone needs a Google account to edit and Workers prefer to stay behind their Worker-IDs

2. general control over who edited what- it would be too difficult to track if someone copied, pasted, or altered other users entry

Numerous suspicious online pages and tools with trial versions, restricted number of or upload sizes, non-functional interfaces (advertisement, viruses, etc.) were reviewed until the *ufile* environment was discovered[33].

4.1.3 Methodology selection

For this task, the basic non-iterative or linear competition approach was used.

4.1.4 Crowd selection: Workers profile

Seven students from TU Wien were recruited to participate in the lab version of the study. MTURK grants requester high-level control of who can see or accept their tasks by keeping a set of metrics for each registered worker such as their task approval rate, the number of submitted tasks, their location, etc. For example, it is common to require that the approval rate is above 90%[87]. For the online variant of the study, we recruited 117 MTURK participants(29 of them approved and 88 rejected) from worldwide, HIT approval rate for all submitted HITs greater than or equal to 90%, no age or gender restriction.

- Selection of workers
  There are a number of MTURK specific terms to be clarified before actually starting with study tools preparation. To begin with, the main actors, in the context of MTURK these are requesters and workers, the former being companies, research institutions, or any other individual offering the digital task to the latter, whose

name speaks for itself. The first question from the side of requesters arises at the mere beginning if they don't want to let just anyone solve their HIT, but rather filter the workers. This is where system qualifications come into play. There are three system qualifications available, based on worker's registration and their activity information:

1. Number of HITs approved
   The amunt of certain workers successfully completed tasks since their registration on MTURK. This is the base rate that requesters use to select workers with existing experience.
2. HIT approval rate for all requester's HITs
   This is the rate of accepted and completed HITs. For example, if a worker has completed 1000 HITs, out of which 150 got rejected, the worker's approval rate is 85%.
   Experienced requesters report that the combination of these two qualifications usually delivers results of significantly higher quality results.
3. Location
   This qualification is important if a requester needs to filter by geographical location. It is an attribute a worker adds to their profile, and it can be filtered (both included and excluded) by continent, country, county, or even city.

Additionally, the possibility of setting up two premium qualifications exists. These kinds of qualifications range from age and gender to exercise frequency, and the type of smartphone workers own and allow for inclusion or exclusion thereof. It is important to state that these kinds of qualifications cost extra.
Finally, requester customized sort of qualifications can be created as well.[3]

- Definition of the number of participants
  The authors in[75] conducted the study to show just how many participants are necessary in order to deliver qualitatively good input from the broadest possible point of view on the issue of detecting usability problems. They found that the information gain becomes ever less after reaching the number of 5 participants. Up to that five, as much as approximately 90% of issues could have been covered. That approach could not be applied to this quantitative study because this one is not only about the user interface but rather about solving a specific set of tasks on two different platforms and afterward comparing so that 29 participants are set to be a minimum.

4.2 Validation

There was no automatic validation system, so all the workers' input had to be checked manually. The focus was on the following three parts:

1. number of submitted files

2. content of submitted files
3. content of the open-ended answers
4. minimum time needed to finish the study
5. overtaking placeholder values

4.2.1 Test run

Before publishing the HIT we worked in MTURK developer requester sandbox, which is a simulated environment that lets one test their HITs prior to official publication in the marketplace. This is a free and quite favorable opportunity provided from MTURK itself, allowing all requesters to test, practice, and redesign their surveys, make sure that all the tasks function correctly in advance, which in return saves time at the official run and avoids paying for additional tasks which cannot even be used out of technical or whichever reasons.
It turned out that more than one test run was necessary. An overview of the most important test runs and lessons learned follows:

- Creating a test run version (Dec 8th, Reward per Assignment: $0.80)
  Four basic tasks to see how MTURK workers react to and interact with a robotic programming platform were created. Learning the importance of must fields

  1. Move the robot arm with mouse only or by setting random X,Y and Z coordinates.
  2. Move the robot arm by setting following X,Y and Z coordinates:

     X=100

     Y=200

     Z=300

  3. Close, open and close the gripper again. Leave it closed and note where the change happens.
  4. Set the gripper exactly on the object.

  Figure 4.1: MTURK Test run use cases

  happened right during the first run since most of the workers leave the field empty if not explicitly required to fill it out.

- Problem with the evaluation criteria
  At this point, the problem with the evaluation criteria was observed: the questions were held too general and unspecific, referring to entire HIT, rather than individual tasks:
  \- *On a scale of 1-10, 1 being the least and 10 the most, how understandable was this task's description?*
  \- *Could you easily understand what was asked of you in this task?*
  \- *Did you manage to solve all tasks?*

- *On a scale of 1-10, 1 being the least and 10 the most, how difficult did you find all tasks?*
- *On a scale of 1-10, 1 being the least and 10 the most, how much improvement potential does this UI have?*
- *What is the one(any) thing you would like to change?*

- Next run should be done with more specific evaluation questions in mind. Additionally, a Likert scale of 10 was deemed unnecessary since it takes users a long time to fill it out, and it is not even the most important or reliable measurement.

- Test run with the same use cases (Reward per Assignment: $0.40)
  Reading user feedback on this test run pointed out the necessary UI adaptations so that the unnecessary additional explications and giving up on the HIT with a supposedly valid argument that the program doesn't react can be reduced to a minimum.
  At this point, the radion button "Connect to robot "was eliminated, since most of the users claimed "not being able to connect to robot prevented them from solving the HIT. Connecting to the robot is first of all impossible at this point and was also never asked of workers to do. What, however, was offered is that they provide a valid argument as to why they couldn't solve a particular use case, and it will be accepted. We removed the button entirely.
  Another idea was born during this run, and it was to change the code dashboard so that it shows all the time, without the need to click on it beforehand. That way, we reduced the number of files to upload to half.

- Dec 30 (Reward per Assignment: $3.0)
  For this run the same use cases were used as the ones as in TU Wien course *"Montage II: Advanced manufacturing"[24]* . As expected, these use cases(see *Appendix* Figure A.11-A.14) were too complicated for the MTURK audience, which additionally had the disadvantage of no template solutions shown and the lack of expertise. The student's results will be discussed in another chapter. However, there was one worker who solved all the tasks, including the open-ended question one, asking for further deployment ideas. Although this is obviously statistically insignificant, it proves it is possible; it only takes more filtering and probably a higher reward.
  After this test run, the need to reduce the proof of work amount of data was made based on workers' comments and the time necessary for the response quality control. Earlier, the screenshots of all the final states of the robotic arm plus the automatically generated code in a separate txt file was requested. In this case, this resulted in 16 files per worker, which surely cost more time to prepare than it was necessary for solving the use cases.

- Next test run
  Breaking the use cases down to pick a task, firstly not including the *place* so we can test if up until this point is easily doable for the MTURK workers. Detecting the need to introduce standardized usability measurement method in order to analyze the results easier.
  Started with Post-Study System Usability Questionnaire but although very exact given that it has 16 questions which can be answered with a 7-point Likert Scale (N/A option) and also has three sub-scales evaluating system usefulness, information, and interface quality turned out to be too long for MTURK workers to fill out correctly and honestly. Also, the main focus and time investment of the workers should be the actual solving of the use cases and providing qualitatively acceptable proof for that and not solving a questionnaire.

- The next run was done using a shorter usability evaluation method SUS. There will be an entire chapter dedicated to SUS and its result analysis.
  Concerning the use cases, it turned out that the *pick* task was easy to solve, so it was immediately expanded with *place* task (supposing the same difficulty level) and seamlessly extended with *reuse* task.

4.2.2 Use cases definition

Having low payment, a small amount of time on disposal, low attention span, no moderating possibilities, and general complexity of the topic in mind, the intention is to try and get the most valuable output from as few use cases as possible, both numerically and textually.
However, a gradual increase in use case difficulty is necessary. Also important is the tracking of solvability percentage- if everyone can solve the most complicated task, then the use cases need to be made more difficult otherwise, the results are not within the scale.
Finally, understanding that not all workers will be able to solve all use cases is necessary; however, they do have to provide the reason for not doing so. Rejected HITs are at least as equally important as the approved ones, the useful insights can be made, and the difficulty correspondingly adjusted.
According to the IFR *pick and place* is the most common application scenario for industrial robots and robots. Examples are loading and unloading machines, placing parts in a grid, or picking bulk material from a box with or without a camera support. The goal of the use-cases was to build up from the very basic tasks such as solely moving a robotic arm in a certain direction to get to know the tool and its base functionalities to creating compound tasks such as *pick* and place.
Following six use- cases have been designed in order to evaluate the efficiency and usability of *Assembly*.

HIT instructions
The instructions comprised of a set of step-by-step textual instructions, not supplemented with images, explaining how to navigate the robot, and manipulate objects. This text could be reviewed at any point during the user study.
During task execution, no means of support were offered. If any kind of technical issue with MTURK itself occurs, their team will be responsible for its resolution. If, however, any issue with *Assembly* occurs, this will lead to useful insights about the program itself and will be documented in the qualitative analysis part.
The *Assembly* interface itself provided neither limits nor indications on what workers were supposed to deliver in each HIT. No explicit success or failure messages were implemented in the interface either, which granted the user with total freedom in the question of either certain HIT was completed or not, should she continue to the next one or not. Although the effort to keep the instructions short, concise, and informative was made, since we are (mostly) dealing with the human users, it was an inevitable consequence that not all of them would comply with the instructions.
Following is the exact instructions text that was shown to workers:
Make sure you have a computer that supports the Google Chrome browser, and please use it for this survey.
Before starting to solve each and every use- case, make sure you click on the dashboard icon on the lower right part of the screen to show automatically generated JS code.
Please read each Use-case carefully and remember to take a screenshot accordingly.
Use- cases instructions
The program is drag and drop based.
The units used for the (x,y,z) coordinates are millimeters.
The units used for the rotation angles along the spatial axes (RX, ry, rz) are degrees.
By the workflow, the task line on the top starting with "start" and "stop" actors is meant.
Useful hint: note that when you put your mouse over any component, a small "Help" window will appear.

Use case 1: Robot motion

- Create a program that performs a robot motion between two points.
- The first point should be set to x=100, y=200, z=300.
- Subsequently, drag the "move to" actor to a location between the "Start" and "Stop" actors in the workflow (i.e., robot task or program) to create/memorize the first waypoint.
- You can freely choose the second waypoint, but it has to be different from the first one.
- Then add another "move to" actor to the workflow to create/memorize the second waypoint in the program.

- Repeat the motion a few times by clicking the play button.
- Note the correct order of successfully creating instructions.
- Do not refresh or close the browser tab, as you will need the current state for the next use case.
- *Make a screenshot of the current screen (with the visible code on the dashboard) and upload it.*

Use case 2: Robot motion in a loop

- As you could see in the previous use case, you could repeat the movement manually by repeatedly clicking on the start button.
- Now, you want to automatize this by creating a simple programming loop. For this, you will use the repetition block called "repeat count".
- Note that, when added to a workflow, the "repeat" block has two parts – a head part (a yellow box with the text "repeat count" on it) and a tail part (empty yellow box).
- Put both of previously created "move to" actors between the head and the tail of the "repeat" actor simply by dragging and dropping them at the corresponding place. This actor uses a variable named "count". Open the variables panel (i.e., "xVariables") and set count = 3.
- Make sure to store this value in the program by adding the "Set variables" (i.e., "x") actor to the workflow before dropping the "repeat" block.
- Now click on the play button. Both movements should now be repeated three times.
- *Make a screenshot of the current screen (with the visible code on the dashboard) and upload it.*

Use case 3: Creating a placeholder task

- You are now familiar with the basic robotic arm operations.
- Refresh the browser window to start with a clean task. Please don't skip this step.
- Then simply rename the task from "Untitled" to "mturk" (double click) and save it by dragging the disc icon to the browser's bookmarks bar to save it.
- You will need it in the final task.
- *No screenshot of this screen is necessary.*

Use case 4: Creating a personalized task

- Refresh the browser window to start with a clean task. Please don't skip this step.
- You will now teach a robot to pick up the small blue cube. Set it to the following position: x=300, y=300, z=89 and add the "move to" actor, same as in the 1st use case.
- Now add the "close" actor to the workflow to close the gripper and grab the cube. Then drive the robot to x=300, y=300, z=300, and add another "move to" actor.
- Finally, rename the task from "Untitled" to *pick* and save it by dragging the disc icon to the browser's bookmarks bar.
- Run the task to *pick* the blue cube up.
- You now created your first example of a personalized task.
- *Make a screenshot of the current screen (with the visible code on the dashboard) and upload it.*

Use case 5: Building upon personalized task

- Refresh the browser window to start with a clean task.
- Now, drive the robot to a "pre-place" location at about z=200 above the table, then down to the table level (z=89) and the "open" actor to the workflow to release the gripper.
- Then use the "relative motion" actor(->dP<dR) to drive the robot back up to z=200 (the relative motion actor uses dP, dR parameters).
- So, select the parameters tab and set z=111 so as for the robot to go to z=200.
- After setting this parameter, drag and drop the "set parameters" actor into the workflow before the "relative motion" (->dP<Dr) actor.
- Rename the task to *place* and save it to the bookmarks, just as you did with the previous tasks.
- *Make a screenshot of the current screen (with the visible code on the dashboard) and upload it.*

Use case 6: Building a compound task

- Refresh the browser window to start with a clean task.
- Now load all of the previously saved tasks from the browser's bookmarks, in the following order: *pick*, *place* and "mturk".
- You will notice that they now appear in the "Actor and Task Library" and that you are currently in the "mturk" task. This allows you to load a fresh task without losing the *pick* and *place* tasks from the library.
- Finally, create one last task called *Pick and place*, merely containing the *pick* and *place* tasks as actors and save it to the bookmarks bar.
- *Make a screenshot of the current screen (with the visible code on the dashboard) and upload it.*

4.2.3 Instructions writing

In summary, a brief general explanation was given:
We are conducting an interactive academic experiment about Web-Based Robot Programming Environments.
We need to understand your opinion about usability and overall user experience of specific Web-Based Robot Programming Environments.
The experiment itself does not have very high complexity and is easily solvable within one hour; however, due to the academic background, the precise description makes it seem more complex.
The ideal outcome of this exercise would be you creating a simple "Pick and Place" task, widely known in the robotics community. All steps will lead you to it.
Note that it is not expected of you to solve all of the Use-cases; however, if you do get stuck on a certain one, in order to get paid, you MUST document in the feedback section the number of Use-case and the reason why you couldn't solve it.
Finally, if you recently took part in this same HIT, please don't try to hand it in again, as we are looking for a diverse audience. The explanation was reinforced with concrete examples:

4.2.4 Demographic questionnaire

For better statistical evaluation, the following demographic information questions were asked at the beginning of HIT:

- What is your age?
- What is your gender?
- Which country do you live in?

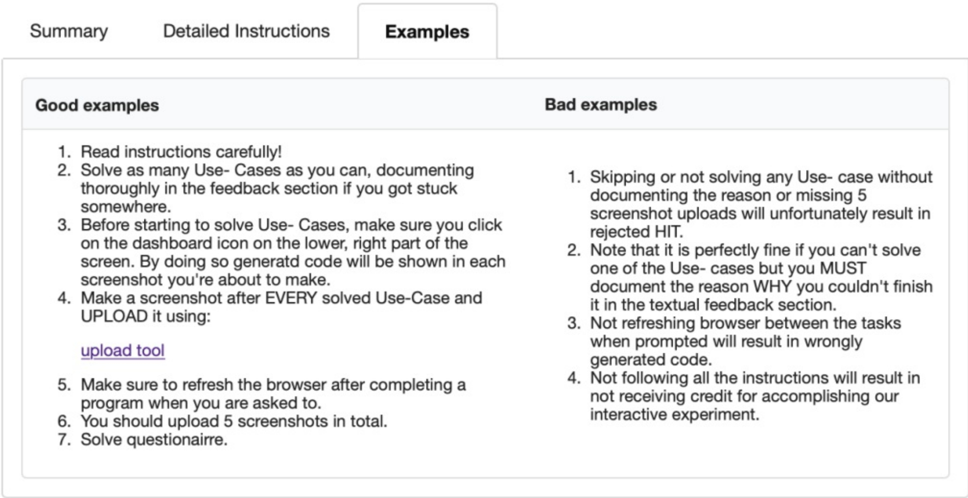


Figure 4.2: MTURK instructions examples

- What is your highest finished level of education?
- Which university did you attend (University name + country)?
- On a scale of 1-5 (1 being the least and 5 the most), how much familiarity level with programming do you have?
- On a scale of 1-5 (1 being the least and 5 the most) how much familiarity level with robotics do you have?

4.2.5 Scam detection

A considerable amount of spammers who deliberately break the MTURK rules to submit any and as many HIT solutions as possible so that they could get their financial reward are present on the platform. They upload diversified submissions, from completely empty files, only required fields filled out or complete random text or sign input. As [87] correctly stated, a part of these workers are spotted automatically by analyzing their submission associated metadata, unquestionably rejecting completely empty HITs. MTURK then declared specific worker a spammer if she is identified with multiple faulty submissions, automatically blocking her and rejecting all of her open submissions as a consequence.
For this thesis, we assumed that all the workers were human, and no software robots were involved. However this is not very probable and is one of the first points to be researched in the future work.
We performed outlier detection based on five measurements:

1. per-participant minimum processing time
2. per-participant maximum processing time

3. the question about University as an adaption of the Instructional manipulation check from[77]. An Ivy League university(Harvard) was set as a placeholder here. The idea was that the worker who either doesn't read the question properly or owns a bot that solves the task for her would simply overtake the placeholder text. On the other hand, what is the probability of someone with a Harvard University degree to be solving MTURK assignments for rather a small amount of re-compensation?

4. arbitrary text in free but compulsory text fields

5. fake or no uploaded files at all

4.2.6 Payment

During test runs we experimented with different price ranges(see section 4.2.1) and finally decided that $5,50 per HIT is the optimal price, being potentially attractive for higher rated workers and at the same time relatively cheap compared to other usability evaluation methods.

4.2.7 Test runs QA

As thoroughly described in section 4.2.1, altogether 5 test runs were carried out, each and every missing some fundamental piece but also delivering lessons learned.

4.3 Execution

Final run March 2021 (Reward per Assignment:$5.50)

- Appropriately detailed instructions and explanations written
- 6 clear use cases
- Defined upload tool
- Usability evaluation method integrated
- UI errors corrected

Even though all the elements of the final HIT were correctly defined, a re-run was necessary. All HITs have a so-called "Task expiry date", which is the maximum task availability time after which it becomes hidden if not re-published by the requester. However, independently thereof, we inferred from the experience on all the tasks that the peak in HIT acceptance is within the first three days. Interestingly enough, it is not immediately on the first day where the most workers get approved, since the results showed that the fastest acceptors are mostly scammers. More on this will be said in another chapter dedicated to fraud detection.

4.3.1 Test surrounding and procedure

Although it is known that laboratory or similar surroundings reduce all possible disturbances to a minimum in order to gain the most trustworthy results possible, this study won't be conducted for one out of two reasons. The first reason is that the main evaluation method will be done by crowdsourcing, which makes it nearly impossible to control the testing environment because of the number and location of the participants. The second and even more relevant one is that the author believes that laboratory conducted studies yield somewhat idealized results, which in this case don't apply since the robotic programming interface will probably be used in factories and non-disturbance-free environments. Although both concepts have their own strengths and weaknesses, due to the above reasons, the participants will be asked to create what feels like a typical working environment for them and try to finish the tasks with as few disruptions as possible. If they, however, end up having unplanned disruptions anyway, these will be analyzed, and conclusions on their impact on the study will be drawn.

4.4 Evaluation

The research method was chosen to be the combination between quantitative research and qualitative research. Quantitative research is the art of testing, which deals with measurements of numerical metrics, such as the amount of time needed to handle certain tasks. Quantitative tests are perfectly convenient for the experimental part of this research since the exact amount of time needed to solve each use case will be one of the important variables measured. MTURK itself will be in charge of possible technical issues. The qualitative research part will handle everything from use case design and documentation, analysis of the concrete test output, and observation of all side effects and events.

- Qualitative analysis
  After the finished HIT the following questions were asked per use case:
  - How understandable was this task's description?
  - Could you easily understand what was asked of you in this task?
  - Did you manage to solve this task? (Yes/ No)
  - How difficult did you find this task? (Likert scale 1-10)
  - If the response on the previous question was between 5-10: What was the most difficult thing about this task?
  - Do you think this UI has improvement potential?
  - What is the one(any) thing you would like to change?
- Quantitative analysis
  Crowdsourced evaluation works hand in hand with quantitative testing because it

automatically measures the performance of the design and tracks usability metrics easily since the data is being recorded automatically[85].
Out of simplicity and efficiency reasons, SUS[44] was chosen for the usability evaluation for this thesis. It consists of the following questions, each of them with with five response options ranging from "Strongly agree" to "Strongly disagree"[44]:

1. I think that I would like to use this system frequently.
2. I found the system unnecessarily complex.
3. I thought the system was easy to use.
4. I think that I would need the support of a technical person to be able to use this system.
5. I found the various functions in this system were well integrated.
6. I thought there was too much inconsistency in this system.
7. I would imagine that most people would learn to use this system very quickly.
8. I found the system very cumbersome to use.
9. I felt very confident using the system.
10. I needed to learn a lot of things before I could get going with this system.

4.5 Results

Final results are visible after the predetermined number of HITs are solved, which in our case was over 25. Each and every one of them was manually verified, and only qualitatively satisfying ones were accepted. In order to analyze the results more exact, they were divided into different sections, considering both accepted and rejected HITs. Since this is quite an extensive section, the entire next chapter is dedicated to it.

CHAPTER 5

Results

For the sake of better understanding and a more straightforward analysis, we divided the findings of the results into multiple sections.
First, we will take a short look at the results of the student study, which we consider our pilot study. Then, overall findings relevant across all the tested groups will be analyzed. After that, our focus will head to the analysis of accepted HIT results, followed by the respective analysis of the rejected ones.

5.1 Student results analysis

Prior to the execution of the MTURK study, a pilot study with the students of the TU Wien was performed as a part of an elective master course called *"Montage II: Advanced manufacturing"[24]*. This course is about presenting current manufacturing concepts specialized in the field of assembly with emphasis on planning, learning factories, and COBOTs. Students were encouraged to learn about the dynamic division of labor at a human-robot workstation, which includes the programming of a robot using intuitive programming concepts. Previous knowledge requirements was a course *"Montage I: Fundamentals of assembly planning and design"[23]* which presumes basic knowledge of operational and corporate management, production and QA.
Knowing the prerequisites the students had to bring and additionally that the study with them was carried out a little later in the semester, they were suitable candidates for the pilot study. The main goals of the pilot study as such are to detect and solve as many potential issues as possible while intending to clarify and polish research questions and use cases and at the same time gathering data on time and cost of the project. However, it is important to point out that not even by the rule of the thumb recommended the number of 12 participants could participate in our study since there were only 7 students inscribed in the course.

Observing figure 5.1, we can see that the students rated *Assembly* relatively high.

SUS Measurement/ StudentID	SUS Frequency	SUS Complexity	SUS Easy	SUS Support	SUS Integration	SUS Inconsistency	SUS Learnability	SUS Cumbersome	SUS Confidence	SUS Learn	SUS Score
Student 1	2	1	5	1	4	2	5	1	3	1	82,5
Student 2	1	3	2	2	2	3	2	5	2	2	35
Student 3	2	4	2	1	1	2	3	5	2	3	37,5
Student 4	5	1	4	2	5	1	4	1	4	1	90
Student 5	4	2	5	1	4	1	5	2	4	1	87,5
Student 6	5	3	4	2	3	3	4	2	5	2	72,5
Student 7	1	4	2	1	2	1	4	3	2	2	50
Average:											**65**

Figure 5.1: Students SUS scores

Understanding how SUS evaluation works, the average score of 65 would mean that the result lies at the beginning of the marginally high acceptability range. A score above 68 stands for the above average, and 65 lies very close to it. It is difficult to say if this is due to the student's relatively high expertise with the matter or if they are partially biased due to the study being a part of their university course, so they graded higher in concern about their course grade. Whatever the case, it definitely helped us to clean up and decide on adequate use-cases based on their solvability and the ideas from students' input.

While observing the figure 5.2 depicting overall SUS tendency, the difference of almost 20% is visible between the results of rejected HIT workers and the pilot study. However what is interesting is that the difference between accepted and rejected HIT workers was comparatively small, amounting to low 3,13%.

This is an interesting point which sets foundation for the further research. It yields two

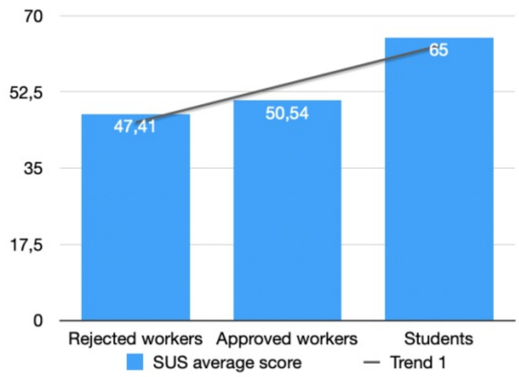


Figure 5.2: SUS scores across all groups

curious questions:

1. Could the SUS evaluation results be considered relevant even if they originate from MTURK rejected worker's input?
2. Is there an observable pattern sample in rejected workers SUS evaluation results which recognition can lead to early scammer recognition(see *Appendix*)?

The first question is out of scope for this thesis. We will, however, try to set the basis for the second question, which could result in a partial answer on the first one.

5.2 Statistical findings across all groups

This chapter will focus on general statistical findings across all MTURK worker groups, which we didn't find essential to observe separately per group. What is more, direct comparison between them is what makes this findings valuable. Note that we don't have such data for the pilot study.

As visible from the above graphic and also what we expected when we decided on

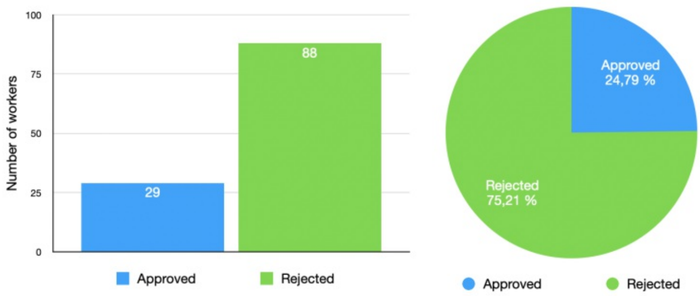


Figure 5.3: Number of approved vs. rejected workers

this evaluation method, the average HIT approval rate lies rather low, at 24.79%. This is something that definitely needs to be considered when opting for crowdsourcing. It indirectly implies the time that needs to be invested in making the according decision on acceptance or rejection of each HIT. Individual analysis of each HIT is unavoidable if the researcher wants to make sure that all the input is valid.

Figure 5.4 focuses on the difficulty level rating comparison between accepted and rejected HITs. The rejected workers rated all of the use cases with a higher number, meaning that they supposedly found them more difficult than the approved workers did. N/A answers were considered in this rating as well. In our opinion, this measure is directly linked with the participants' previous experience with programming in general and, more specifically, with robotics. However, analyzing workers input on those two criteria turned out to be contradictory, since the rejected workers claimed to have considerably more experience in both of those fields, as visible in the following figure 5.5:

To our understanding, this can only be the result of workers dishonesty.

Another measurement that, in comparison, turned out as expected was the average duration. The workers were restricted to a maximum of 2 hours for solution times. However, the data here isn't scaled (there are 59 workers more in the rejected group than in the accepted), the trend is clearly showing that the approved workers on average took more than double the time the rejected workers did, which implies that they dealt with the tasks more profound. When observing individual solving times more in detail, data scattering among rejected workers stands out. In the following figure 5.7, resolution times

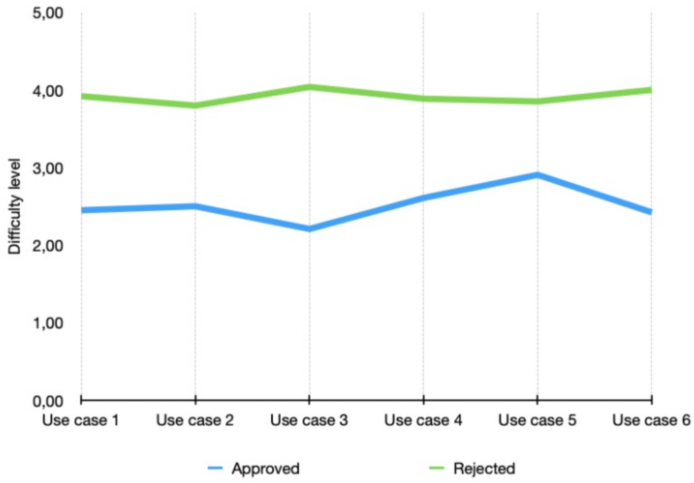


Figure 5.4: Use case difficulty evaluation approved vs. rejected workers

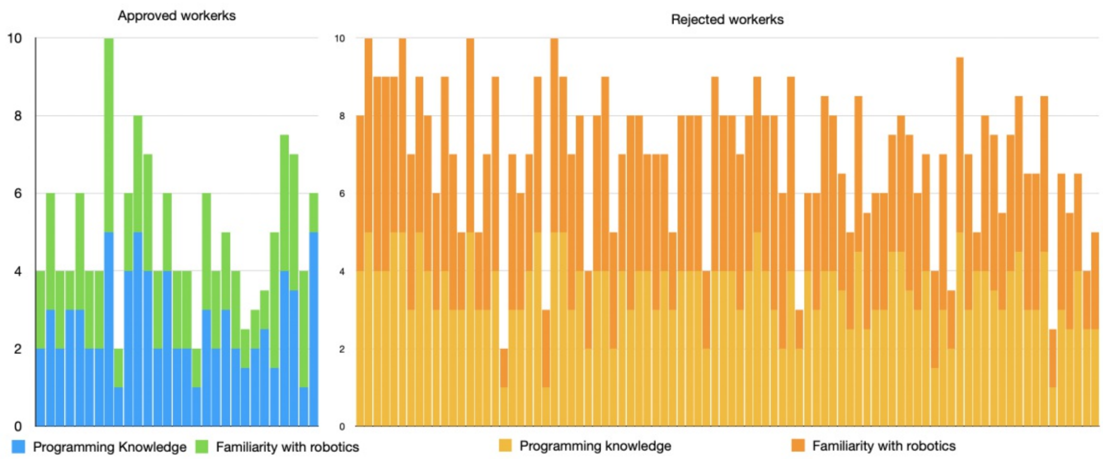


Figure 5.5: Previous knowledge in programming+ familiarity with robotics approved vs. rejected workers

vary from only a couple of minutes up until the full 2 hours. Similar to [92] conclusion, we assume that the disparity in overall task completion times and results is in large part the consequence of some users skimming over the instructions or not reading them at all. We consider it rather inconvenient that the individual task resolution time cannot be deduced from the automatic MTURK output.

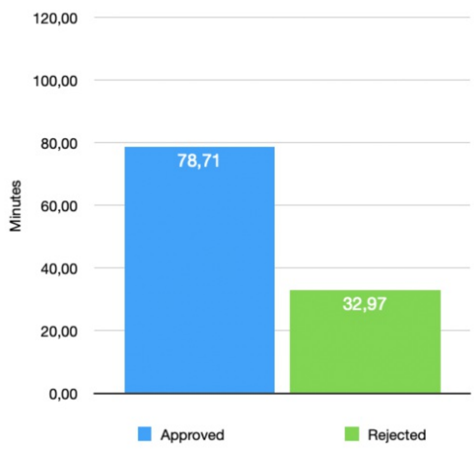


Figure 5.6: Average duration in minutes approved vs. rejected workers

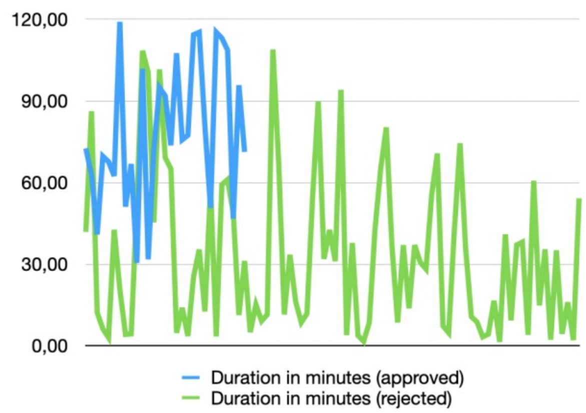


Figure 5.7: Duration in minutes approved vs. rejected workers

5.3 Sentiment analysis in R of workers textual input across all groups

The bar chart in figure 5.8 shows the count of the occurrence of the most common words used by workers. However, it is viable to mention that the R query filtered the following words out: "use", "case" and "task", since they don't tell us much about workers sentiments, but rather point out the commonly used words in the entire HIT. The word "good" being the first one is definitely a favorable fact since this word is the symbol of positivity.
To be able to make the conclusion about the connotation of the following words, additional word association analysis needed to be done in R. This analysis builds upon correlation, a statistical technique that grants and insight into if and how strongly pairs of variables are

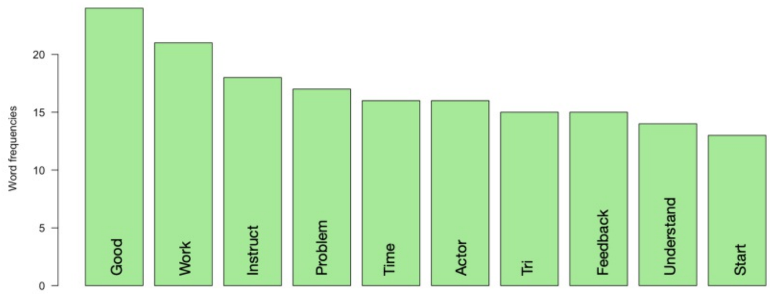


Figure 5.8: Top 10 used words in workers textual input

related. This knowledge might then be used adequately to analyze the context around the most frequently occurring words. This bar plot allows for a fast and straightforward

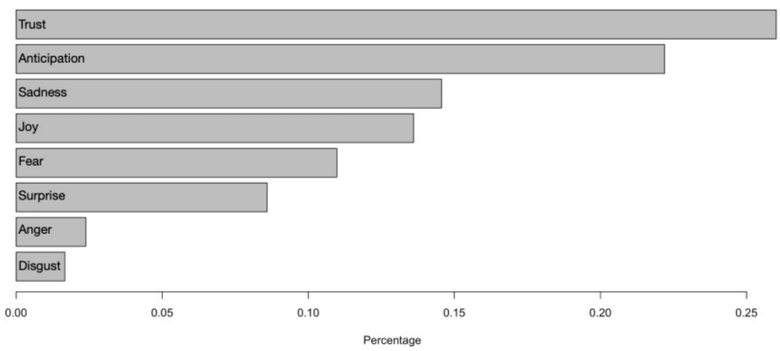


Figure 5.9: Count of words in workers textual input associated with each sentiment expressed as a percentage

correlation of the proportion of words associated with each emotion in the text. As visible from figure 5.9, the emotion "trust" has the longest bar and shows that words associated with this positive emotion constitute over 25% of all the relevant words in all the workers' textual input. On the other end, the emotions of "disgust" and "anger" have the two undoubtedly shortest bars and show that words associated with these negative emotions even together constitute less than 4% of all the meaningful words in workers' textual input. Overall, words associated with the positive emotions account for over 60% of the meaningful words in the text, which can be interpreted as fairly positive feedback from the workers' side.
Entire textual input can be seen in the *Appendix* from the figures A.5-A.10. The exact explanation and division between approved and rejected workers will be discussed in more detail in the upcoming sections.

5.4 Improvement suggestions proposed by workers across all groups

In order to understand and make use of workers' input better, their textual input hasn't only been analyzed with "R" but as well-read and examined manually.
Most of the written feedback has been repeated at least two times, and for the sake of a better overview, we decided to split them into five categories:

1. Don't understand what was meant by:
    - Drive the robot
    - Screenshot: did we ask entire code or just the visible chunk
2. Had hard time figuring out how to:
    - Bookmark a task
    - Copy the code from the clipboard or understand when the code gets generated
    - Set variables and set param
    - Work with the "relative motion" actor dp
    - Understand task 4
    - Understand task 5
3. Not functioning properly:
    - Play button after the first use not executing correctly or at all
    - Browsers other than Google Chrome (Robot disappearing from Safari)
    - Saving tasks
    - Cube on wrong coordinates
    - Glitch between the execution of *pick* and *place* tasks where robot drops the cube while continues its waypoint
    - Individual tasks disappearing when refreshing the page
4. Suggestions:
    - Add the possibility of using the top right input fields in the instructions
    - The bookmarked actors should be shown separately after reloading the task
    - More specific instructions needed, image or video tutorials suggested

- Less cluttered UI
- Robotic-arm should move using the mouse and keyboard
- Make help section more understandable. Apparently, it is very confusing for the newbies
- Zipping the uploads

5. Uncategorised:

- although the simulation basis is interesting, it would be interesting to observe the interaction with the real robot
- More than three workers claimed not to have an idea where to even begin
- Tries to be Scratch: It wasn't clear how to write commands or functions
- complaints about missing the instruction for placing the certain actor in the workflow, which leads us to the conclusion that the workers are barely working independently and require literate instructions in order to finish the HIT
- however when it plays into their favor, workers definitely were independent, stating the following error message as a reason for not being able to solve the entire HIT multiple times:
  "Error: cannot connect to the robot"
  Connecting to the robot was never asked of them, and it definitely wasn't necessary to finish any task.
- Overtaking placeholder value was recognized as one of the most reoccurring problems in open-ended questions
- even though the workers were kindly asked not to repeat the task if they had already done it, 2 of them did it anyway and reported high learning effects- they reported faster solution and a higher rating
- When writing the name of the task (near the disk icon) and using the left/right arrow to move inside the text, the robot visual shifts too. If you hit the "JS" and "Sikuli" button, the javascript "textual" program disappears from the right side
- File upload was cumbersome and probably harder than the entire task and took the longest

5.5 Approved HIT analysis

In this section, we are going to analyze the demographics and responses of approved HIT workers.

5.5.1 Results of the demographic questionnaire of approved workers

As figure 5.10 shows, the following three countries dominate the approved HIT workers nationality:

1. India
2. Brazil
3. USA

Given these countries numerous population, comprising almost 25% of world's entire population[37] this is nothing curious. What is interesting however is that this statistic changed in the context of rejected HIT workers as we will see in the next section. The

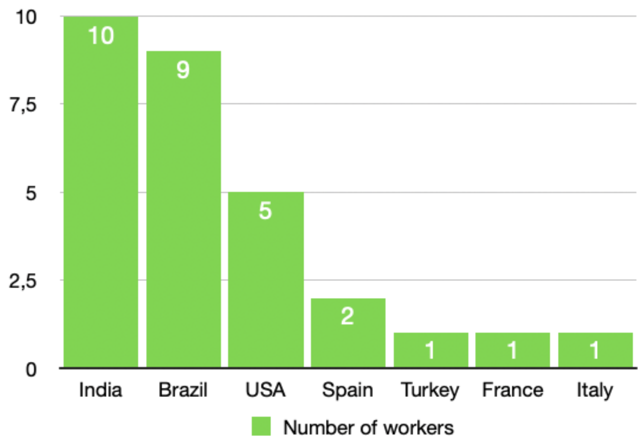


Figure 5.10: Approved workers nationality

next measure we looked into was the workers' age. We intentionally didn't set either low or high limits, as we wanted to be as inclusive with the workers as possible. Our software doesn't have any explicit content, so the standard 18+ limit was unnecessary, and the limit to above is deemed irrelevant out of the same inclusiveness reason. As we can see in the above graphics, the minimum age is 22 and the maximum 45, with 31 as the average age. There is merely one outlier above at 52 years. the
Gender distribution is particularly clear: only 3 out of 29 workers declared themselves as female: On the academical background, we can see that the majority of the workers filed Bachelor studies as the highest academical achievement. However the field "Other" was highly diversified, from elementary school up to literally "Some college".

5.5.2 Sentiment analysis in R of approved workers textual input

Free text workers' input can be divided into two sections, the first being the argument as to why the certain task wasn't solved and the second merely free feedback. Both of the

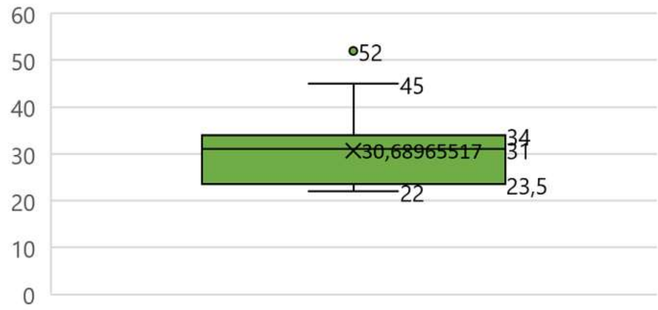


Figure 5.11: Approved workers age

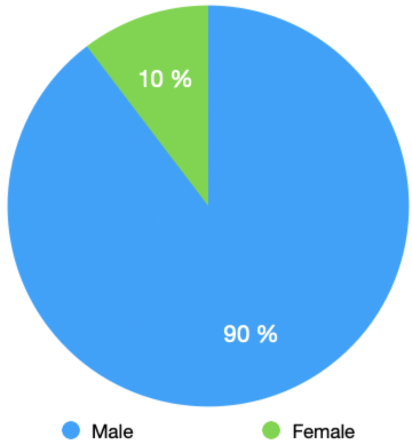


Figure 5.12: Approved workers gender

fields were marked as mandatory, so the workers had to write something. Exact workers' input can be seen in the *Appendix*, in the figures A.5 showing the arguments for not solving certain tasks and figure A.6 respectively being the free feedback.
The bar chart of the workers' textual input shows a relatively even distribution between the ten words that occur the most often. This might mean that the workers mostly spotted the same issues, had the same problems, or liked the same feature. Since all the answers are private, no danger of crowd thinking is possible here, so we interpret this as a good sign, giving us clear and repetitive directions for where to look for corrections and advancements. The word count percentage in accepted workers' textual input associated with each of the eight basic emotions is the most pronounced in the first three, accounting for more than 60% of total emotions. This puts trust, anticipation, but also sadness in the leading emotions read from the accepted workers' input. Trust can be explained by our high requester rating and anticipation obviously because of the suspense in how to solve certain tasks and what comes next. Sadness, on the other hand, could possibly be explained as the outcome of the bugs and program errors finding, which impeded

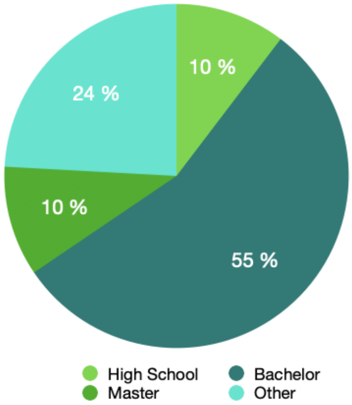


Figure 5.13: Approved workers education

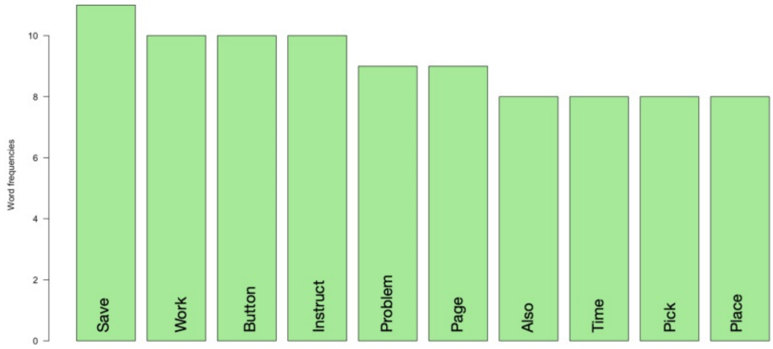


Figure 5.14: Top 10 used words in approved workers textual input

workers from solving one or more tasks. Fear dominates over joy for more than 10%, which is curious and may be reasoned by the general reputation of the robotic industry context, which is rather known as a difficult field. Looking back at the workers' input on familiarity with robotics and/or programming, this makes sense because most of the workers from this category rated their experience with less than 50%. The next emotion is "surprise, "which accounts for a little over 0,05% but can be argued either by workers' surprise on the asked questions or with their own ability to deliver answers. Finally, "anger "and "disgust "are placed in the end, both with less than 0,05% value, which is obviously a positive sign.

5.5.3 SUS score of accepted tasks

As a usability evaluation scale we used is commonly known SUS scale defined by [44]. Note that out of the simplicity reasons the option of N/A was offered. According to the author [44], all items should be checked, however if certain respondent feels that they cannot respond to a particular item, they should mark the center point of the scale. This is why we subsequently changed all the N/As to 3. Our average result of the

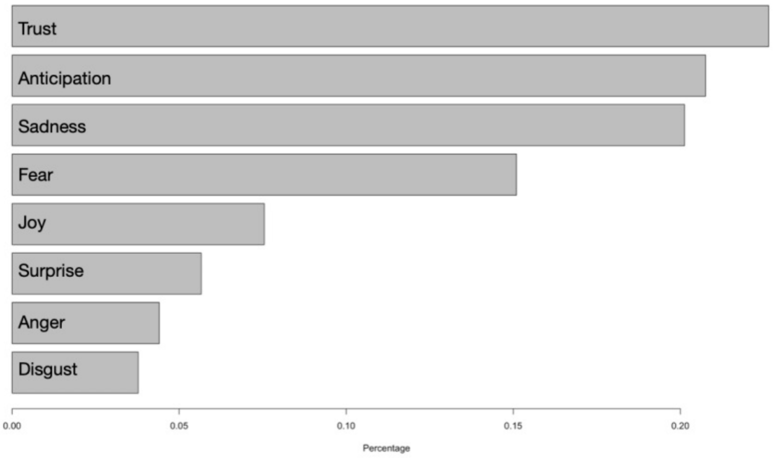


Figure 5.15: Count of words in approved workers textual input associated with each sentiment expressed as a percentage

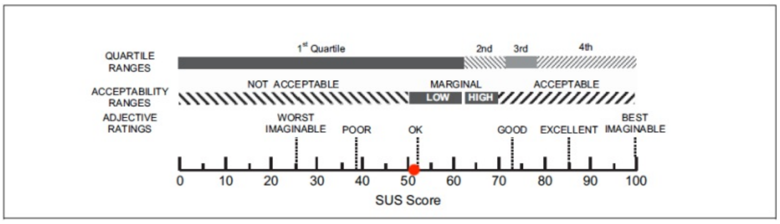


Figure 5.16: Our result(5̃0,54) positioned in grade rankings of SUS score with an adjective rating scale based on [40]

SUS evaluation leads to the conclusion that the UI has a lot of improvement potential. *Assembly* was developed as the university program, and this is still its alpha version, so this was to be expected. Specification of improvement potentials is out of the scope of this thesis. Individual ratings can be seen in the *Appendix* from figure A.1.

5.6 Rejected HIT analysis

This section will be dedicated to the analysis of the demographics and responses of rejected HIT workers. The similarities and differences between the same data from the approved HIT workers are interesting to observe. One of our incorrect assumptions was that there wouldn't be any relevant differences between the two.

5.6.1 Results of the demographic questionnaire of rejected workers

Starting with the nationality analysis, in the case of rejected workers, there is a clear forerunner, which is the USA. This country was only at the 3rd place in the approved workers' statistics. The next one is India, which beforehand was at first place and now

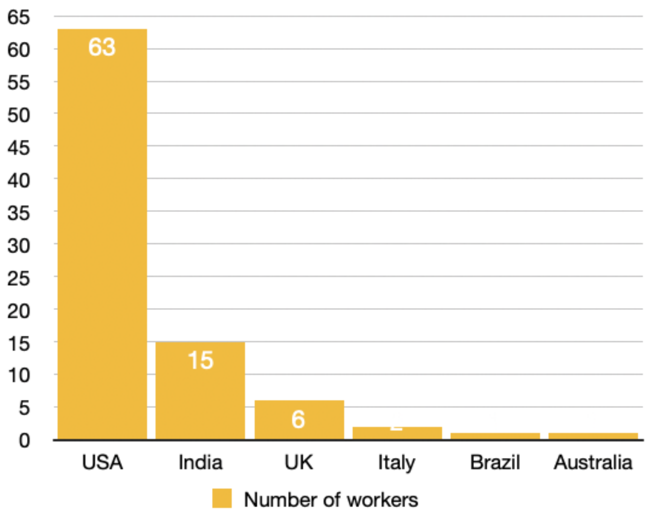


Figure 5.17: Rejected workers nationality

follows as the second but with 76% fewer workers. The 3rd place goes to the UK, which in the case of approved workers didn't even appear.
Now observing the workers' age, we notice higher data scattering. Here the minimal age is set at 18 and maximal at 46, with the average value of 35. The majority workers are between 29 and 37, with six outliers above the maximum, at 50, 54, 57, 59 and 63 years. Gender distribution is predominantly male here as well, however less definite than

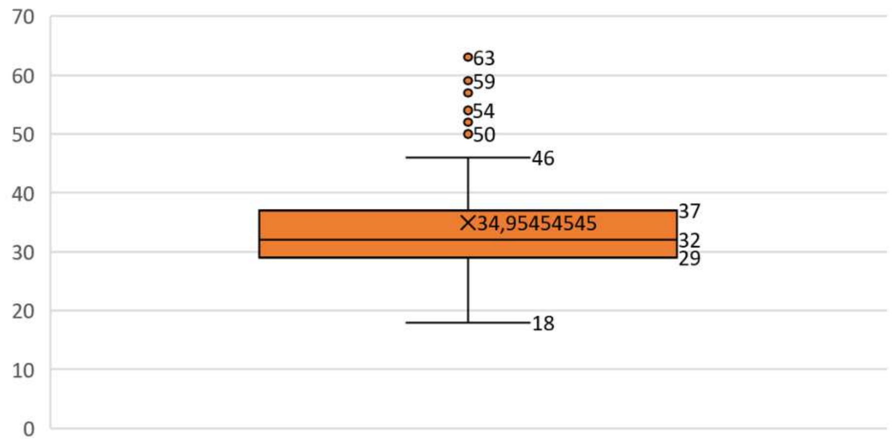


Figure 5.18: Rejected workers age

in the approved workers. One worker didn't want to declare their gender, which is an important point that we will take away and include in the future HITs. Educational statistics is particularly interesting since it seems illogical at first glance. Here we notice

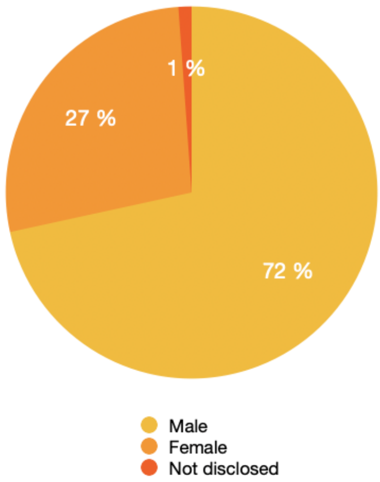


Figure 5.19: Rejected workers gender

the predominance of Master's study input, which can be attributed to two things. Since we are analyzing fake workers' input here, we have to understand that they probably intended to fake all of their inputs, and in this context, they opted for the highest possible degree of education. Another thing is that we used "Harvard" as the placeholder for the university. This turned out to be an almost infallible detector of fakes. None of the approved workers had Harvard as their university. Also, it doesn't make much sense that the students or Alumni of the leading Ivy League institution need extra income as low and as basic as the one from MTURK.

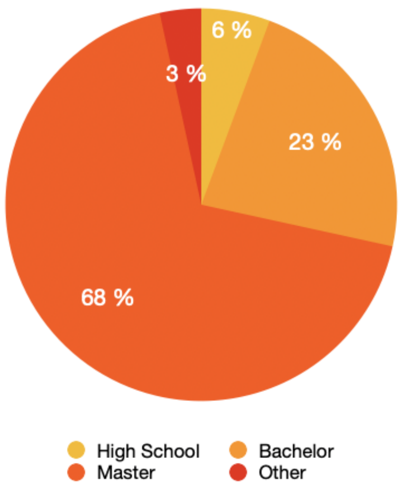


Figure 5.20: Rejected workers education

5.6.2 Sentiment analysis in R of rejected workers textual input

Textual input of rejected workers' groups can also be divided into two groups, just as for the approved workers. Exact workers' input can be seen in the *Appendix*, in the figures A.7 and A.8 showing the arguments for not solving certain tasks and figure A.9 and A.10 respectively being the free feedback.
The only word coinciding in both groups is "work", which doesn't bring us much insight itself. However, the lack of common words helps us understand that the two groups focused on different things. It is interesting to notice that the rejected group used many words from the placeholders themselves, such as "system ", "feedback ", "example ", "nothing ". As mentioned earlier, this was one of the easiest arguments to separate scammers from workers delivering relevant input.
The percentage of the count of words in rejected workers' textual input associated

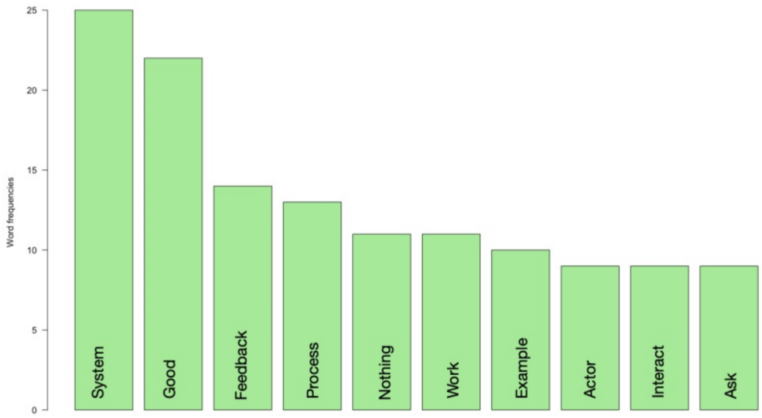


Figure 5.21: Top 10 used words in rejected workers textual input

with each of the eight basic emotions coincides in four out of eight with the accepted workers' evaluation. However, it is more pronounced in the first two, for as much as 5%. This means that trust and anticipation are prevailing emotions read from the rejected workers' input. Trust can be explained like previously, by our high requester rating and anticipation obviously because of the suspense in how to solve certain tasks and what comes next. Joy dominates over sadness for more than 5% while both surprise and fear amount to approximately 10%, both of which can be explained as the "anticipation ". "Anger "and "disgust "are placed in the end, both with 1% or less, which is obviously a good sign.

5.6.3 SUS score of rejected tasks

The average result of the SUS evaluation of rejected tasks leads to the same conclusion as the one for the accepted tasks, and that is that the UI has a lot improvement potential. We notice 3,13% lower rating on this group compared to the approved one. Specification

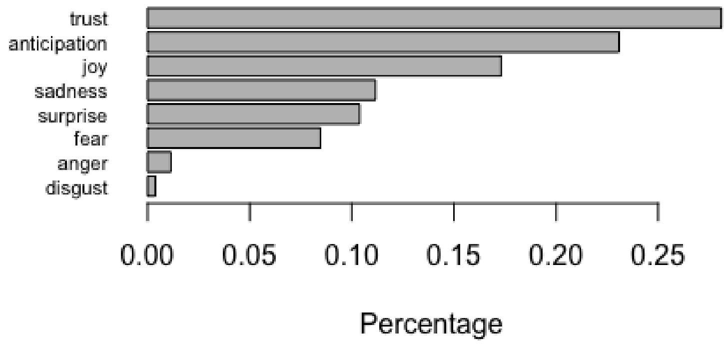


Figure 5.22: Count of words in rejected workers textual input associated with each sentiment expressed as a percentage

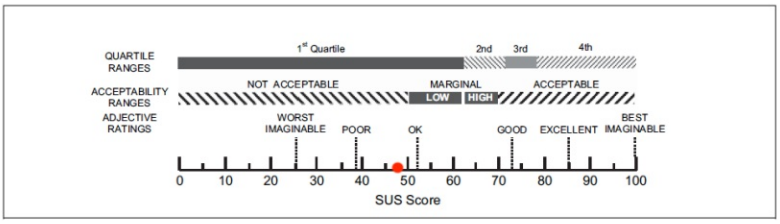


Figure 5.23: Our result(4̃7,41) positioned in grade rankings of SUS score with an adjective rating scale based on [40]

of improvement potentials is out of the scope of this thesis. Individual ratings can be seen in the *Appendix* from the figures A.2-A.4.

CHAPTER 6

Discussion and outlook

In this final chapter, we will dig into the findings of this thesis. We will begin with a summary of the results and their meaning. Next, we will analyze every research question and validate or reject the hypothesis we made in the beginning. Then we will shift the focus to the relevance of the findings through our discussion but also placing the thesis within the context of other relevant studies from the literature review. Of course, the limitations part should not be missing from this chapter, so we will dedicate one section to acknowledging the shortcomings of the thesis. The final section will be dedicated to setting up the foundation for future research in this and similar fields.

6.1 Interpretations

We will begin with going over the research questions and analyze the answers this thesis offers.

1. The first research question dealt with the feasibility and reusability of crowdsourcing approach in the context of evaluation of web-based robotic programming interfaces. Starting with the first criteria, we can say that this approach definitely proved to be feasible due to the diverse and valuable input we received from the MTURK workers for a rather small input of resources. Overall we spent 223,23€ for both the test and final run. For this amount, we gained 117 valuable insights from the final run, both from 29 accepted workers and 88 rejected ones, and also numerous input from many test runs before. It is important to emphasize that it was not necessary to pay for the input of rejected workers. However, this averages to 7,69€ for each accepted HIT, which is finally not even close to the low rates promised when using crowdsourcing tools[56]. Our analysis supports the theory that this can be attributed to the many tries, fails, and lessons learned from the initial test runs. One of the main contributions of this thesis was creating the framework for implementation of such

study, which, if followed, should definitely save a considerable amount of time and money, among other things.
Next, the crowdsourcing method's reusability is definitely high out for many reasons. The first reason is that the tools such as MTURK mostly work over cloud storage, allowing access to created tasks anytime from anywhere. Next, in the case of MTURK, the tool itself offers editable and extendable templates, which means that the end-user won't need to start from scratch even if she is creating her first task. This is especially practical for the creation and adaptation of the test cases since those almost always need improvement. The option of creating a personalized template is also available.

2. Second question was about the extent of support of crowdsourcing usability evaluation of a web-based robotic programming interface. We can surely say that most of the coarse usability issues will be detected by this method. However, this is questionable for very specific issues or details. One solution could be defining the target audience very specifically, which would cost more money for each criterion apart from the standard ones[3], but would eventually offer a fix. We did not test this since our goal was a diversified audience and broad acceptance.
More on this will be said in the limitations section.

3. Next question was on MTURK audience diversity both from a demographic and technical expertise point of view. The results indicate that this research question definitely can be positively answered, as demonstrated in the previous "Results" section.
To sum it up, we gathered data from 5 continents, altogether nine countries, age ranges from 18 up to 64, mixed genders with one non disclosed, however predominantly male, very assorted academical degrees and according to the self-assessment also quite diversified previous knowledge in programming and familiarity with robotics in general (look figure 5.5).

4. Final question dealt with strengths and weaknesses of our web-based robotic programming interface *Assembly*. In general, we received a lot of feedback; however, it took a manual effort to distinguish between useful and spam input.
As reported in the "Results" section, SUS results were rather low, with 24,79% of workers managing to solve the tasks within the given time frame of two hours, more specifically in less than 80 minutes. All participants from the Student group managed to solve all the tasks. Apart from the SUS score, the most valuable input for this section was the mandatory free text but only in the online group. Within these fields, a number of useful commentaries were made. We split them into five groups according to the thematic they were handling. In the first group, the understandability of the task instructions was handled where workers mostly had the problem of understanding the concept of driving the robot.
The second group dealt with the difficulty of HIT execution, where the top candidates were inability to bookmark the tasks, finding automatically generated code,

and working with the relative motion actors.
The third category dealt with problems in the program, errors, and bugs found by HIT workers while trying to perform the task. By far, the most repeated issue was on task saving and reusing, functioning incorrectly for most workers and also us. This is one of the first things that needs to get looked into and fixed. The second one is the failure of the "Play button " after one successful execution.
In the next section, we gave participants the possibility to suggest improvements. The main takeaways here were tutorials, where many workers wished for or even needed image or video input. In this final run, we opted for the used textual input only with MTURK and image tutorials with the students. This was partially on purpose, to avoid the difficulty with the image tutorials where it could easily happen that workers basically "copy " the image, without thinking on how to use the program and solve the incorrect task order, and partially because it was quite difficult to include images where we wanted in the MTURK templates.
In the final section, we gathered all random input, which was difficult to sort into one of the previous categories. The most repeated one here was the one on the connection error to the robot, which wasn't even asked of the workers in any task, yet many of them tried to do it, couldn't because there was no actual physical robot to connect to and finally they took this as a reason to abort the task, since they thought to have found the error in the program, impeding them from continuation. This was very interesting to observe since it is opposed to the second most repeated input in this category, which was on necessity for more detailed instructions. In the latter input, the workers are basically asked to be guided by the hand because of the complexity of the task when it convenes them. In the former input, however, they found an error nobody asked about, and since it played for them this time, they used it to get out of solving the task and getting paid anyway.

The above data contributes a clearer understanding of this thesis contribution. The results should be taken into account when considering how to improve *Assembly*[63] and be able to respond to the first [8] challenge about creating a necessarily simplified, user-friendly front end for the common end-user, detaching it from the complex back end.

6.2 Implications

The perks of crowdsourcing have been well researched and include factors such as less complicated access to a broad and highly diverse participants pool, considerably lower cost of executing experiments on any level, both things leading to overall faster iteration between having the idea, developing test cases and yielding results from executing final run experiments[72].
What hasn't been studied as thoroughly, however, is the application of this method in the context of UI evaluation, more specifically in the robotic programming interface context. As we thoroughly examined in the literature review section and later confirmed in the

results section, our findings mostly agree with previous research. No paper that we are aware of stated that crowdsourcing was an inappropriate method for gathering necessary input on evaluation of UIs; however, they did almost exclusively use this method to design in the first place[61], rather than evaluate already existing designs.
Building on existing evidence, this thesis adds to the rest of the research by providing a strong crowdsourced evaluation methodology framework based on [98] and specially adapted to robotic programming context.
This thesis provides new insight into the relationship between crowdsourcing and UI evaluation of web-based programming interfaces, in our case, in the context of robotics. Furthermore, it doesn't only provide the framework but also proposes a process of investigation, creation and finally documentation of the use cases and assembly assistance tools for human-robot collaboration in single-user assembly, bearing in mind solvability by people with no prior programming nor robotic experience. Finally, this inevitably leads to exploring new teaching and learning methods in the field of assembly activities as well as new human-robot collaboration practices.

6.3 Limitations of the approach and results

Quality control was one of the most time-consuming and difficult things to do while handling results, as recognized as well by [47]. Due to the online nature, easy reachability of wide audience and low monetary award, the abundance of scammers, and generally low quality of most of the answers are inevitable. They can be divided into various categories:

- the simplest one is data entry in-compliant with the instructions
- the next thing that occurred often was solving task with the identical input, but from different accounts
- software robots
- random auto-fill content, mostly completely out of the context

There is no pre-built QA in MTURK or in any other crowdsourcing tool to our knowledge, which implies that these checks have to be done manually. However there are separate solutions on the market providing commercial services of QA on a large scale tasks, such as:

- CrowdFlower[9]
- HitBuilder[19]

Currently, robotics is still too specific a domain to have universal commercial crowdsourcing solutions[87].
Furthermore, as recognized by [52], lack of QA in combination with a non-simulated

environment means that testing cannot be automated, which means that heuristic score functions for predicting crowd success need to be used. The perfect solution for this would be exact physics models for all the objects used in the task[52], but this is obviously out of scope for this thesis.
Although we set a fixed budget for crowd tasks in a batch mode, the question remains if an incremental approach would be more efficient. This could be solved by picking the next task based on all input provided by the crowd so far. The robot could then skip queries for previously resolved instances or even automatically terminate HIT once a certain performance threshold has been reached[52].
Bearing in mind workers' rather low educational and predominantly basic English language background due to geographical location and the nature of financial motivation in MTURK, it is important that use case instructions are written using basic vocabulary and unambiguous meaning. One way to guard against this is to introduce a language test before getting to HIT itself, of course, if this is relevant for the specific task. The same goes for the programming and robotic skills in our case. We didn't find English proficiency as crucial so as to introduce it as an elimination criterion, and neither programming and robotic knowledge; however, we did include the latter two in a questionnaire so that we can apply respective relevance to their answers.
Another important point, as recognized by [87] is the question of data privacy in crowdsourcing. Being an online service with a low connection between the worker and requester, presenting most of the HITs to any public yet registered observer, the question of data security arises[87]. In our context, this didn't pose a problem or risk, but it is an important factor for HITs with sensible data.

6.4 Recommendations for future research

The generalizability of the results is limited by the elected platform and invested funds. The most important recognized limitations of the MTURK platform itself are the following:

- no possibility to retrace individual MTURK tasks, which means that the results can't tell us the exact way workers came to a solution
- in spite of a high number of templates in MTURK, the options to customize them are not many
- difficult to incorporate proof of work within MTURK platform
- robot software solution detection
- automatic scam detection
- automatic hit rejection

Further research should definitely focus on the last three points, developing a system on how to recognize numerous scammers and ideally initiate automatic rejection in very clear cases.
Concerning *Assembly* itself, the next steps will be the implementation of the recognized issues and subsequently new evaluation runs. For those runs, more complex use cases will be necessary.
One of the first ideas of this thesis was benchmarking *Assembly* against the market leader, but this turned out to be beyond the scope for various reasons. The first reason is that currently, there is no established market leader in the context of web-based robotic programming environments but rather many competitors. Furthermore, even if we chose one of the competitors, we would've confronted more hurdles. To begin with, the platform itself is not enough; it is necessary to incorporate a robot simulator in it. This was the problem with Scratch[31]. The next relevant option was drag&bot[14], a graphical user interface for various industrial robot hardware, plug&play based. We were in direct contact with them and got all the use cases checked and approved as relevant for their side as well; however, the account registering issue decided that we won't cooperate in the end. It was basically impossible to use the drag&bot without creating an account, which is completely understandable, but it was too much to ask from MTURK crowd.
To conclude, this thesis introduces the idea of using crowdsourcing in the context of web-based robotic programming environments, recognizes its shortcomings, and, while acknowledging them, delivers a strong methodology framework implemented upon ISO standard 9241- 210. Moreover, this framework is then proven on the example of the evaluation of *Assembly*[63], delivering important insights on the robotic software itself, but also the crowdsourcing platform MTURK[3].

List of Figures

Acronyms

AI Artificial intelligence. 1

AWS Amazon Web Services. 8

CDEC Crowdsourced design evaluation criteria. 20

COBOT Collaborative robot. 4, 14, 15, 26, 49

COMEMAK Cobot Meets Makerspace. ix, xi, 3, 4, 26

CPPS Cyber physical production systems. 1

CUI Character user interface. 3

DOF Degrees of freedom. 7

FFG Austrian research promotion agency. 3, 26

GUI Graphical user interface. 3, 16–19

GWAP Games with a purpose. 22, 23

HCI Human computer interaction. 10, 18

HIT Human intelligence task. xiv, 8, 9, 36–41, 44–51, 53, 56, 57, 59–61, 63, 65–67, 69

HITL Human-in-the-Loop. 14, 15

HMI Human machine interaction. 4

HRI Human robot interaction. 2–4, 21, 23, 24

ICS Information control systems. 18

IFR International federation of robotics. 40

IMC Instructional manipulation check. 46

IOT Internet of things. 1, 2

ISO International organization for standardization. 25, 26, 70, 71

JSON JavaScript object notation. 6, 18

LFD Learning from demonstration. 16

MTURK Mechanical Turk. xiii, 3, 8–11, 21, 22, 27, 30, 35–42, 44–52, 62, 65–71

N/A No answer. 40, 51, 59

NUI Natural user interface. 3

PBD Programming by demonstration. 16

PSSUQ Post-Study System Usability Questionnaire. 32, 40

QA Quality assessment. 21, 30, 46, 49, 68

RMS Robot management system. 23, 24

ROI Return on investment. 2

ROS Robot operating system. 18, 19

SAR Spatial augmented reality. 13, 14

SME Small and medium-sized enterprise. 14

SUS System Usability Scale. 32, 40, 48, 50, 59, 60, 63, 64, 66, 71, 72, 86–89

TCP Tool center point. 7

UCD User centered design. 25, 26, 71

UI User interface. 3, 4, 10, 11, 17, 20, 21, 32, 35, 39, 46, 56, 60, 63, 67, 68

UX User experience. 19, 25, 26, 31, 32

VPL Visual programming language. 18

Bibliography

[1] 99designs. https://99designs.at. Accessed: 2021-06-01.

[2] Amazon alexa. https://developer.amazon.com/en-US/alexa. Accessed: 2021-05-01.

[3] Amazon mechanical turk. https://blog.mturk.com. Accessed: 2021-06-30.

[4] Amazon mturk s3 instructions. https://blog.mturk.com/tutorial-how-to-create-hits-that-ask-workers-to-upload-files-using-amazon-c Accessed: 2021-05-01.

[5] Apple siri. https://www.apple.com/siri/. Accessed: 2021-05-01.

[6] Blackboard design pattern. https://social.technet.microsoft.com/wiki/contents/articles/13215.blackboard-design-pattern.aspx. Accessed: 2021-06-01.

[7] Blockly. https://developers.google.com/blockly. Accessed: 2021-06-01.

[8] comemak. https://www.comemak.at. Accessed: 2021-05-01.

[9] Crowdflower. https://visit.figure-eight.com/People-Powered-Data-Enrichment_T. Accessed: 2021-07-01.

[10] crowdsourcing glossary. https://www.clickworker.com/crowdsourcing-glossary/power-of-the-crowd/. Accessed: 2021-08-15.

[11] Crowdspring. https://www.crowdspring.com. Accessed: 2021-06-01.

[12] Design hill. https://www.designhill.com. Accessed: 2021-06-01.

[13] Designcrowd. https://www.designcrowd.com. Accessed: 2021-06-01.

[14] dragandbot. https://www.dragandbot.com. Accessed: 2021-07-01.

[15] Franka emika. https://www.franka.de/de/. Accessed: 2021-06-01.

[16] Gartner dictionary. https://www.gartner.com/en/information-technology/glossary/smart-factory. Accessed: 2021-06-30.

[17] Google assistant. https://assistant.google.com. Accessed: 2021-05-01.

[18] Guerra creativa. https://www.guerra-creativa.com/en/. Accessed: 2021-06-01.

[19] Hitbuilder. https://ga-dev-tools.appspot.com/hit-builder/. Accessed: 2021-07-01.

[20] Kuka. https://www.kuka.com/. Accessed: 2021-06-01.

[21] Matlab. https://uk.mathworks.com/products/matlab.html. Accessed: 2021-05-30.

[22] microworkers. https://www.microworkers.com. Accessed: 2021-06-01.

[23] Montage i: Fundamentals of assembly planning and design. https://tiss.tuwien.ac.at/course/courseDetails.xhtml?dswid=3055dsrid=657courseNr=330284. Accessed: 2021-08-05.

[24] Montage ii: Advanced manufacturing (tu wien). https://tiss.tuwien.ac.at/course/courseDetails.xhtml?courseNr=330288semester=2021Wdswid=5005dsrid=624. Accessed: 2021-08-05.

[25] Nielsen norman group. https://www.nngroup.com. Accessed: 2021-06-30.

[26] Pssuq. https://uiuxtrend.com/pssuq-post-study-system-usability-questionn Accessed: 2021-06-30.

[27] Quantitative user-research methodologies: An overview. https://www.nngroup.com/articles/quantitative-user-research-methods/. Accessed: 2021-06-30.

[28] Quantitative vs. qualitative usability testing. https://www.nngroup.com/articles/quant-vs-qual/. Accessed: 2021-06-30.

[29] Salesforce. https://www.salesforce.com/blog/2018/12/what-is-the-fourth-industrial-revolution-4IR.html. Accessed: 2020-06-30.

[30] Savioke. https://www.savioke.com. Accessed: 2021-05-30.

[31] Scratch. https://scratch.mit.edu. Accessed: 2021-06-01.

[32] Statistics on color blindness. https://iristech.co/statistics/. Accessed: 2021-08-05.

[33] ufile. https://ufile.io. Accessed: 2021-06-01.

[34] Uml. https://www.uml-diagrams.org/use-case-diagrams.html. Accessed: 2021-05-04.

[35] Universal robots. https://www.universal-robots.com/. Accessed: 2021-04-30.

[36] Usability is a key element of user experience. https://eu.landisgyr.com/better-tech/usability-is-a-key-element-of-user-experience. Accessed: 2020-06-30.

[37] World population. https://www.worldometers.info/world-population/population-by-country/. Accessed: 2021-06-01.

[38] Ajaykumar, G., and Huang, C.-M. User needs and design opportunities in end-user robot programming. In *Companion of the 2020 ACM/IEEE International Conference on Human-Robot Interaction* (2020), pp. 93–95.

[39] Albert, W., and Tullis, T. *Measuring the user experience: collecting, analyzing, and presenting usability metrics.* Newnes, 2013.

[40] Bangor, A., Kortum, P., and Miller, J. Determining what individual sus scores mean: Adding an adjective rating scale. *Journal of usability studies 4*, 3 (2009), 114–123.

[41] Bauer, W., Bender, M., Braun, M., Rally, P., and Scholtz, O. Leichtbauroboter in der manuellen montage – einfach einfach anfangen. erste erfahrungen von anwenderunternehmen.

[42] Beck, M. Glumb. http://robot.glumb.de/. Accessed: 2021-04-30.

[43] Biggs, G., and Macdonald, B. A survey of robot programming systems. In *in Proceedings of the Australasian Conference on Robotics and Automation, CSIRO* (2003), p. 27.

[44] Brooke, J. Sus: a "quick and dirty'usability. *Usability evaluation in industry 189* (1996).

[45] Chernova, S., DePalma, N., Morant, E., and Breazeal, C. Crowdsourcing human-robot interaction: Application from virtual to physical worlds. In *2011 RO-MAN* (2011), IEEE, pp. 21–26.

[46] Choi, S., Eakins, W., Rossano, G., and Fuhlbrigge, T. Lead-through robot teaching. In *2013 IEEE Conference on Technologies for Practical Robot Applications (TePRA)* (2013), IEEE, pp. 1–4.

[47] Chung, M. J.-Y., Forbes, M., Cakmak, M., and Rao, R. P. Accelerating imitation learning through crowdsourcing. In *2014 IEEE International Conference on Robotics and Automation (ICRA)* (2014), IEEE, pp. 4777–4784.

[48] Crick, C., Osentoski, S., Jay, G., and Jenkins, O. C. Human and robot perception in large-scale learning from demonstration. In *Proceedings of the 6th international conference on Human-robot interaction* (2011), pp. 339–346.

[49] de Jong, J. Json editor online. `https://jsoneditoronline.org/`. Accessed: 2021-06-30.

[50] Djuric, A. M., Urbanic, R., and Rickli, J. A framework for collaborative robot (cobot) integration in advanced manufacturing systems. *SAE International Journal of Materials and Manufacturing 9*, 2 (2016), 457–464.

[51] Emeli, V. Robot learning through social media crowdsourcing. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems* (2012), IEEE, pp. 2332–2337.

[52] Forbes, M., Chung, M., Cakmak, M., and Rao, R. Robot programming by demonstration with crowdsourced action fixes. In *Proceedings of the AAAI Conference on Human Computation and Crowdsourcing* (2014), vol. 2.

[53] Gajos, K. Z., Czerwinski, M., Tan, D. S., and Weld, D. S. Exploring the design space for adaptive graphical user interfaces. In *Proceedings of the working conference on Advanced visual interfaces* (2006), pp. 201–208.

[54] Ghazal, M., Haneefa, F., Ali, S., Al Khalil, Y., and Sweleh, A. A framework for teaching robotic control using a novel visual programming language. In *2016 IEEE 59th International Midwest Symposium on Circuits and Systems (MWSCAS)* (2016), IEEE, pp. 1–4.

[55] Grossman, T., and Balakrishnan, R. The bubble cursor: enhancing target acquisition by dynamic resizing of the cursor's activation area. In *Proceedings of the SIGCHI conference on Human factors in computing systems* (2005), pp. 281–290.

[56] Hara, K., Adams, A., Milland, K., Savage, S., Callison-Burch, C., and Bigham, J. A data-driven analysis of workers' earnings on amazon mechanical turk.

[57] Hara, K., Adams, A., Milland, K., Savage, S., Callison-Burch, C., and Bigham, J. P. A data-driven analysis of workers' earnings on amazon mechanical turk. In *Proceedings of the 2018 CHI conference on human factors in computing systems* (2018), pp. 1–14.

[58] Hertzum, M., Borlund, P., and Kristoffersen, K. B. What do thinking-aloud participants say? a comparison of moderated and unmoderated usability

sessions. *International Journal of Human-Computer Interaction 31*, 9 (2015), 557–570.

[59] Hold-Geoffroy, Y., Gardner, M.-A., Gagné, C., Latulippe, M., and Giguere, P. ros4mat: A matlab programming interface for remote operations of ros-based robotic devices in an educational context. In *2013 International Conference on Computer and Robot Vision* (2013), IEEE, pp. 242–248.

[60] Howe, J. The rise of crowdsourcing. *Wired magazine 14*, 6 (2006), 1–4.

[61] Huang, J., Lau, T., and Cakmak, M. Design and evaluation of a rapid programming system for service robots. In *2016 11th ACM/IEEE International Conference on Human-Robot Interaction (HRI)* (2016), IEEE, pp. 295–302.

[62] Huber, A., and Weiss, A. Developing human-robot interaction for an industry 4.0 robot: How industry workers helped to improve remote-hri to physical-hri. pp. 137–138.

[63] Ionescu, B. T. Assembly. `http://assembly.comemak.at`, doi = 10.5281/zenodo.3839314. Accessed: 2021-04-30.

[64] Jha, S. S., and Nair, S. B. A logic programming interface for multiple robots. In *2012 3rd National Conference on Emerging Trends and Applications in Computer Science* (2012), IEEE, pp. 152–156.

[65] Joo, H. A study on understanding of ui and ux, and understanding of design according to user interface change. *International Journal of Applied Engineering Research 12*, 20 (2017), 9931–9935.

[66] Komarov, S., Reinecke, K., and Gajos, K. Z. Crowdsourcing performance evaluations of user interfaces. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (New York, NY, USA, 2013), CHI '13, Association for Computing Machinery, p. 207–216.

[67] Komenda, T., Reisinger, G., and Sihn, W. A practical approach of teaching digitalization and safety strategies in cyber-physical production systems. *Procedia manufacturing 31* (2019), 296–301.

[68] Kormushev, P., Calinon, S., and Caldwell, D. G. Imitation learning of positional and force skills demonstrated via kinesthetic teaching and haptic input. *Advanced Robotics 25*, 5 (2011), 581–603.

[69] Leeper, A. E., Hsiao, K., Ciocarlie, M., Takayama, L., and Gossow, D. Strategies for human-in-the-loop robotic grasping. In *Proceedings of the seventh annual ACM/IEEE international conference on Human-Robot Interaction* (2012), pp. 1–8.

[70] Lehmann, C., Städter, J. P., and Berger, U. Anwendungsbeispiele zur integration heterogener steuerungssysteme bei robotergestützten industrieanlagen. In *Handbuch Industrie 4.0 Bd. 2*. Springer, 2017, pp. 45–58.

[71] Linsinger, M., Stecken, J., Kutschinski, J., and Kuhlenkötter, B. Situational task change of lightweight robots in hybrid assembly systems. *Procedia CIRP 81* (2019), 81–86.

[72] Mason, W., and Suri, S. Conducting behavioral research on amazon's mechanical turk. *Behavior research methods 44*, 1 (2012), 1–23.

[73] Materna, Z., Kapinus, M., Beran, V., Smrž, P., and Zemčík, P. Interactive spatial augmented reality in collaborative robot programming: User experience evaluation. In *2018 27th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)* (2018), IEEE, pp. 80–87.

[74] Nath, K., and Iswary, R. What comes after web 3.0? web 4.0 and the future. In *Proceedings of the International Conference and Communication System (I3CS'15), Shillong, India* (2015), pp. 337–341.

[75] Nielsen, J., and Landauer, T. K. A mathematical model of the finding of usability problems. In *Proceedings of the INTERACT'93 and CHI'93 conference on Human factors in computing systems* (1993), pp. 206–213.

[76] Nielsen, J., and Levy, J. Measuring usability: preference vs. performance. *Communications of the ACM 37*, 4 (1994), 66–75.

[77] Oppenheimer, D. M., Meyvis, T., and Davidenko, N. Instructional manipulation checks: Detecting satisficing to increase statistical power. *Journal of experimental social psychology 45*, 4 (2009), 867–872.

[78] Pan, Z., Polden, J., Larkin, N., van Duin, S., and Norrish, J. Recent progress on programming methods for industrial robots. In *ISR/ROBOTIK* (2010).

[79] Perkel, D. Copy and paste literacy? literacy practices in the production of a myspace profile. *Informal learning and digital media* (2006), 21–23.

[80] Peshkin, M., and Colgate, J. E. Cobots. *Industrial Robot: An International Journal* (1999).

[81] Richert, A., Shehadeh, M., Plumanns, L., Gross, K., Schuster, K., and Jeschke, S. Educating engineers for industry 4.0: Virtual worlds and human-robot-teams: Empirical studies towards a new educational age. In *2016 IEEE Global Engineering Education Conference (EDUCON)* (2016), Ieee, pp. 142–149.

[82] Schuster, K., Gross, K., Vossen, R., Richert, A., and Jeschke, S. Preparing for industry 4.0–collaborative virtual learning environments in engineering education. In *Engineering Education 4.0*. Springer, 2016, pp. 477–487.

[83] Sears, A., and Shneiderman, B. Split menus: effectively using selection frequency to organize menus. *ACM Transactions on Computer-Human Interaction (TOCHI) 1*, 1 (1994), 27–51.

[84] Shaw, A. D., Horton, J. J., and Chen, D. L. Designing incentives for inexpert human raters. In *Proceedings of the ACM 2011 conference on Computer supported cooperative work* (2011), pp. 275–284.

[85] Sirjani, B. maze.co. `https://maze.co/guides/usability-testing/`. Accessed: 2020-06-30.

[86] Sjödin, D. R., Parida, V., Leksell, M., and Petrovic, A. Smart factory implementation and process innovation: A preliminary maturity model for leveraging digitalization in manufacturing moving to smart factories presents specific challenges that can be addressed through a structured approach focused on people, processes, and technologies. *Research-Technology Management 61*, 5 (2018), 22–31.

[87] Sorokin, A., Berenson, D., Srinivasa, S. S., and Hebert, M. People helping robots helping people: Crowdsourcing for grasping novel objects. In *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems* (2010), IEEE, pp. 2117–2122.

[88] Stenmark, M., Haage, M., and Topp, E. A. Simplified programming of re-usable skills on a safe industrial robot: Prototype and evaluation. In *Proceedings of the 2017 ACM/IEEE International Conference on Human-Robot Interaction* (2017), pp. 463–472.

[89] Tellex, S., Kollar, T., Dickerson, S., Walter, M., Banerjee, A., Teller, S., and Roy, N. Understanding natural language commands for robotic navigation and mobile manipulation. In *Proceedings of the AAAI Conference on Artificial Intelligence* (2011), vol. 25.

[90] Thomas, U., Hirzinger, G., Rumpe, B., Schulze, C., and Wortmann, A. A new skill based robot programming language using uml/p statecharts. In *2013 IEEE International Conference on Robotics and Automation* (2013), IEEE, pp. 461–466.

[91] Toris, R., and Chernova, S. Robotsfor.me and robots for you, 2013.

[92] Toris, R., Kent, D., and Chernova, S. The robot management system: A framework for conducting human-robot interaction studies through crowdsourcing. *Journal of Human-Robot Interaction 3*, 2 (2014), 25–49.

[93] Verma, D., and Rao, R. P. Goal-based imitation as probabilistic inference over graphical models. In *Advances in neural information processing systems* (2006), Citeseer, pp. 1393–1400.

[94] Von Ahn, L., and Dabbish, L. Designing games with a purpose. *Communications of the ACM 51*, 8 (2008), 58–67.

[95] Weintrop, D., Afzal, A., Salac, J., Francis, P., Li, B., Shepherd, D. C., and Franklin, D. 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* (2018), pp. 1–12.

[96] White, B. Discovering the future of the web. *Journal of computing and information technology 23*, 1 (2015), 87–93.

[97] Wu, H., Corney, J., and Grant, M. Relationship between quality and payment in crowdsourced design. In *Proceedings of the 2014 IEEE 18th International Conference on Computer Supported Cooperative Work in Design (CSCWD)* (2014), IEEE, pp. 499–504.

[98] Wu, H., Corney, J., and Grant, M. An evaluation methodology for crowdsourced design. *Advanced Engineering Informatics 29*, 4 (2015), 775–786.

[99] Yenorkar, R., and Chaskar, U. Gui based pick and place robotic arm for multipurpose industrial applications. In *2018 Second International Conference on Intelligent Computing and Control Systems (ICICCS)* (2018), IEEE, pp. 200–203.

[100] Yukhimets, D., Sakhnenko, A., and Sych, A. Designing method of information-control systems realization for visual programming robotic manipulators. In *2020 International Conference on Industrial Engineering, Applications and Manufacturing (ICIEAM)* (2020), IEEE, pp. 1–6.

[101] Zaidan, O. F., and Callison-Burch, C. Crowdsourcing translation: Professional quality from non-professionals. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies* (Portland, Oregon, USA, June 2011), Association for Computational Linguistics, pp. 1220–1229.

[102] Zubrycki, I., Kolesiński, M., and Granosik, G. Graphical programming interface for enabling non-technical professionals to program robots and internet-of-things devices. In *International Work-Conference on Artificial Neural Networks* (2017), Springer, pp. 620–631.

Appendix

#	SUS Measurement/ WorkerID	SUS Frequency	SUS Complexity	SUS Easy	SUS Support	SUS Integration	SUS Inconsistency	SUS Learnability	SUS Cumbersome	SUS Confidence	SUS Learn	SUS Score
1	A104BGJEV8I35L	4	4	4	2	3	4	3	3	4	1	60
2	A3FWO3NYW0CXJ9	3	5	3	5	2	3	1	5	4	4	27,5
3	A9AP3ED0K8LS8	3	4	3	4	4	2	3	4	1	4	40
4	A2WYCY1FMQOD5F	2	1	3	2	2	3	3	1	4	2	62,5
5	AMEO4HBYM9LDY	1	3	3	1	1	4	4	4	3	3	42,5
6	AXXJNJ4CG6XLG	3	4	3	4	3	4	2	4	1	4	30
7	A2IKPU88PDQTMF	1	3	2	5	2	2	1	4	1	4	22,5
8	A31PVZ4KDUMEEV	2	4	1	4	2	4	1	4	1	4	17,5
9	A1MONY92DKKHGM	5	3	5	3	3	3	5	1	5	3	75
10	A2WYCY1FMQOD5F	4	2	4	2	3	2	4	2	4	2	72,5
11	A9AP3ED0K8LS8	3	4	1	4	3	1	1	3	2	3	37,5
12	A9AP3ED0K8LS8	1	4	1	4	1	3	4	4	1	4	22,5
13	A1JVUD5XUB9H48	4	2	3	4	3	3	1	4	2	2	45
14	A31PVZ4KDUMEEV	3	1	4	1	3	4	4	2	4	2	70
15	A57R225V29849	4	2	3	3	3	3	1	3	3	4	47,5
16	A3TT6YVWDXD43F	4	1	4	1	3	1	4	1	4	1	85
17	A29NA4283YJ26D	1	4	2	4	1	4	3	5	2	5	17,5
18	A192COVXPX0XEB	5	1	5	1	5	1	1	1	1	5	70
19	A1BZB1RXKGPGTK	3	4	3	2	3	4	2	4	3	2	45
20	A2KAFWN3FZGJID	5	1	5	1	5	1	4	1	5	1	97,5
21	A3CBJ5YIEK61EQ	1	4	4	3	3	3	4	2	4	3	52,5
22	A1BZB1RXKGPGTK	3	4	4	4	2	4	1	5	3	1	37,5
23	AXLMOFXKXSU0B	2	4	2	4	1	4	2	4	2	4	22,5
24	A2KAFWN3FZGJID	5	1	5	1	5	1	4	1	5	3	92,5
25	A11C00ZHBQIYZ	3	4	1	4	1	3	3	4	1	3	27,5
26	A1JVUD5XUB9H48	3	4	3	3	1	4	2	3	1	3	32,5
27	A9AP3ED0K8LS8	2	4	1	4	1	2	1	4	3	4	25
28	A3CBJ5YIEK61EQ	4	3	4	3	3	2	4	3	3	1	65
29	A1Z3NTRGIUZ240	5	5	3	5	3	5	3	4	3	5	32,5
	Average:											**47,41**

Figure 1: SUS all aproved scores

#	SUS Measurement/ WorkerID	SUS Frequency	SUS Complexity	SUS Easy	SUS Support	SUS Integration	SUS Inconsistency	SUS Learnability	SUS Cumbersome	SUS Confidence	SUS Learn	SUS Score
1	A3IC12H51S52ZE	1	3	1	1	2	1	1	2	1	1	45
2	A3BCDYK5ZHQ563	3	1	1	3	1	1	1	1	3	1	55
3	A3QU964EOJXS56	1	3	3	1	3	1	1	3	3	3	50
4	A39980PKUJE4NS	1	3	1	1	3	2	1	3	1	3	37,5
5	ARC708JEQ6WY	3	3	1	3	1	3	1	3	1	3	30
6	A37NFMBIIOYO07	1	1	3	1	3	3	1	1	3	1	60
7	AONHP5DLTIPFX	2	2	1	2	2	2	2	2	1	1	47,5
8	AX9QRJ63BG6T	1	3	2	1	2	1	3	3	1	1	50
9	A3CNIS2UZB4Y2W	1	1	2	1	1	2	3	1	1	1	55
10	ABMX3XETGFAAE	3	1	2	2	3	2	3	2	3	2	62,5
11	A11NPRAGI18E8C	1	1	3	3	1	1	2	1	3	1	57,5
12	A1W05TSPORJPXR	2	4	1	3	2	3	1	3	2	4	27,5
13	A51U5BEIC5XVR	2	3	1	2	3	3	3	4	2	4	37,5
14	A24RQYIDMV7OPK	1	3	1	1	3	1	3	3	1	3	45
15	A24JKHC4HTY6CD	4	3	1	4	3	2	3	4	2	3	42,5
16	A3J5K2E5RS1P2P	2	2	2	2	4	2	2	3	2	1	55
17	A3UWL7Z2LMBIRV	1	1	1	1	1	1	1	1	1	1	50
18	A1ZBGLCWJ4IQ8P	2	1	3	2	2	1	3	1	2	1	65
19	AONHP5DLTIPFX	2	3	1	3	2	3	1	2	2	2	37,5
20	A1NSS58MHS3QXY	4	2	2	3	2	3	2	3	2	3	45
21	A2Z9T8L42FS8KG	1	2	2	3	4	2	2	2	2	2	50
22	AB5N25TN92NC4	2	4	1	1	3	3	4	1	4	2	57,5
23	AADMC4P43V769	5	5	5	3	4	2	5	4	5	3	67,5
24	ASS8RE4GEBJK6	1	3	2	4	1	4	4	3	4	4	35
25	A1CJ4ADVYT1NDO	3	3	1	2	3	2	1	1	3	1	55
26	A17GHP8L89ZSD5	3	1	3	1	2	3	4	1	3	1	70
27	A21K77M8FGO1CP	2	1	2	1	3	2	1	3	1	2	50
28	A61E32KDLN5MZ	4	4	3	3	4	3	1	4	2	4	40
29	A2CMG7PDXXY3ZR	1	1	1	1	1	1	1	1	1	1	50
30	A3UM3L5UM107ZZ	2	1	3	1	1	3	1	2	3	1	55

Figure 2: SUS all rejected scores 1/3

31	A2DSMSEN8GAZUA	3	2	1	1	3	3	3	3	3	2	55
32	A3GOR38ZREUSKR	3	3	3	3	3	1	4	3	2	2	57,5
33	A1JLHMKBGSRROO	2	1	1	3	1	3	1	3	3	1	42,5
34	A1RC17FXGMK03J	1	5	3	5	3	4	2	4	3	3	27,5
35	A22UAN64L0TYJL	2	2	3	2	3	2	3	2	3	2	60
36	A235RBM8NLP6HC	2	3	1	2	1	3	2	1	2	1	45
37	A2GF7O6UQ7V7SZ	2	1	4	4	3	2	3	3	4	3	57,5
38	A6T3P3SIOETHT	4	3	4	4	4	3	4	3	3	4	55
39	AIEJT3DNSEJMT	1	1	1	1	1	1	2	1	2	2	52,5
40	A1NLXD111QHDKZ	2	2	1	1	2	1	1	1	1	1	52,5
41	A54KH61CDVIB6	2	3	3	1	2	2	2	2	2	2	52,5
42	A2PIW46PJENAZ4	4	2	3	2	3	5	3	2	3	3	55
43	A3INW9GHEU7FLU	3	1	3	1	1	1	1	1	1	1	60
44	AYZX9T7RBX3ZP	1	2	1	1	1	1	1	1	1	5	37,5
45	A3MVS9T55AUM2B	1	1	2	1	1	4	2	2	1	2	42,5
46	A2X6JK36FJ3G3E	2	3	2	3	2	2	3	2	1	3	42,5
47	AO2WNSGOXAX52	2	3	1	1	2	2	2	2	2	1	50
48	A2C0IDLBIOPOMG	3	3	2	3	2	2	4	3	3	3	50
49	AQ6I0DFFZTZHH	1	2	1	1	2	1	2	1	2	1	55
50	A13JQTLPWXZD6L	2	3	1	3	2	3	1	1	1	2	37,5
51	A2ASBOP6RG352M	4	3	4	3	4	1	4	1	4	4	70
52	A3TK42KL2AR9LL	2	1	2	1	1	2	1	1	3	1	57,5
53	AQ23W8QKD33UJ	4	1	5	4	3	2	4	2	4	3	70
54	A31472JRQHE5RW	1	4	3	2	3	3	4	3	1	3	42,5
55	A1SMX7EK6H90YH	1	2	2	2	2	3	1	2	2	3	40
56	ATTDVS5IWEE75	3	1	1	2	1	1	1	1	2	1	55
57	A1GMYDH5MKN105	1	1	1	1	1	1	1	1	1	1	50
58	A3TK42KL2AR9LL	1	2	1	1	1	1	3	1	3	2	55
59	A1LA71INO16FVS	1	2	3	1	3	2	2	2	2	2	55
60	A21RZDH4S1Z9T7	1	3	1	1	1	1	1	1	2	1	47,5

Figure 3: SUS all rejected scores 2/3

61	A2X6JK36FJ3G3E	3	2	3	2	2	4	3	3	2	2	50
62	AJISH3SHWGM7L	1	2	1	1	2	2	1	1	2	2	47,5
63	A2TANP6R5WPOKE	1	2	2	2	2	2	2	3	2	3	42,5
64	A24RQYIDMV7OPK	1	3	1	3	1	3	1	3	1	2	27,5
65	A37NFMBIIOYO07	3	2	1	2	1	2	1	3	1	2	40
66	A18WFPSLFV4FKY	2	2	3	1	2	1	2	2	2	1	60
67	AS4BNE6MKMXIU	1	4	4	4	3	5	3	5	4	4	32,5
68	A22UAN64L0TYJL	1	1	3	3	2	3	2	3	2	1	47,5
69	A1ZNQRVW3VFM59	3	3	2	3	3	3	3	3	2	3	45
70	A32CVZVH3C1I20	1	2	3	2	1	2	3	2	1	2	47,5
71	AZ9OXXIQSXDQF	3	3	4	3	2	2	3	4	4	1	57,5
72	A19PVBIU4PBOZP	3	1	3	1	3	3	1	3	1	3	50
73	A2BH59P34O89NC	2	2	1	2	3	2	1	2	3	2	50
74	A3J5K2E5RS1P2P	5	4	4	3	5	5	4	4	3	4	52,5
75	A21K77M8FGO1CP	2	1	2	1	2	1	2	1	2	1	62,5
76	A20RCHED1QJNFN	1	1	1	2	1	1	2	2	2	1	50
77	A1JLHMKBGSRROO	1	1	2	1	2	1	2	1	2	1	60
78	A2CMG7PDXXY3ZR	1	1	1	1	1	1	1	2	1	1	47,5
79	A3JV54EH1LEK4H	1	1	1	1	2	1	2	1	1	1	55
80	A1LJJQAAOGOKEM	2	2	2	1	2	2	2	3	2	2	50
81	A2JI1AH1YMSBC4	2	1	1	2	1	1	2	1	1	2	50
82	A2MLBF3DDKDRFG	1	1	1	1	1	3	3	1	2	2	50
83	AHBBTTTKY0DJY	4	4	4	4	3	4	4	4	3	3	47,5
84	A2Z9T8L42FS8KG	1	2	2	1	2	1	2	2	2	2	52,5
85	A2MPBCPHW9A6TL	3	3	4	3	2	4	4	3	3	1	55
86	A2YOLW621EFX4V	2	1	3	3	3	2	1	2	2	2	52,5
87	A1SKBBTD0PJI6G	4	3	4	4	4	3	4	3	4	4	57,5
88	A119EX2L0DNN1B	2	3	2	2	2	3	2	3	3	3	42,5
	Average:											**49,97**

Figure 4: SUS all rejected scores 3/3

WorkerId	Answer.Nosolve_feedback
A2WYCY1FMQOD5F	just did not understand completely on how to do the task because it said drive and I wasnt sure what was meant by that.I also was not able to book mark the task so I couldnt even complete 6 due to it.
A9AP3ED0K8LS8	As I said in the previous tasks, the software has a problem. The tasks that I ask to save, do not save. And the worst, this time came with a saved task and when updating it always came back. In addition, the cube was in an incorrect position, which meant that it was unable to catch it, with the coordinates indicated in the HIT. It needs to be fixed. Grateful.
A104BGJEV8I35L	I did not get the robot to place. I got it too pick up, but the I did not understand how to use the >dP< function. So when I ran the pick and place, only one of the actions worked.
A3FWO3NYW0CXJ9	N/A
A9AP3ED0K8LS8	N/A
A2WYCY1FMQOD5F	I felt I was able to the best of my abilities
AMEO4HBYM9LDY	I couldn't get the robot to move because the "Play" button is not working. I followed the instructions, dragging the actor "Move to" to placing him between "start" and "stop". On the first attempt, I moved the actor "> dP <dR" and I another "Move to" actor. I clicked on the play button and it didn't work again. I dragged the actor "> dP <dR" to the trash and then chose the actor "Open" and replaced him. Again I clicked on the button and was unable to execute the movement. The notepad with Javascript code was also blank, without a line of code.
AXXJNJ4CG6XLG	None
A2IKPU88PDQTMF	The explanation for saving the file and dragging it to the bookmark tab didn't work for me. Everytime I tried to drag the bookmarks back to the page or the disk it would either give me a blank page or it didn't work.
A31PVZ4KDUMEEV	Like the 1rst time I did this evaluation, I couldn't solve Use Case 6: the robotic arm drops the blue cube between the execution of 'Pick' and the execution of 'Place' and I can't figure out why.
A1MONY92DKKHGM	The Start button does not function properly after the first use, Sometimes the code disappears but I've managed to do all the tasks. Please tell people that, to set the spacial variables, they can use the top right input fields.
A9AP3ED0K8LS8	I managed to do all the tasks. The only problem I had is that the tasks could not be saved. When updating the page, they disappeared. I sent all the screen prints to you. Even so, I was able to finish all the tasks.
A1JVUD5XUB9H48	I was unable to solve Case 2, as it does not mention where to place the (X) actor in the workflow. That is the only case where the problem occured.
A31PVZ4KDUMEEV	Use case 6: You never explicitly say it, but I suppose that you guys wanted the robotic arm to pick up the cube and place that cube elsewhere. The problem I encountered is that between 'Pick' and 'Place' there's a glitch where the arm drops the cube, as if the 'open' command had been used in between. As a result, the arm moves without the cube.
A57R225V29849	I have solved all cases, as best as i could.
A3TT6YVWDXD43F	The last use case wasn't perfectly completed because the previous task which I saved on the bookmark bar wasn't opening again and was showing a new untitled file, I tried many times but it looks like every saved task was having same url and was not loading.
A29NA4283YJ26D	I sloved all tasks
A192COVXPX0XEB	I only understood 5, but i can't run the aplication
A1BZB1RXKGPGTK	I did understand the whole scenario
A2KAFWN3FZGJID	I managed to solve all the cases.
A3CBJ5YIEK61EQ	yes, there are some mistakes in the task, but finally i did those but some things are really hard to understand please use some more specific instructions
A1BZB1RXKGPGTK	true
AXLMOFXKXSU0B	true
A2KAFWN3FZGJID	true
A11C00ZHBQIYZ	true
A1JVUD5XUB9H48	true
A9AP3ED0K8LS8	true
A3CBJ5YIEK61EQ	true
A1Z3NTRGIUZ240	true

Figure 5: No solve approved workers' feedback

WorkerId	Workers' feedback
A2WYCY1FMQOD5F	The site had the task already done and when I tried to make a new one with mturk I was not able to book mark. I was not able to do the last part due to it and I did try and attempt the others just to do them but final part was unable to be completed.
A9AP3ED0K8LS8	The software needs to be fixed. It's easy and intuitive to use, but it has problems.
A104BGJEV8I35L	I found this challenging, but not unlike my inability to work my task augmenter on my Macbook.
A3FWO3NYW0CXJ9	1) Start button is broken, it can only play once. To play the task again I had to refresh and rebuild everytime. Made learning very problematic at first since it leaves no room for mistakes. 2) The set variable and set params are not very intuitive. It's not obvious what they do until you open and read the tooltip about them making "snapshots" of the state. 3) Task five could use more context. It's not explained until task 6 that it's supposed to be chained, and that you should imagine the state of the world is "task Pick just ran", so it doesnâ€™t make a lot of sense at first (e.g. you program the gripper to open but it already was open, and it's not holding the cube).
A9AP3ED0K8LS8	I've done similar tasks before. Now I got the second version, but it remains the same. When updating the browser, the progress is completely lost and does not save the previous saved tasks. You need to fix this!
A2WYCY1FMQOD5F	I felt that some of the instructions could be left confusing if you just read them once. The way the wording was for some of the problems made me double think of what I was doing like messing with the parameters for Z felt like I did not know really what to do because there was Z and ZR
AMEO4HBYM9LDY	This is the second time that I have tried to make this job. In the previous opportunity, I only reported that I had not succeeded, also by a technical problem. This time I, as feedback, recorded my computer screen to show you what happened. I liked the programming environment and I would very much like to be able to solve this task. The link on my Google Drive: https://drive.google.com/drive/folders/13YEotd-c7O36gmyj2aQ7NFMI06n-xVWZ?usp=sharing
AXXJNJ4CG6XLG	None
A2IKPU88PDQTMF	The play button was not working after I pressed it once. I was also not sure when screenshotting the page if you wanted all of the code or just what was available on the screen. I think more instructions for people who might run into those type of questions would be nice. I can also do the assembly over again if needed.
A31PVZ4KDUMEEV	I still cannot repeat the movement manually by repeatedly clicking on the start button; it's still necessary to refresh the page each time. What's new compared to my first time is that I'm apparently getting used to using this environment: I finished it a lot faster and this is reflected in my different answers in this questionnaire. This is very positive.
A1MONY92DKKHGM	I'm on a Macbook Pro, using macos HighSierra 10.13.6 and chrome browser version 89.0.4389.90. If I write the name of the task (near the disk icon) and I use the left/right arrow to move inside the text, the robot visual shifts too. If you hit the "JS" and "Sikuli" button, the javascript "textual" program disappears from the right side.
A9AP3ED0K8LS8	I had already done such a task before and had warned that tasks are not saved and are lost when refreshing the browser page. I noticed that the problem still persists. Please fix it.
A1JVUD5XUB9H48	In case 2 the instructions could have been clear to set the count.
A31PVZ4KDUMEEV	I used Google Chrome but there was a problem anyway: I could only use the 'Start' button once. If I wanted to repeat a motion, then I had to refresh the page and redo everything.
A57R225V29849	Instructions could include pictures
A3TT6YVWDXD43F	Yes, there was a bug saving the coordinates value to the parameter, if we use same actor multiple times then the parameters were setting similar if done through the left side configuration. Apart from that, the help section is also very bad, it literally doesn't helps any newbie understanding how the whole system works. I hope it would become better and it is best suited for much technical sounded person which I am still new. Also the last one was not able to do perfectly as per instructions and the case was not running perfectly. But still I screenshot the flow and hoping that you would accept my hit. Thanks and All the best.
A29NA4283YJ26D	Good luck for your study.
A192COVXPX0XEB	I think the aplication is really cool and helpfull for learn about robotics.
A1BZB1RXKGPGTK	It is very innovative program......It could be better.
A2KAFWN3FZGJID	I had a little more difficulty in the last case, but I managed to solve it.
A3CBJ5YIEK61EQ	use some more specific instructions, in case 1 the play button only be able to use once, i hope i done it in correct way
A1BZB1RXKGPGTK	This program is very good .Kind of innovative to me.
AXLMOFXKXSU0B	Everything was all right with the app, nothing ambiguous.
A2KAFWN3FZGJID	I had some difficulties at first, but then I was able to easily understand how it works.
A11C00ZHBQIYZ	Didn't really understand if Use case 4 goal was to just to load the "Pick" task.
A1JVUD5XUB9H48	no
A9AP3ED0K8LS8	In task four, when updating the page my task "Pick" simply disappeared and I could not complete it. I tried to do it without updating the page but it also didn't work. This is a bug that needs to be fixed.
A3CBJ5YIEK61EQ	the instructions are not clear, instead of instructions you can use tutorial video thats my suggestion.
A1Z3NTRGIUZ240	Three things, there was no clip board to copy the code... And once I realized I needed to update the X,Y,Z in the top right bar before placing the action(e.g. "Move To"), The upload was cumbersome, was I too zip it? or upload them all once and record the URLs? The upload itself is probably harder than the entire task and took the longest. Other than that I found the task enjoyable.

Figure 6: Open approved workers' feedback

WorkerId	Answer.Nosolve_feedback
A3IC12H51S52ZE	none
A3BCDYK5ZHQ563	the problem photo is update not clear
A3QU964EOJXS56	For example a system use case might be "return book when overdue" and would describe the interactions of the various actors (borrower, librarian) with the system in carrying out the end-to-end process
A39980PKUJE4NS	A use case is a tactic used in system analysis to identify, clarify, and categorize system requirements. The use case is made up of a set of possible sequences of interactions between systems and users within an environment and related to a particular goal
ARC708JEQ6WY	For example a system use case might be "return book when overdue" and would describe the interactions of the various actors (borrower, librarian) with the system in carrying out the end-to-end process
A37NFMBIIOYO07	A use case is a tactic used in system analysis to identify, clarify, and categorize system requirements. The use case is made up of a set of possible sequences of interactions between systems and users within an environment and related to a particular goal.
AONHP5DLTIPFX	GOOD
AX9QRJ63BG6T	For example a system use case might be "return book when overdue" and would describe the interactions of the various actors (borrower, librarian) with the system in carrying out the end-to-end process
A3CNIS2UZB4Y2W	NO
ABMX3XETGFAAE	System use cases specify the requirements of a system to be developed. They identify in their detailed description not only the interactions with the actors but also the entities that are involved in the processing. They are the starting point for further analysis models and design activities.
A11NPRAGI18E8C	CLIMATE CHANGE
A1W05TSPORJPXR	no, in some
A51U5BEIC5XVR	none
A24RQYIDMV7OPK	For example a system use case might be "return book when overdue" and would describe the interactions of the various actors (borrower, librarian) with the system in carrying out the end-to-end process.
A24JKHC4HTY6CD	I HAVE NO MORE PROBLEM IN THE CASES.
A3J5K2E5RS1P2P	System use cases specify the requirements of a system to be developed. They identify in their detailed description not only the interactions with the actors but also the entities that are involved in the processing. They are the starting point for further analysis models and design activities.
A3UWL7Z2LMBIRV	define goals
A1ZBGLCWJ4IQ8P	GOOD
AONHP5DLTIPFX	GOOD
A1NSS58MHS3QXY	System use cases specify the requirements of a system to be developed. They identify in their detailed description not only the interactions with the actors but also the entities that are involved in the processing. They are the starting point for further analysis models and design activities.
A2Z9T8L42FS8KG	the gradients the writing lives along, how to improve from wherever we start. 7. ... Ivar's course on use cases in the early 1990's, as part of evaluating Ivar's ...
AB5N25TN92NC4	to spend time analyzing the testing results .
AADMC4P43V769	Case 2,3,4,5, ans 6, i really had no clue where to even begin. I tried following the directions but just could not do it. I honestly di try tho. I spent a lot of time trying to figure it out. I hope i can still payed for this because i did spend a lot of time trying to solve it.
ASS8RE4GEBJK6	yes
A1CJ4ADVYT1NDO	didn't understand certain instruction. Which one?
A17GHP8L89ZSD5	DON'T UNDERSTAND
A21K77M8FGO1CP	The most obvious actor in the case of any website is a visitor to the site. ... Instead, use cases define goals and purpose: the problems we are trying to solve.
A61E32KDLN5MZ	when I don't understand, I try to solve the problem
A2CMG7PDXXY3ZR	the gradients the writing lives along, how to improve from wherever we start. 7. ... Ivar's course on use cases in the early 1990's, as part of evaluating Ivar's . needed the model, because I still couldn't understand what Ivar was. assured me that most of what I have to say about a
A3UM3L5UM107ZZ	Nothing.

Figure 7: No solve rejected workers' feedback 1/2

WorkerId	Workers' feedback
A22UAN64L0TYJL	Some examples of positive feedback are contractions in child birth and the ripening of fruit; negative feedback examples include the regulation of blood glucose levels and osmoregulation
A235RBM8NLP6HC	GOOD
A2GF7O6UQ7V7SZ	robot not work, is the only error i found, nothing of cases work on.
A6T3P3SIOETHT	NA
AIEJT3DNSEJMT	nice survey
A1NLXD111QHDKZ	You: That is so great to hear. We really try our best to [do what you're being praised for]. And thank you so much for taking the time to provide your feedback. Customer: For sure, thank you for providing such great service!
A54KH61CDVIB6	everything quite ok i tried my best
A2PIW46PJENAZ4	I thought the system was easy to use.
A3INW9GHEU7FLU	good
AYZX9T7RBX3ZP	This way, you'll always know what your customers REALLY want and how ... Analytics and data gives us all sorts of insights into what our customers want ... There are two basic ways to approach surveys. And before we jump to any conclusions, we want to get as much feedback That kind of information is invaluable.
A3MVS9T55AUM2B	Yes, for example, some links do not work, and upon completion of some tasks, do not get the code.
A2X6JK36FJ3G3E	it was so good and so easy to share.
AO2WNSGOXAX52	UX
A2C0IDLBIOPOMG	NO
AQ6I0DFFZTZHH	good
A13JQTLPWXZD6L	very engging
A2ASBOP6RG352M	N/A
A3TK42KL2AR9LL	NO
AQ23W8QKD33UJ	please make instruction simple and clear
A31472JRQHE5RW	It doesn't connect me to robot
A1SMX7EK6H90YH	It helps us to see ourselves from other people perspectives, identify what skills to work ... You should target to make the whole feedback sharing process positive. ... You might want to improve performance and communication, ...
ATTDVS5IWEE75	GOOD
A1GMYDH5MKN105	NOTHING
A3TK42KL2AR9LL	NO
A1LA71INO16FVS	is very good
A21RZDH4S1Z9T7	GOOD
A2X6JK36FJ3G3E	good.
AJISH3SHWGM7L	nothing
A2TANP6R5WPOKE	NO
A24RQYIDMV7OPK	good
A37NFMBIIOYO07	good
A18WFPSLFV4FKY	NO
AS4BNE6MKMXIU	No.
A22UAN64L0TYJL	NOTHING
A1ZNQRVW3VFM59	Very nice
A32CVZVH3C1I20	Some examples of positive feedback are contractions in child birth and the ripening of fruit; negative feedback examples include the regulation of blood glucose levels and osmoregulation
AZ9OXXIQSXDQF	ok
A19PVBIU4PBOZP	nice
A2BH59P34O89NC	could'nt find certain button
A3J5K2E5RS1P2P	YES
A21K77M8FGO1CP	NOTHING
A20RCHED1QJNFN	THE WORK VERY IMPORTAN
A1JLHMKBGSRROO	NO
A2CMG7PDXXY3ZR	nothing
A3JV54EH1LEK4H	good
A1LJJQAAOGOKEM	NO
A2JI1AH1YMSBC4	NOTHING
A2MLBF3DDKDRFG	VERY INTERESTING
AHBBTTTKY0DJY	onnum milla punda
A2Z9T8L42FS8KG	nathing
A2MPBCPHW9A6TL	pothitu poda punda
A2YOLW621EFX4V	good
A1SKBBTD0PJI6G	NA
A119EX2L0DNN1B	NO

Figure 8: No solve rejected workers' feedback 2/2

WorkerId	Workers' feedback
A3IC12H51S52ZE	none
A3BCDYK5ZHQ563	nothing
A3QU964EOJXS56	Ask in as short a way as possible. No matter how complex the situation, keep your email brief and to the point. ... Be clear about what you're asking for feedback about. People in a hurry don't read long sentences. .
A39980PKUJE4NS	Ask in as short a way as possible. No matter how complex the situation, keep your email brief and to the point.
ARC708JEQ6WY	Know why you're asking for customers feedback. Ask yourself why you're asking for customer feedback. ...
A37NFMBIIOYO07	Ask in as short a way as possible. No matter how complex the situation, keep your email brief and to the point. ... Be clear about what you're asking for feedback about. People in a hurry don't read long sentences. .
AONHP5DLTIPFX	NOTHING
AX9QRJ63BG6T	If an employee goes the extra mile: â€œ Last week I asked you to give me a helping hand on Project X as I was struggling to keep up with the workload. ... When a colleague overcomes and obstacle
A3CNIS2UZB4Y2W	NO
ABMX3XETGFAAE	YES
A11NPRAGI18E8C	PA
A1W05TSPORJPXR	in very useful work and better than to next thanking you!
A51U5BEIC5XVR	none
A24RQYIDMV7OPK	Establishes and maintains a good working relationship. Respectful to the employees' idea and is never overbearing. Manager recognizes the value of people skills and experience. The manager gives personal interest to his management activities
A24JKHC4HTY6CD	ANYTHING FROM THE INSTRUCTION.
A3J5K2E5RS1P2P	YES GOOD
A3UWL7Z2LMBIRV	Feedback giving s
A1ZBGLCWJ4IQ8P	GOOD
AONHP5DLTIPFX	UK
A1NSS58MHS3QXY	YES GOOD
A2Z9T8L42FS8KG	It helps us to see ourselves from other people perspectives, identify what skills to work ... You should target to make the whole feedback sharing process positive. ... You might w
AB5N25TN92NC4	conducting dozens of usability tests if we are not able to spend time analyzing their results is a complete waste of time, So need some shortcuts to analyze it .
AADMC4P43V769	just hard to do
ASS8RE4GEBJK6	no
A1CJ4ADVYT1NDO	anything from instructions to UX
A17GHP8L89ZSD5	ux
A21K77M8FGO1CP	It helps us to see ourselves from other people perspectives, identify what skills to work ... You should target to make the whole feedback sharing process positive. ... You might want to improve performance and communication,
A61E32KDLN5MZ	everything is easy, you just have to read the instructions
A2CMG7PDXXY3ZR	Analytics and data gives us all sorts of insights into what our customers want from our business. ... Why are people using one feature three times as often as another? Content Marketing our team creates epic content that will get shared, get links,. If you want to feature a survey
A3UM3L5UM107ZZ	anything instruction from USA
A2DSMSEN8GAZUA	NO
A3GOR38ZREUSKR	Good
A1JLHMKBGSRROO	NO
A1RC17FXGMK03J	None

Figure 9: Open rejected workers' feedback 1/2

WorkerId	Answer.Nosolve_feedback
A2DSMSEN8GAZUA	I SOLVED ALL QUSTIONS
A3GOR38ZREUSKR	It's a short quiz that doesnâ€™t require a lot of resources to administer, so very useful if youâ€™re constrained by budget but still need good info fast
A1JLHMKBGSRROO	GOOD
A1RC17FXGMK03J	No, case are solve.
A22UAN64L0TYJL	he gradients the writing lives along, how to improve from wherever we start. 7. ... Ivar's course on use cases in the early 1990's, as part of evaluating Ivar's ... I needed the model, because I still couldn't understand what Ivar was ... assured me that most of what I have to say about a use case fits within one of the UML ellipses,.
A235RBM8NLP6HC	NONE
A2GF7O6UQ7V7SZ	I use the robot and not have a answer for the server. "errror: cannot connect to the robot"
A6T3P3SIOETHT	Unable to understand the process to activate.
AIEJT3DNSEJMT	I can solve it
A1NLXD111QHDKZ	A use case is a description of how a person who actually uses that process or system will accomplish a goal. It's typically associated with software systems, but can be used in reference to any process. For example, imagine you're a cook who has a goal of preparing a grilled cheese sandwich.
A54KH61CDVIB6	tuff to understand and continue
A2PIW46PJENAZ4	I found the system very cumbersome to use
A3INW9GHEU7FLU	none
AYZX9T7RBX3ZP	To be complete, every use case must describe a specific goal and the actors that will perform ... Instead, use cases define goals and purpose: the problems we are trying to solve.To see some more use cases in action, visit guibot.com, which offers alternativeYou can think of a use case as a collection of scenarios.
A3MVS9T55AUM2B	The problem lies in the time and lack of experience in this area. Also, there are problems related to devices and connections.
A2X6JK36FJ3G3E	There was nothing could problem occur
AO2WNSGOXAX52	NO
A2C0IDLBIOPOMG	Where to work
AQ6I0DFFZTZHH	A use case is a written description of how users will perform tasks on your website. It outlines, from a user's point of view, a system's behavior as it responds to a request.
A13JQTLPWXZD6L	i dont undersant
A2ASBOP6RG352M	N/A
A3TK42KL2AR9LL	DIDN'T UNDERSTAN CERTAIN INSTRUCTION
AQ23W8QKD33UJ	instruction not clear for new user and start button not working
A31472JRQHE5RW	it showed me an error
A1SMX7EK6H90YH	the gradients the writing lives along, how to improve from wherever we start. 7. ... Ivar's course on use cases in the early 1990's, as part of evaluating Ivar's ..
ATTDVS5IWEE75	true
A1GMYDH5MKN105	true
A3TK42KL2AR9LL	true
A1LA71INO16FVS	true
A21RZDH4S1Z9T7	true
A2X6JK36FJ3G3E	true
AJISH3SHWGM7L	true
A2TANP6R5WPOKE	true
A24RQYIDMV7OPK	true
A37NFMBIIOYO07	true
A18WFPSLFV4FKY	true
AS4BNE6MKMXIU	true

Figure 10: Open rejected workers' feedback 2/2

Exercise: Comparing Robot Programming Environment

Course: 330.288 Montage II: Advanced Manufacturing (SE 2,0) 2020W

Instructor: Tudor Ionescu (tudor.ionescu@tuwien.ac.at)

Estimated time: 1-2h for first time users

Goal: The goal of the exercise is to conduct an evaluation of a robot programming environment by following a procedure similar to the one used in *Weintrop, D., Afzal, A., Salac, J., Francis, P., Li, B., Shepherd, D.C. and Franklin, D., 2018, April. 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 (pp. 1-12).*

Prerequisites:

1. Make sure you have a computer that supports the Google Chrome browser. Download the browser if you have not installed it.
2. Check that you can access http://assembly.comemak.at
3. Get familiar with the Assembly programming environment by experimenting with it.

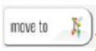
Exercise instructions:

You are asked to create 5 robot tasks (i.e., programs) in Assembly. The programs will have an increasing level of difficulty. Take a screenshot of the window and save it to a folder. You will have to upload a few screenshots with the tasks you create to TUWEL.

Make sure to refresh the browser after completing each program so that a fresh task appears in the window.

The units used for the (x,y,z) coordinates are millimeters. The units used for the rotation angles along the spatial axes (rx, ry, rz) are degrees.

You will receive a questionnaire in a separate document to fill out and upload to TUWEL together with the screenshots. Complete all tasks before filling the questionnaire.

Task 1: Create a program that performs a robot motion between two points. You can freely choose the waypoints. Drag the "move to" actor (i.e., 

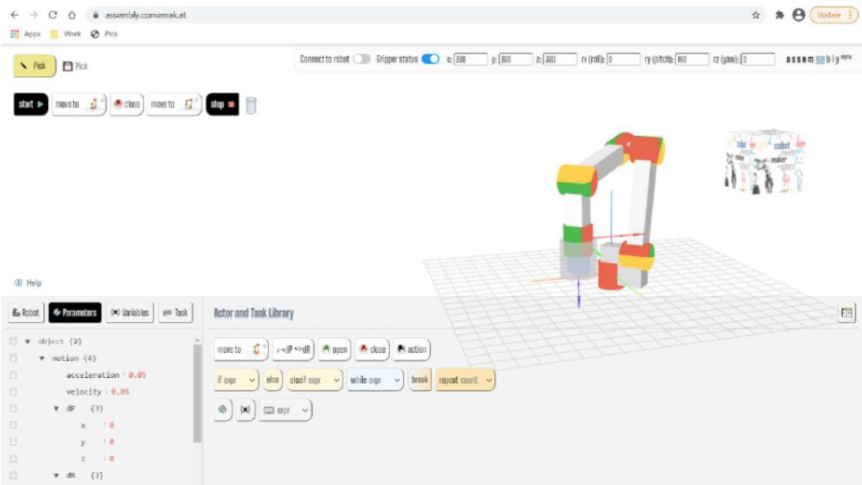
) to a location between the "Start" and "Stop" actors in the workflow (i.e., robot task or program) to create/memorize a waypoint. Then move the robot to another location and add another "move to" actor to the workflow to create/memorize the second waypoint in the program. Repeat the motion a few times by clicking the play button.

Task 2: Pick the small blue cube by driving the robot to x=300, y=300, z=89 and closing the gripper. Use the "moveTo" and "gripper close" actors from the "Actor and Task Library" to remember the pick location. Then drive the robot to x=300, y=300, z=300 and memorize this location as a waypoint. Rename the task from "Untitled" to "Pick" and save it by dragging it to the browser's bookmarks bar.

Take a screenshot of this task to upload later to TUWEL.

The result should look like this:

Figure 11: *"Montage II: Advanced manufacturing[24]"* use cases 1/4



Task 3: Refresh the browser window to start with a clean task. Then simply drag the empty "Untitled" task to the bookmarks bar to save it. You will need it in the next task.

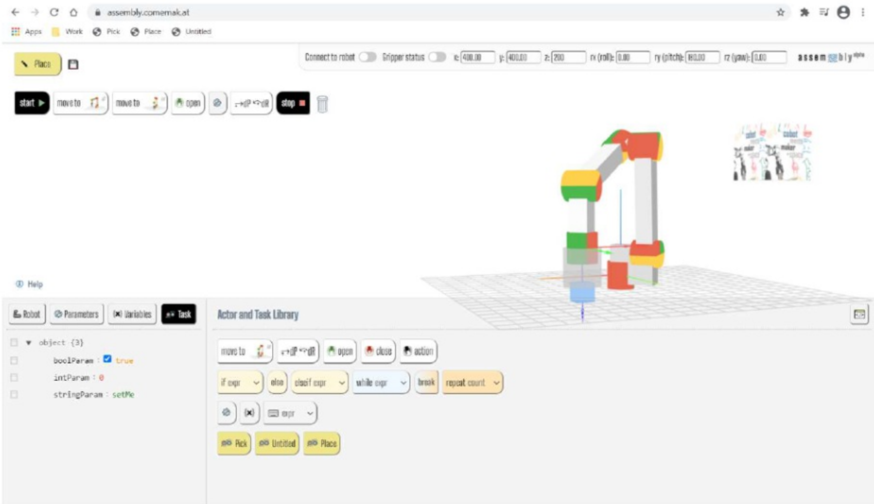
Task 4: Load the "Pick" task you have just created and saved by clicking on the corresponding bookmark. Run the task to pick the blue cube up.

Task 5: Drive the robot to a "pre-place" location at about z=200 above the table, then down to the table level (z=89) and open the gripper to release the gripper. Then use the "relative motion" actor (i.e., →dP ↩dR) to drive the robot back up to z=200. The relative motion actor uses dP, dR parameters. So, select the parameters tab (i.e., Parameters) and set z=111 so as for the robot to go to z=200. After setting this parameter, drag and drop the "set parameters" actor (i.e.,) into the workflow, anywhere before the relative motion actor. Rename the task to "Place" and save it to the bookmarks bar.

Take a screenshot of this task to upload later to TUWEL.

The result should look like this:

Figure 12: *"Montage II: Advanced manufacturing[24]"* use cases 2/4



Task 6: Load the “Pick” and “Place” tasks by clicking on the corresponding bookmarks. You will notice that they appear in the “Actor and Task Library” (if they weren’t already there). You can now use these tasks like any other actor. Click on the “Untitled” bookmark to load a fresh task without “losing” the “Pick” and “Place” tasks from the library. Finally create a “PickAndPlace” task using the “Pick” and “Place” tasks as actors and save it to the bookmarks bar. Test the task by refreshing the browser window, then loading the “Pick” and “Place” tasks to the “Actor and Task Library”, and then load the “PickAndPlace” task and run it.

Take a screenshot of this task to upload later to TUWEL.

The result should look like this:

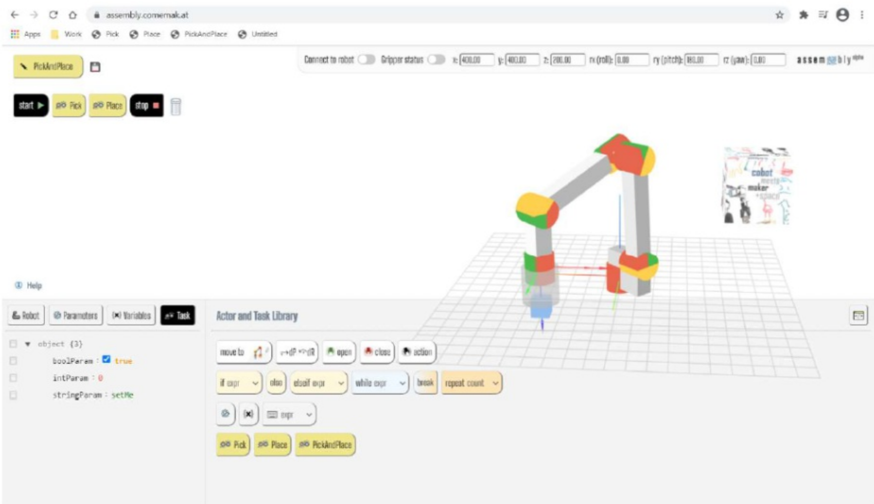


Figure 13: *"Montage II: Advanced manufacturing[24]"* use cases 3/4

Task 7: Line in task 1, create a program to perform a robot motion between two points for 10 times. You can freely choose the waypoints and you can use the "repeat" block (i.e., repeat count) to implement the loop. Note that, when added to a workflow (i.e., robot task or program), the "repeat" block has two parts – a head and a tail part. Put whatever you want to repeat between the head and the tail of the "repeat" actor. Also, the "repeat" actor uses a variable named "count." Open the variables panel (i.e., (x) Variables) and set count = 10. Make sure to store this value in the program by adding the "Set variables" (i.e., (x)) actor to the workflow, before the "repeat" block.

Take a screenshot of this task to upload later to TUWEL.

Task 8: Think of a way to create new, reusable versions of the "Pick" and "Place" tasks by using the relative motion actor instead of the moveTo actor whenever possible inside a task. Create a new version of the "PickAndPlace" task, in which the moveTo actor is only used outside the "Pick" or "Place" tasks. The goal is to reuse the Pick and Place tasks to move the cube to different locations without having to modify them.

Take a screenshot of this task to upload later to TUWEL.

The new PickAndPlace task should look something like this:

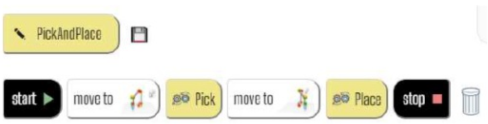


Figure 14: *"Montage II: Advanced manufacturing[24]"* use cases 4/4