TU WIEN Informatics

Erweiterung der adaptiven Aufgabenteilung zwischen einem Menschen und einem Roboter unter Berücksichtigung des Lernens am Arbeitsplatz

DIPLOMARBEIT

zur Erlangung des akademischen Grades

Diplom-Ingenieur

im Rahmen des Studiums

Wirtschaftsinformatik

eingereicht von

Thomas Wendelin, BSc
Matrikelnummer 01228866

an der Fakultät für Informatik

an der Technischen Universität Wien

Betreuung: Univ.-Prof. Dr.-Ing. Sebastian Schlund
Mitwirkung: DI Bernd Hader, BSc
Dr.in. techn. Christina Schmidbauer, Msc, Bsc

Wien, 26. September 2023

Thomas Wendelin | Sebastian Schlund

Technische Universität Wien
A-1040 ▪ Karlsplatz13 ▪ Tel. +43-1-58801-0 ▪ www.tuwien.at

Enhancing adaptive task sharing between a human and robot considering workplace learning

DIPLOMA THESIS

submitted in partial fulfillment of the requirements for the degree of

Diplom-Ingenieur

in

Business Informatics

by

Thomas Wendelin, BSc
Registration Number 01228866

to the Faculty of Informatics

at the TU Wien

Advisor: Univ.-Prof. Dr.-Ing. Sebastian Schlund
Assistance: DI Bernd Hader, BSc
Dr.in. techn. Christina Schmidbauer, Msc, Bsc

Vienna, 26th September, 2023

Thomas Wendelin Sebastian Schlund

Erklärung zur Verfassung der Arbeit

Thomas Wendelin, 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, 26 September 2023

____

Thomas Wendelin

Danksagung

Ich möchte an dieser Stelle meinen Dank an alle aussprechen, die mir unterstützend bei der Erstellung dieser Diplomarbeit zur Seite gestanden sind.

Speziell Bedanken möchte ich bei meinen Betreuern Univ-.Prof. Dr.-Ing. Sebastian Schlund, DI Bernd Hader und Dr.in techn. Christina Schmidbauer. Auf ihre Unterstützung und Anregungen konnte ich immer zählen. Besonders für das Feedback und die wertvollen Ideen möchte ich mich bedanken.

Außerdem möchte ich mich bei meinen Eltern Karin und Werner sowie meiner Schwester Irene bedanken, die mich während des Studiums unermüdlich unterstützt haben.

Ich möchte mich besonders bei meiner Freundin Hanna für ihre Geduld und ihre motivierenden Worte bedanken.

Zu guter Letzt möchte ich mich bei Romeo für die tatkräftige Unterstützung bedanken.

Kurzfassung

Um die Effizienz von Prozessen zu verbessern und Monotonie zu vermeiden, werden Prozesse teilweise automatisiert. Kollaborationsroboter sind Robotersysteme, die zur (Teil-)Automatisierung eingesetzt werden können und Hand in Hand mit Menschen im selben kollaborativen Arbeitsbereich arbeiten. Dabei stellt sich immer die Frage nach der Aufgabenzuweisung. Der Stand der Technik ist eine starre und statische Aufgabenzuweisung. Die adaptive Aufgabenteilung (ATS) in der Montage zielt darauf ab, die Flexibilität der Produktionsprozesse zu verbessern. Bei diesem Ansatz wird nicht im Voraus entschieden, welche Aufgabe dem Menschen oder dem Roboter zugewiesen wird, sondern es können einige Aufgaben definiert werden, die von beiden ausgeführt werden können. Für diesen Ansatz wurde bereits ein Software-Prototyp implementiert, der als Werkerassistenzsystem (WAS) für ATS dient. Allerdings gibt es noch einige ungelöste Probleme. Der menschliche Arbeiter sollte mit Informationen über die Auswirkungen der Aufgabenzuweisung versorgt werden, um qualitative Entscheidungen zu gewährleisten. Derzeit werden im Assistenzsystem keine Informationen über die Auswirkungen von Änderungen im Prozess bereitgestellt, wenn Aufgaben vom Menschen auf den Roboter oder umgekehrt verteilt werden. Darüber hinaus ist in einer zunehmend technisierten Arbeitswelt das Erlernen neuer Fähigkeiten von entscheidender Bedeutung. Das Potenzial zum Erlernen neuer Fähigkeiten wird durch die ATS-Methode gefördert. Der bestehende WAS-Prototyp berücksichtigt jedoch nicht das Lernen am Arbeitsplatz in Verbindung mit der Aufgabenzuweisung. Um diese Probleme zu lösen, verwenden wir die Forschungsmethodik Design Science. Die von uns vorgeschlagene Lösung wird in mehreren Iterationen implementiert und zielt darauf ab, benutzerfreundlich zu sein, das Lernen am Arbeitsplatz zu ermöglichen und Informationen zur Unterstützung der Entscheidungsfindung zu liefern. Darüber hinaus wird die Lösung mit einer Online-Nutzerstudie evaluiert, deren Ergebnis zeigt, dass die Lösung als benutzerfreundlich angesehen werden kann.

Abstract

To improve the efficiency of processes and avoid monotony, the processes are partially automated. Collaborative robots are robotic systems that can be used for (partial) automation and to work hand in hand with humans in the same collaborative workspace. This always brings with it the issue of job or task allocation. The state of the art is a rigid and static allocation of tasks. Adaptive task sharing (ATS) in assembly strives to improve flexibility in production processes. This approach does not decide in advance which task is dedicated to the human or the robot; instead, some tasks can be defined to be carried out by both of them. A software prototype has already been implemented for this approach, which serves as a worker assistance system (WAS) for ATS. However, there are some unsolved problems. The human worker should be provided with information about the effects of allocating tasks to ensure qualitative decisions. Currently, no information is provided in the assistance system about the impacts of changes to the process if tasks are assigned from the human to the robot or vice versa. Additionally, in a world of work that is becoming increasingly technological, learning new skills is crucial. The potential for learning new skills is promoted by the ATS method. However, the existing WAS prototype does not consider workplace learning in combination with task allocation. To overcome these problems, we use the design science research methodology to conduct the research. Our proposed solution is implemented in multiple iterations and aims to be user friendly, enable workplace learning, and provide information to support decision-making. Furthermore, the solution was evaluated with an online user study, which showed that it can be considered user friendly.

Table of Contents

Kurzfassung IX

Abstract XI

Table of Contents XIII

1 Introduction 1

1.1 Motivation and Problem Statement 1

1.2 Expected Results 3

1.3 Methodology and Approach 4

1.4 Structure of the Work 6

2 Theoretical Fundamentals 7

2.1 Human-Robot Interaction 8

2.1.1 Cobots and Task Assignment 8

2.1.2 Adaptive Task Sharing 10

2.2 UI/UX Design 15

2.2.1 Usability 15

2.2.2 User Experience 16

2.3 Learning Opportunities 18

2.3.1 Learning Theory 18

2.3.2 Learning Curve Models 19

2.3.3 Prediction of Learning Time 19

3 State of the Art 22

3.1 Systematic Literature Review 22

3.1.1 Summary of the State of the Art 25

4 Implementation 33

4.1 First Iteration 34

4.1.1 Parameterization 35

4.1.2 Dashboard 37

4.1.3 User Properties 41

4.1.4 Functionality of Modeler 42

4.2 UI Expert Feedback Cycle (Second and Third Iteration) 46

4.2.1 Heuristic Evaluation 46

4.2.2 Resolution of Guideline Violations ... 50
4.3 Proof of Concept ... 52
5 Evaluation ... 58
5.1 Study Design ... 58
5.1.1 Hypothesis ... 58
5.1.2 Study Procedure ... 58
5.1.3 Technical Implementation ... 64
5.2 Participants ... 64
5.2.1 Data Cleansing ... 64
5.2.2 Demographic Data ... 65
5.3 Results ... 67
5.3.1 Correct Solving of the Task ... 67
5.3.2 Usability/User Experience ... 68
5.3.3 Qualitative Data ... 71
6 Conclusion, Discussion, and Outlook ... 73
6.1 Conclusion ... 73
6.2 Discussion and Outlook ... 75
7 List of References ... 76
8 List of Figures ... 83
9 Equation Directory ... 86
10 List of Tables ... 87
11 Appendix ... 88
11.1 Questionnaire ... 88
11.2 UI guidelines ... 91

1 Introduction

1.1 Motivation and Problem Statement

Cobots are robotic systems that can work hand in hand with humans in the same collaborative workspace at the same time whilst the human worker is supported by the robot [1]. When a cobot is integrated at a workstation, task allocation between the two agents is relevant. To replace rigid and static task allocation paradigms, adaptive task sharing (ATS) in assembly strives to improve flexibility in production processes [2]. This ATS approach does not decide in advance which task is dedicated to the human or the robot. Instead, some tasks can be defined to be carried out by both of them, which makes the task shareable [3]. For this approach, a software prototype was implemented, which serves as a worker assistance system (WAS) for ATS, where the worker is offered a user interface (UI) to create a human, a robot, or a shareable task, which then can be allocated adaptively [3]. Additionally, the prototype connects to the cobot, which enables execution of the process [3].

However, as shown in Figure 1, apart from the possibility of creating tasks, allocating tasks, and executing processes, there is no decision-support information on the effects of task allocation.

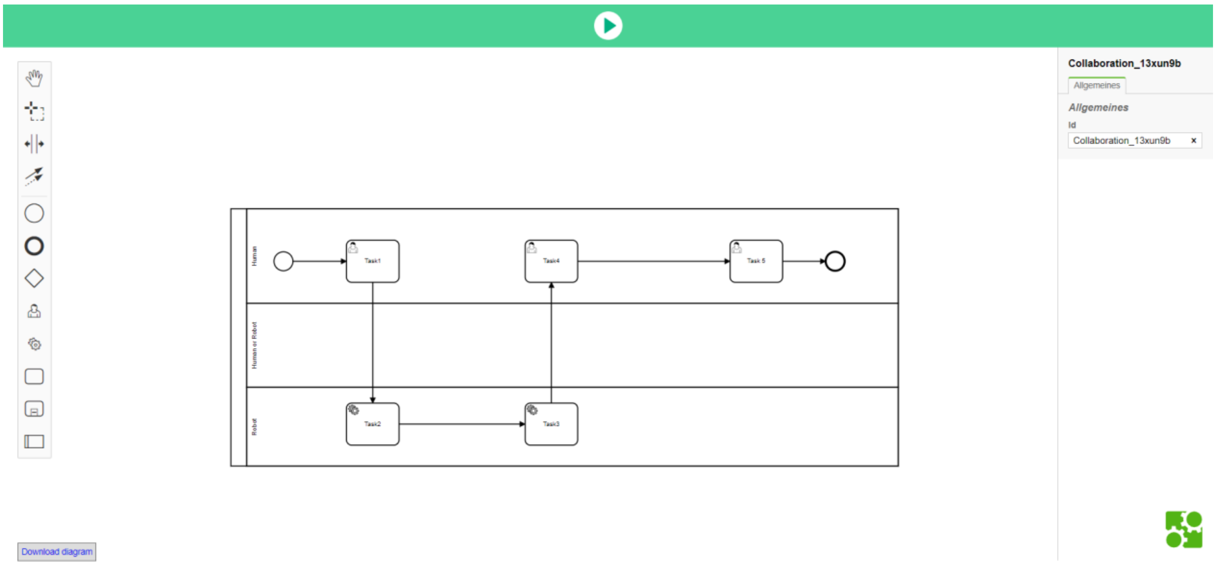

Figure 1: User Interface of existing prototype for ATS (own screenshot)

Additional criteria should be visible for the worker to support the decision-making process of task assigning [2]. As the human should concentrate on the work itself instead of thinking about other things around the process [4], more assistance for the worker is needed. As the costs and times of human and robot task execution vary, information regarding changes to the process execution time or process costs can support the worker in deciding how to allocate tasks. Moreover, there is no decision-support information considering workplace learning. According to Buxbaum et al. [5], human-robot interaction (HRI) design necessitates a working design that is competency oriented:

> *"The technical possibilities should allow an individualized solution, which allows to assign tasks to the human being according to the current skill level." [5, p. 568]*

New workers especially need to focus on the assigned task and need clarity regarding their current skill level. However, the existing prototype does not consider such learning opportunities for learning new tasks in combination with task allocation and does not recommend that a worker consider a task as learned. Furthermore, the WAS prototype lacks additional task-level parameters necessary to calculate process costs and times and for workplace learning. This is also mentioned in the discussion and outlook chapter of the associated master thesis of the prototype [6], where the author states that parameterization could be investigated as well as whether the UI will remain user friendly after the integration.

The research questions for this thesis are as follows:

RQ1: Which approach is suitable to realize adaptive task sharing between a human and robot considering learning opportunities regarding learning new tasks?

RQ2: What is a suitable design for a worker assistance system to improve decision support for adaptive task sharing and enable workplace learning?

RQ3: What is a user-friendly way to implement this approach?

1.2 Expected Results

The result of this thesis should be an approach for enabling learning opportunities and how to implement it into an existing prototype to include the approach and improve worker decision support. The existing prototype enables the easy creation and allocation of collaborative processes, which function as a worker assistance system. However, as stated in Section 1.1 and shown in Figure 1, there is no decision-support information on the impact of task allocation. Therefore, more information should be provided to the worker, which should be displayed in a distinct area that also considers workplace learning. This requires the parameterization of each task.

Figure 2 depicts what this could look like. The respective parameters for each task can be added to the right tool bar. The decision information area in the upper section can display different information such as process time and costs, depending on whether tasks are assigned to the robot or the human, as well as learning progress for each task and if a task can be considered learned.

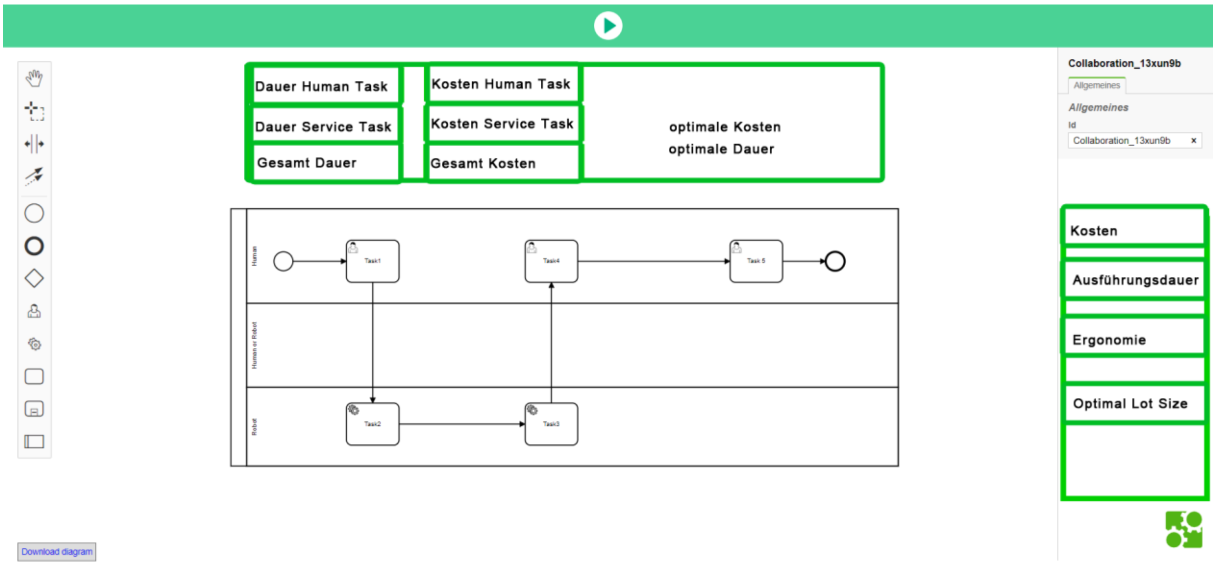

Figure 2: Draft of expected user interface (own figure)

Summarized, the expected results should be a user-friendly user interface that includes the following:

- Information on costs and process/task durations based on the task assignment
- Decision-making support
- Integration of learning opportunities

Moreover, the evaluation should indicate whether the implemented prototype can be considered user friendly.

1.3 Methodology and Approach

The design science research methodology (DSRM) by Peffers et al. [7] is used to conduct the research. The proposed process model by Peffers et al. [7] is depicted in Figure 3.

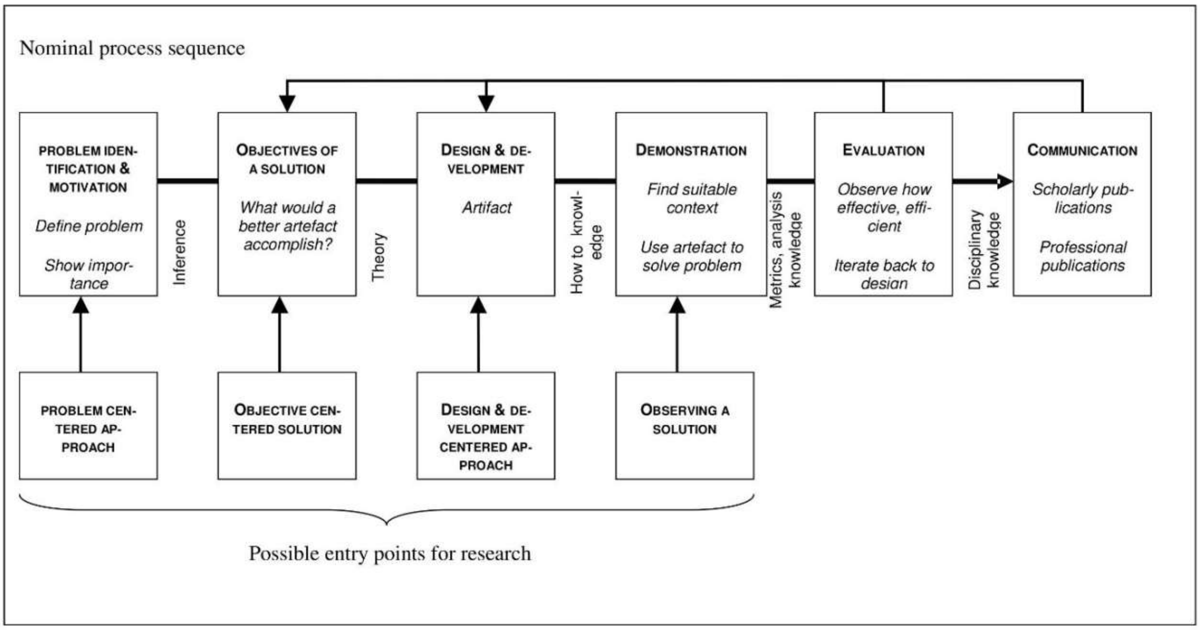

Figure 3: DSRM process model [7]

The approach is divided into six activities as follows:

1. Problem identification and motivation: In the first step, the problem is defined and justified for its relevance. In addition, the state of the art is analyzed. To do this, a systematic literature review [8] is performed.

2. Objectives to Solution: In the design science research process, Peffers [7] defines the objectives to be either quantitative or qualitative. Hence, as we are developing an artifact that is not yet addressed, we are defining qualitative objectives. These objectives are evaluated in a later step. Additionally, required resources like the state of problems are considered to determine the objectives.

3. Design and development: In this phase, the artifactual solution is built, which Hevner et al. [9] defines as constructs, models, methods, or instantiations. We implement the enhancements for the existing prototype in multiple iterations. The enhancements include further parameters at task level, a dashboard, and a learning curve model that defines when a new task is considered as learned. After that, based on UI guidelines, a heuristic evaluation is performed by experts. In the next iteration, the feedback is used to resolve guideline violations.

4. Demonstration: In the fourth step, the efficacy of the artifact is demonstrated to solve the problem [7]. We deploy one industrial use case to show that the developed solution solves the problem.

5. Evaluation: In the fifth step, we observe and measure how well the artifact supports a solution to the problem [7]. If the objectives are not met well enough, the researchers can either iterate back to the design and development step and enhance the artifact or continue to the next step and leave further improvement for future projects [7].
   We deploy an online user study where participants must complete different tasks. Afterward, they are asked to fill out the System Usability Scale Questionnaire [10] and the User Experience Questionnaire (UEQ) [11].

6. Communication: This scientific work is published as a master's thesis and is therefore accessible for further research.

1.4 Structure of the Work

This thesis is divided into six chapters.

Chapter 2 provides an explanation of the key topic areas that are important for the work. This includes human-robot interaction, UI/UX design, and learning opportunities.

Chapter 3 discusses the state of the art, including a systematic literature review to provide an overview of recent developments and relevant previous works.

In **Chapter 4,** the enhancements to the prototype are implemented in three iterations. The first iteration includes an iterative implementation of the requirements with feedback loops. In the second iteration, experts perform a heuristic evaluation based on UI guidelines, and in the third iteration, the feedback is used to resolve violations of guidelines.

Chapter 5 is devoted to evaluation. In this chapter, an online user study is conducted to assess the user friendliness of the system. Therefore, the SUS questionnaire and UEQ are used.

In **Chapter 6,** the thesis results are summarized and critically discussed. The research questions are answered, and an outlook for further research is given.

2 Theoretical Fundamentals

This chapter breaks down and discusses in depth the theoretical background needed for this thesis. Because this work emphasizes enhancing adaptive task sharing and integrating learning into the system, we are looking at three topics: human-robot interaction, UI/UX design, and learning opportunities.

Human-Robot Interaction

In this section, we dive into the topics of human-robot interaction, collaborative robots, and task allocation.

UI/UX

This section is dedicated to discussing usability and user experience and how they can be evaluated.

Learning opportunities

This section discusses learning and estimating learning time in manufacturing and assembly.

2.1 Human-Robot Interaction

The idea of humans and robots working together has been discussed in a wide variety of application areas [12]. Different definitions for such interactions can be found in the literature.

Fong et al. [13] define human-robot interaction (HRI) as

"the study of the humans, robots, and the ways they influence each other." *[13, p. 11]*

Goodrich and Schultz [12] define HRI as follows:

"Human-Robot Interaction (HRI) is a field of study dedicated to understanding, designing, and evaluating robotic systems for use by or with humans." [12, p. 204]

Onnasch et al. [14] establish a taxonomy for HRI scenarios applicable to various application scenarios. One focus area of the use of robotics for the fulfillment of work tasks is of industry and assembly [15].

Previously, humans and robots had to perform different tasks in different locations throughout the factory; they were separated by high protective fences and without contact [15]. Traditional industrial robotic systems have limited adaptability, high installation costs, and require a large amount of space due to the need for costly and bulky fencing and other periphery safety equipment [16].

Nowadays, the modern marketplace requires shorter production schedules and higher levels of customization, which necessitates more adaptable and versatile assembly processes [16]. This is where collaborative robots have the potential to address a variety of issues that arise throughout the production and assembly workflow [16]. With such a collaboration of humans and robots, additional topics arise, like the allocation of tasks [17]. Therefore, we discuss collaborative robots and the determination of task allocation between humans and robots in the following section.

2.1.1 Cobots and Task Assignment

The term "cobot" is an acronym for collaborative robots that was introduced in 1995 by Colgate et al. [18]. They define a cobot as follows:

"A 'cobot' is a robotic device which manipulates objects in collaboration with a human operator." [18, p. 1]

Since then, much research has been conducted on this subject, which has led to the following definition by Schmidbauer [19]:

"A cobot is a compliant, reprogrammable, multipurpose robotic arm designed for a physical interaction with humans that can flexibly be used stationary or mobile." [19, p. 12]

Figure 4: Different collaborative robots from different manufacturers [20]

The ability of collaborative robots to interact directly with human workers sets them apart from more traditional types of industrial robots [21]. The workplace is shared by humans and robots simultaneously working on the same product [22]. Figure 5 outlines the different types of collaboration classified by Bauer et al. [22]. Furthermore, safety standards are in place to ensure that workers and cobots can work together safely [20].

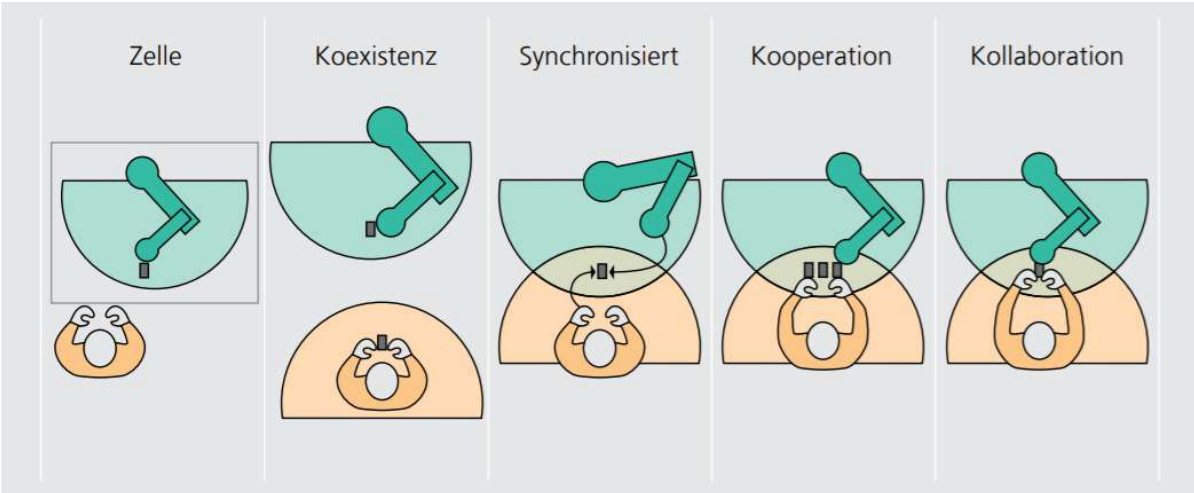

Figure 5: Definition of the level of collaboration by Bauer et al. [22]

One crucial aspect of cobot workplace planning in the manufacturing industry is the division of labor between humans and robots [17]. Challenger et al. [23] distinguish between three strategies:

- Leftover: Everything that is technically and financially possible is automated, while humans do the rest (that which is left over)
- Compensatory: Allocation is based on one entity's capabilities/advantages, with compensation for the other entity's disadvantages.

- Complementary: Functions and tasks are shared based on capabilities and additional criteria, while humans and machines work together as a team.

The leftover and compensatory approaches have the drawback of being static, rigid, and inflexible, necessitating significant implementation effort to apply changes [3]. They cause workers' monotony, increasing the likelihood of complacency, diminished situational awareness, and lack of a desire to learn [3].

The unfavorable consequences of static task allocation were primarily responsible for developing the complementary paradigm of task allocation [3]. Complementary task allocation is now commonly acknowledged as the most effective strategy for creating human-machine teams in terms of obtaining optimal performance [2]. It provides advantages such as increased flexibility, productivity gains, economic efficiency, and enhanced transparency [2]. In addition, the approach provides more learning opportunities for workers specifically compared to the static allocation of tasks [24].

Even though the strategy is frequently advocated for, it is rarely put into actual industry practice [2]. Therefore, a new approach called adaptive task sharing is introduced, which we discuss in the following subsection.

2.1.2 Adaptive Task Sharing

Adaptive task sharing (ATS) introduces shareable tasks, so-called shareables, which are feasible for humans and robots [3]. Figure 6 illustrates the difference between static task allocation and the newly introduced ATS method using shareable tasks, which can be executed by humans and robots and allocated adaptively [3]. Some tasks may only be feasible for humans or robots due to quality reasons, ergonomics, or the impossibility of automating some tasks [3].

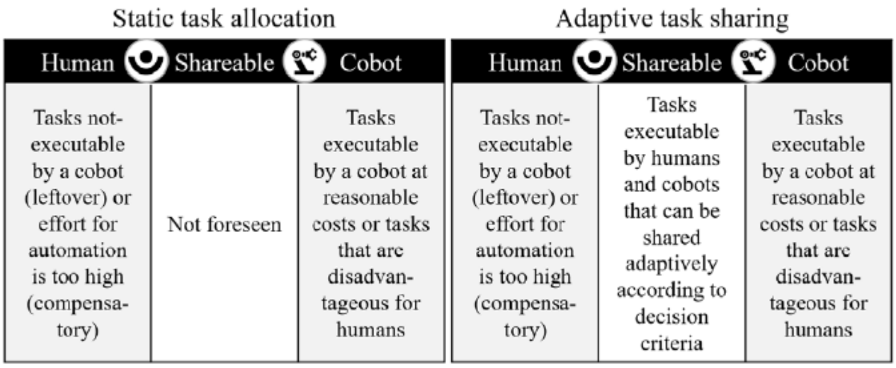

Figure 6: Static task allocation vs adaptive task sharing [3]

At first, the tasks are pre-assigned to either the robot or human based on robot feasibility and human suitability, while the remaining tasks are considered shareable [19]. Before starting a new process operation, the human can rearange shareables influenced by different decision criteria [2]. For instance, if the lot size is small, more tasks can be allocated to the human, and if the lot size is large, more tasks can be allocated to the robot [3].

Calculation of economic efficiency and ergonomic effects

Economic efficiency needs to be determined for varying order sizes regarding the needed time for the process, process costs, and ergonomic effects to visualize the effects of the task allocation [19].

The relevant variables needed for ATS economic efficiency calculation are as follows [19]:

Table 1: Variables for economic efficiency calculation [19]

Variable	Explanation	Unit	Symbol
Hourly costs human	An average hourly cost of human work should be taken if the real numbers are not available.	€	cH
Hourly costs robot	An average hourly costs of robot usage should be taken if the real numbers are not available.	€	cR
Execution time human	The durations of the execution are calculated or recorded.	s	tH
Execution time robot	The durations of the execution are calculated or recorded.	s	tR
Setup costs	If, in addition to the robot and standard gripper, other tools or fixtures are required, the costs must be estimated/calculated. Setup costs also include the programming/hand guiding time of the robot if the task is not already available as a reusable task.	€	c_{setup}
Repetition rate	The repetition rate must be known (e.g., lot size), calculated or estimated. The question to be answered is "How often will this task be needed?"	Constant	f
Setup costs/repetition rate	The setup costs must be set in relation to the repetition rate.	€	c_{setup}/f
Costs per part human	The cost of one part produced by human.	€	cPH
Costs per part robot	The cost of one part produced by robot.	€	cPR
Fastest, most time-efficient process	The sum of all optimal execution times, i.e., the shortest process execution time yielded by either a human or robot.	s	topt
Cheapest, most cost-efficient process	The sum of all optimal task execution costs, i.e., the most cost-efficient process execution costs yielded by either a human or robot .	€	copt
Lot size for which robotization is more cost efficient	The repetition rate with which robotization of this task is more cost efficient than a manual execution.	Constant	fopt

To calculate the costs per part for humans (c_{PH}), the hourly costs for humans (c_H) are multiplied by the execution time in seconds (t_H) and divided by 3,600 [19]. Human labor costs in Austria average €38.04 [25],per hour which can vary by industry and company [19]. The execution time is calculated using Methods Time Measurement - Universal Analyzing System (MTM-UAS) in seconds, resulting in the following equation [19]:

$$c_{PH} = \frac{c_H * t_H}{3600} \qquad \text{Equation 1}$$

In order to calculate the costs per part for robots (c_{PR}), the hourly costs for robots (c_R) are required, including the costs of purchasing, depreciation, and operating costs [19]. Furthermore, the execution time for robots (t_R) is needed, which can be assessed using either time stopping or MTM [26] for HRI [19].

To complete the equation for (c_{PR}), the costs associated with preparing the robot for a particular task, the so-called setup costs (c_{setup}), are divided by the repetition rate (f), which indicates the lot or order size [19]. This leads to the following final equation [19]:

$$c_{PR} = \frac{c_R * t_R}{3600} + \frac{c_{setup}}{f} \qquad \text{Equation 2}$$

The most time-efficient process is determined by adding all minimal execution times [19]:

$$t_{opt} = \sum_{i=1}^{n} \min\left(t_{H,i}, t_{R,i}\right) \qquad \text{Equation 3}$$

The most cost-efficient process is calculated by adding all minimal costs per part values [19]:

$$c_{opt} = \sum_{i=1}^{n} \min\left(c_{PH,i}, c_{PR,i}\right) \qquad \text{Equation 4}$$

Physical Ergonomics

To assess physical ergonomics, rapid upper limb assessment (RULA) [27] is a well-suited method for determining human suitability for assembly tasks [19]. The possible RULA scores range from 1 to 7, with recommended actions shown in Table 2 for each score.

Table 2: RULA scores and recommended actions [27]

Score	**Measure**
1–2	An acceptable state
3–4	A change should be made
5–6	Change needs to happen soon
7	Immediate changes are required.

Tasks with scores equal to or greater than 5 are recommended to be assigned to the robot [19]. Table 3 provides a more general explanation with the possibility of different thresholds.

Table 3: Decision criteria for human suitability evaluation [19]

Decision Criteria	Explanation	Unit	Symbol
Physical ergonomics	If the score is higher than or equal to the threshold, the task is assigned to the robot R; otherwise, it is a shared task B.	Score	E

Prototype (worker assistance system for enabling ATS)

For the purpose of enabling adaptive task sharing (ATS), a prototype was developed by Schmidbauer et al. [3]. It functions as a worker assistance system (WAS) with a user interface that illustrates the assembly process in business process modeling notation (BPMN) [2]. As a basis for the web-based UI, the open-source project BPMN.io [28] is used. The UI is divided into three areas, as seen in Figure 7. The first section is composed of a workspace and a tool palette to create the process using BPMN components [2]. In the second area, there is a parameter panel that allows the user to set various parameters, such as the commands that the robot needs to carry out [2]. The third component is a top bar with a play button to start the process [2].

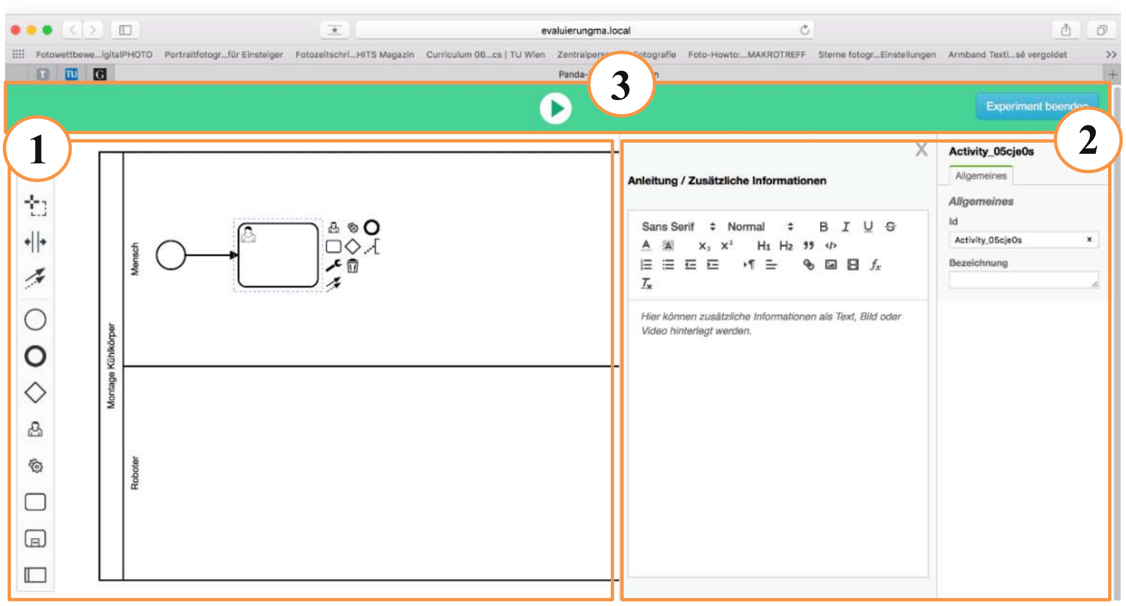

Figure 7: UI of the prototype composed of the three areas[2]

When the process is started, everything except the allocated tasks is hidden so as to focus attention solely on the workflow. Additionally, an animation is used to indicate which activity is currently being carried out [2].

For the execution of the created process, the prototype utilizes the Camunda BPMN Workflow Engine [29], which has the primary purpose to monitor and execute the processes that have been designed [4]. Furthermore, an external node JS task client is used to execute the BPMN engines service tasks [4]. This task client communicates with the Franka Emika Panda Cobot using the REST API in order to initiate the appropriate task [4].

Figure 8 provides a visual representation of the system architecture. Additional information regarding the prototype can be found in [6].

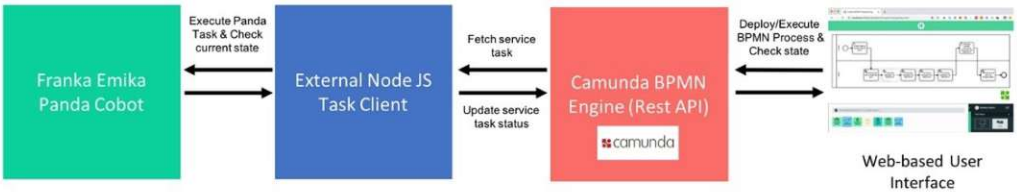

Figure 8: ATS prototype architecture [3]

2.2 UI/UX Design

To design a user-friendly user interface (UI), it is essential to clarify the term “user friendly”.

Essentially, user friendly is a synonym for usability [30]. Moreover, a user-friendly application should demonstrate a good user experience (UX) [31] or, more specifically, a sufficient user experience [32]. Frequently, there arises confusion because of the relation between usability and user experience [33]. Hassan et al. [33] argue that usability and user experience complement each other [33]. This underlines that both concepts are significantly important to create user-friendly user interfaces.

Therefore, the two terms “usability” and “user experience” are defined in more detail below.

2.2.1 Usability

In the literature, there are various definitions of usability. We look at a few definitions to better understand the concept and how to measure it.

Dumas and Redish [34] define usability as follows:

"People who use the product can do so quickly and easily to accomplish their own tasks." [34, p. 4]

They identify four points on which the definition relies [34]:

Usability means focusing on users.
The objective is to know and understand the prospective target users and groups, not merely to realize one's thoughts and aspirations.

People use products to be productive.
The interface helps users achieve personal performance objectives. Therefore, it is important to be conversant with user expectations, their work, and the activities the system automates, modifies, or improves.

Users are busy people trying to accomplish tasks.
Productivity gains are a key metric by which consumers evaluate a product's usability. Questions should be asked, for example "How long does it take the typical user to get what they want?"

Users decide when a product is easy to use.
When it comes to concerns about the ease of use of a certain product, the emphasis should be directed at the consumers rather than the product designers or developers.

Nielsen [35] states the following about usability:

"Usability is a quality attribute that assesses how easy user interfaces are to use." [35]

Furthermore, he defines usability by five quality components as follows [35]:

Learnability: How easy are basic tasks for first-time users?

Efficiency: How quickly are users able to complete activities when they are familiar with the design?

Memorability: How simple is it for users to return to their prior level after not utilizing the system?

Errors: How often do users make errors?; How serious are those errors?; and How simple is it for users to get back on track after making an error?

Satisfaction: How enjoyable is it to make use of the design?

Another definition is given by the ISO standard 9241-11 [36], which defines usability as the:

"extent to which a system, product or service can be used by specified users to achieve specified goals with effectiveness, efficiency and satisfaction in a specified context of use." [36, p. 2]

According to the ISO standard, the three measurable elements for usability are as follows [36]:

Effectiveness: the user's accuracy and completeness in achieving specific goals.

Efficiency: the level of resources that were spent with respect to the outcomes that were accomplished, where resources can be time, human effort, costs, and materials.

Satisfaction: the subjective reaction a person has as a direct result of their interaction with a system.

2.2.2 User Experience

In literature, there are several definitions for the term "user experience."

The Nielsen Norman Group [37] defines it as follows:

"'User experience' encompasses all aspects of the end-user's interaction with the company, its services, and its products." [37]

Another definition is given by ISO standard 9241-210 [38], which defines user experience as the

> *"user's perceptions and responses that result from the use and/or anticipated use of a system, product or service." [38, p. 4]*

However, these definitions of user experience are very broad. Moreover, user experience is assumed to be an all-embracing concept that incorporates a variety of reactions, including cognitive, emotional, and even physical responses, in relation to the actual application of a product [39]. As a consequence of this, the user experience is difficult to measure.

User experience can also be defined in a different way, namely as a set of distinct quality criteria that include both traditional usability criteria, such as efficiency and controllability, and hedonic quality criteria, such as stimulation, fun-of-use, novelty, emotions, or aesthetics [39]. This offers a different perspective on the concept of user experience. The distribution into quality criteria has the advantage that it separates the broad concept of user experience into several straightforward criteria, each of which describes a distinct and relatively well-defined aspect of user experience that can be measured separately [39]. There are six different measurement scales used, as can be seen in Table 4 and Figure 9.

Table 4: Six dimensions for user experience [39]

Dimension	**Explanation**
Attractiveness	The product's general impression: How do users like the system?
Perspicuity	Is it easy to get used to the product?
Efficiency	Can users accomplish their assignment with the product without exerting more effort than necessary?
Dependability	Do the users have the feeling of being in control?
Stimulation	Is using the product engaging and motivating?
Novelty	Does the product excite and inspire the user to utilize it?

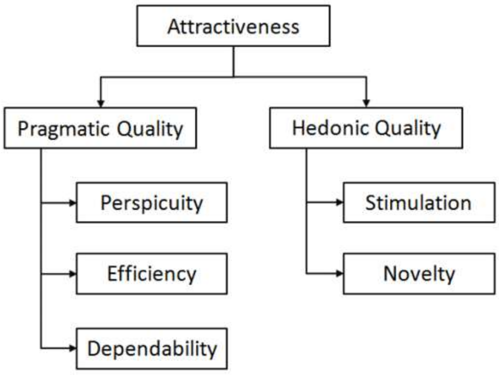

Figure 9: Scale structure of the dimensions [39]

2.3 Learning Opportunities

According to recent research, having a trained labor force is critical for remaining globally competitive and stimulating economic development [40]. To remain competitive in today's demanding and ever-evolving environment, it is essential for employees, as well as organizations, to continually update their skill sets [40]. Overall, the demand for new skills is considered very high. According to research by Gartner Inc., 58% of the workforce need new skills to do their jobs successfully [41].

Learning is of the utmost importance in a working environment increasingly dominated by technology [19]. Particularly for workers in the industrial sector, it is crucial to develop new skills and capabilities [42]. As computers and software take over previously performed tasks from them, humans need to acquire new skills and become proficient in a broader range of activities [19].

However, due to machines taking over routine tasks and diverting human attention to tasks that are not routine, there are fewer opportunities for human operators to learn routine processes [24].

2.3.1 Learning Theory

Since we are now aware of how crucial learning is in today's world, we take a closer look at learning in industrial settings and how it can be modeled.

The process of acquiring specific qualifications via intentional or unintentional means is known as learning [43]. It is referred to as work-based learning when the real learning occurs in the workplace and during the work process [44].

During the course of the process, a basic pattern of these qualifications is developed, which can then be improved through a practice phase involving regular or sporadic repetition [45].

Due to the fact that the majority of production job activities need sensorimotor abilities, it is necessary to acquire these skills in a manner appropriate to each activity [46]. This indicates that the worker requires a specific period of time before he or she is capable of carrying out the new job in a manner that fulfills the standards for both the quantity and quality of the work [47].

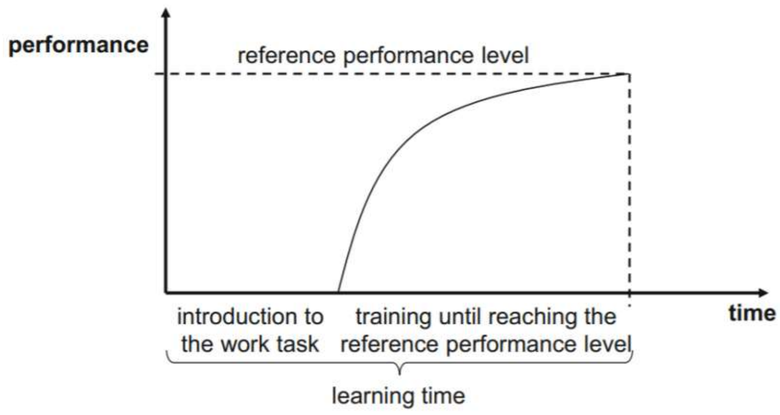

Figure 10: Schematic representation of learning time [47]

The amount of time required to learn how to carry out a new task and practice it repeatedly until it can be performed at a level that is considered to be proficient is known as the learning time [47]. The learning time is composed of two phases. The introduction to the work task constitutes the first phase [47]. The second phase consist of training until reaching the reference performance level, as shown in Figure 10 [47]. The performance improvements that occur during the second phase as a direct result of the development of skills and knowledge can be very effectively described by means of learning curves [47].

2.3.2 Learning Curve Models

Learning curves typically show a decrease in the time required for an activity's execution as the number of executions increases [45]. Wright [48] is known for inventing the first learning curve for use in an industrial setting [45]. The basic model of a learning curve proposed by Wright [48] suggests that performance can indefinitely improve, making the model unrealistic and unsuitable for the majority of applications [47]. As a result, numerous further models have been developed, such as the one from De Jong [49], who extended Wrights power function model by a limiting value [47]. The literature presents other varieties of learning curve models [50][51][52], that all follow the same curve progression [53]. One limitation of these models is that the mere description of learning times makes them unsuitable for serious production planning [47, 54]. Forecasting of the learning time is an important part as well [47]. However, previous prediction methods were too inaccurate for industrial applications [54]. In response to these limitations, Jeske [55, 56] devised a new prediction method for sensorimotor work tasks [47].

2.3.3 Prediction of Learning Time

To predict the learning time in production planning, Jeske [56] adapted De Jong's [49] learning curve model. Different parameters are taken into account, including information about the working person, the task difficulty, and the work instructions [56]. Figure 11 depicts the model of the learning curve along with its parameters.

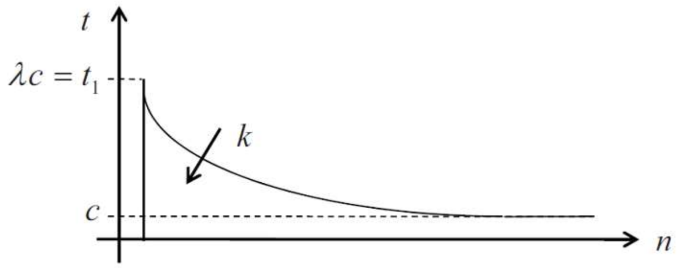

Figure 11: Representation of the learning curve and the model parameters to be determined [56]

The initial execution time, denoted by t_1, is calculated by multiplying the limiting value for the learning progress c by the model parameter λ [56]:

$$t_1 = c * \lambda \qquad \text{Equation 5}$$

The limiting value c needs to be determined using a suitable approach, which Jeske recommends to be the methods time measurement - universal analyzing system (MTM-UAS) [56]. MTM-UAS is widely used in series production planning to record times and tasks of processes [57].

The time required for n executions is calculated as follows [56]:

$$t_n = c + (\lambda - 1)cn^{-k}n^{-ke^{\frac{k}{2}(n-1)}} \qquad \text{Equation 6}$$

For the model parameter λ, Jeske determined a regression model with a constant and four influencing variables, which resulted in the following equation [56]:

$$\lambda = 2{,}256 + 0{,}978\sqrt{{H_{UAS}}^2 + {H_B}^2} - 0{,}755E_{Assembly} - 0{,}45D + 0{,}87G \qquad \text{Equation 7}$$

H_{UAS} is defined as the entropy of the elements, which describes the required movements and characterizes them with respect to their information content [56]. Furthermore, H_B defines the physical parts of the work task and characterizes them concerning their information content [55]. Together, they represent the difficulty of the work task [56].

The parameter $E_{Assembly}$ represents the worker's experience with assembly, which must be surveyed subjectively for every worker on a Likert scale (none/1, little/2, average/3, a lot/4). The parameter D describes the type of task description on a Likert scale (text based/1, text and picture based/2, picture based/3, video based/4), and the parameter G represents the gender of the worker on a Likert scale (men/1, women/2) [56].

The parameter k, which stands for the learning velocity, is defined in the following equation [56]:

$$k = 0{,}141 + 0{,}073\lambda - 0{,}008FF_1 + 0{,}006FF_6 + 0{,}013A \qquad \text{Equation 8}$$

The learning velocity k is accelerated by the model parameter λ and the age-representing parameter A [56].

The parameters FF_1 and FF_6 represent the fine motor skills of the worker and also affect the learning velocity k [56]. They are characterized by Fleishman factors [58] in terms of T values, which can vary from 0 to 100 [56]. In the range of 40 to 60, performance is considered average, with higher values indicating better performance [56]. FF_1 is defined as aiming, which has an inhibiting effect, and FF_6 stands for the wrist-finger speed, which has positive effects on learning velocity [56]. This can be described as speed-accuracy tradeoff due to the increased time required for better aiming [56].

Following the calculation of all required variables, the number of repetitions required to reach the reference performance must be estimated in order to calculate the learning time [56].

To approximate the execution times needed to reach the reference performance, the following equation is used [56]:

$$N = e^{\left(\frac{1}{-k} ln \frac{\xi - 1}{\lambda - 1}\right)} \qquad \text{Equation 9}$$

The parameter ξ represents the level of acceptance and is described as the multiple of the limitation value c $for\ \xi > 1\ (f.e. 1{,}05)$ [56].

Finally, the learning time can be calculated. Every execution time form t_1 to t_n is summed up; Therefore, possible break times are not taken into account [56]:

$$\sum_{n=1}^{N} t_n = \sum_{n=1}^{N} c + (\lambda c - c) n^{-k\left(1 + e^{\left(\frac{k}{2}(1-n)\right)}\right)}$$
$$= cN + c(\lambda - 1) \sum_{n=1}^{N} n^{-k\left(1 + e^{\left(\frac{k}{2}(1-n)\right)}\right)} \qquad \text{Equation 10}$$

It is important to keep in mind that this approximation results in a 10% underestimation of the learning time [56].

3 State of the Art

This chapter discusses the current state of the art of task allocation between a human and a robot while considering learning.

To achieve this, a systematic literature review (SLR) is conducted to gather current research developments. The goal is to provide an overview of the most recent trends and challenges as well as relevant insights for the following sections in summarizing the findings.

3.1 Systematic Literature Review

A systematic literature review is used to identify existing relevant publications focusing on task allocation between a human and robot while considering learning. The guidelines by Kitchenham [9] serve as a basis for performing the review. The relevant research question is "**Which approach is suitable to realize adaptive task sharing between a human and robot considering learning opportunities regarding learning new tasks?"**. The following databases were used:

- Scopus
- IEEE Xplore
- Google Scholar

The review is divided into two steps. In the first step, we derive keywords from the research question for the domain of task allocation in human-robot collaboration. In the next step, we try to add the domain of learning to narrow down the results to the relevant topics.

The search keywords, derived from the research question and depicted in Table 5, are relevant for the domain of task allocation between a human and robot. These keywords can slightly deviate for different databases as the results depend on their scientific focus.

Table 5: Keywords for task allocation between a human and robot

Keywords
Human
Robot
Task allocation

Furthermore, the exclusion terms listed in Table 6 are used to refine the search results and find more relevant articles.

Table 6: Exclusion terms for task allocation between human and robot

Exclusion terms
Mobile robots
Multi robots
Rescue
Unmanned
Swarm
Humanoid robot

The keywords and exclusion terms resulted in the research queries presented in Table 7.

Table 7: Search queries for task allocation between a human and robot

Search query	Database	Results
("human" AND "collaborative robot") AND "task allocation" AND -"Multi-robot" AND -"swarm" AND -"unmanned" AND -"rescue" AND -"mobile robot" AND -"multi robots" AND -"humanoid robot"	Google Scholar	286
TITLE-ABS-KEY(human AND robot) AND TITLE-ABS-KEY("task allocation") AND NOT TITLE-ABS-KEY("Multi-robot" OR "swarm" OR "unmanned" OR "rescue" OR "mobile robot" OR "multi robots" OR "humanoid robot")	Scopus	182
("All Metadata":"task allocation) AND ("All Metadata":"robot") AND ("All Metadata":"human") NOT ("All Metadata":"mobile robot") NOT ("All Metadata":"humanoid robot") NOT ("All Metadata":"multi robots") NOT ("All Metadata":Swarm) NOT ("All Metadata":unmanned)	IEEE Xplore	94

The search of the three databases resulted in 562 hits. The results were reviewed for duplicates. In total, 76 duplicates were detected, narrowing the results to 486 articles.

After obtaining the results for task allocation for a human and robot, we tried to add the domain of learning to the search queries in a second step. As learning progress over time can be described by learning curves (see Section 2.3.2), we focus on literature that addresses the topic of learning curves in the context of production. As Pena et al. [59] recently conducted a systematic literature review focusing on the usage of learning curves in assembly operations, we tried to use the keywords defined in their study. As a result, the following keywords are defined:

Table 8: Keywords for learning curves [59]

Keywords
Learning curves
Human factors
Throughput prediction
Standard times
Productivity

Pena et al. [59] uses these keywords to construct the following query:

Table 9: Query string for learning curves in assembly operations [59]

Query string
(“standard times” AND “learning curve”) OR (“throughput prediction” AND “variability”) OR (“productivity” AND “learning curve” AND “prediction”) OR (“productivity” AND “human factor” AND “prediction”) OR (“productivity” AND “labor intensive” AND “prediction”)

When adding this query string to the search queries listed in Table 7, neither database returned a result.

In order to avoid overlooking any publications, we used a keyword that does not limit the search as much as the query in Table 9. The added keyword we used for this search is “learning.” The search of Google Scholar was further refined by excluding “reinforcement learning” and “deep learning” to narrow down the results.

The final search queries are depicted in Table 10.

Table 10: Search queries including the keyword "learning"

Search query	Database	Results
("human" AND "collaborative robot") AND “task allocation" AND ("learning") AND -"Multi-robot" AND -"swarm" AND -"unmanned" AND -"rescue" AND -"mobile robot" AND -"multi robots" AND -"humanoid robot" AND -"reinforcement learning" AND -"deep learning	Google Scholar	124
TITLE-ABS-KEY (human AND robot) AND TITLE-ABS-KEY ("task allocation") AND TITLE-ABS-KEY ("learning") AND NOT TITLE-ABS-KEY ("Multi-robot" OR "swarm" OR "unmanned" OR "rescue" OR "mobile robot" OR "multi robots" OR "humanoid robot")	Scopus	17
(("All Metadata":"task allocation") AND ("All Metadata":"robot" AND "human") AND ("All Metadata":“learning”) NOT ("All Metadata":"mobile robot") NOT ("All Metadata":"humanoid robot") NOT ("All Metadata":"multi robots") NOT ("All Metadata":Swarm) NOT ("All Metadata":unmanned))	IEEE Xplore	6

The search of the three databases resulted in 147 hits. The results were again reviewed for duplicates, and 14 duplicates were detected, which narrowed the results down to 133 articles.

Kitchenham [8] advises using inclusion/exclusion criteria. Therefore, only full-text publications and articles accessible by VPN of the TU Wien were considered.

After scanning the titles and abstracts of the resulting hits, no relevant articles for this thesis within the learning domain were determined. It can be assumed that this topic is underrepresented in literature.

3.1.1 Summary of the State of the Art

In the summary of the state of the art, we use the results from the domain of task allocation between a human and robot to briefly provide an overview of recent advancements and a state-of-the-art reference model for task allocation. Furthermore, since there is no previous research combining learning within task allocation between a human and robot, we look at learning curves in the context of production management to give an overview and derive findings for this thesis.

Task allocation between a human and robot

In their review, Petzoldt et al. [60] propose approaches for task allocation in a human-cobot collaborative assembly process. They distinguish the allocation types into static and dynamic task allocation. Static or offline methods are defined as the allocation of tasks before executing the assembly process, where no allocation changes can be made during the execution, whereas with dynamic or online approaches, task allocation can be changed while the process is being executed, thus benefiting from the possibility of adapting to current situations during the execution. [60]

As shown in Figure 12, Petzoldt et al. distinguish the two categories of static and dynamic further into four subcategories as follows [60]:

- Task allocation based on suitability assessment
- Simulation-supported task allocation
- Reactive and ad-hoc task allocation
- Proactive task allocation

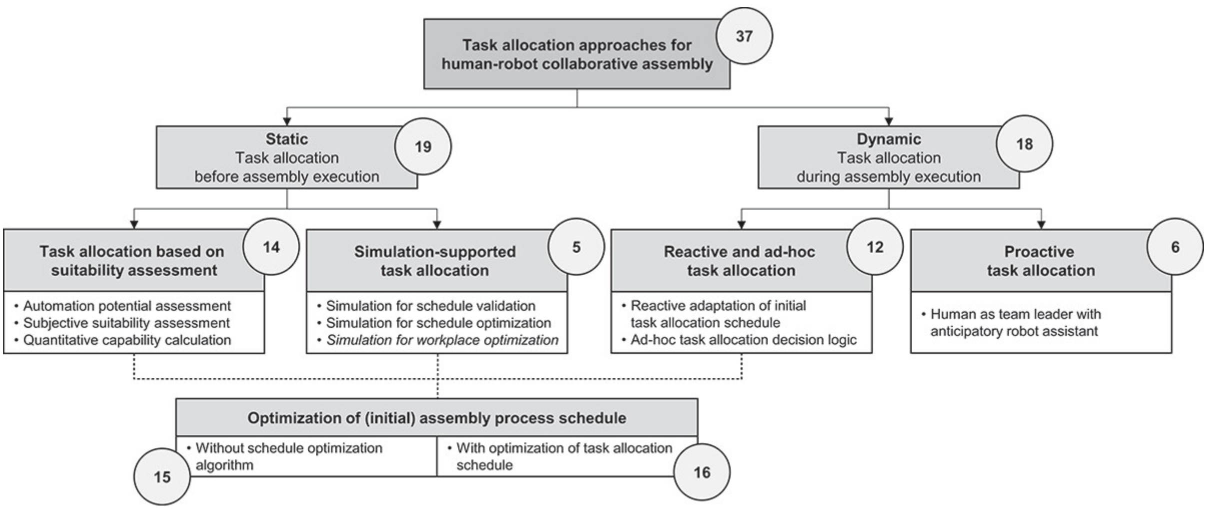

Figure 12: Categorization of task allocation approaches [60]

Task allocation based on suitability assessment

This subcategory defines a method of assigning tasks by first checking the suitability of humans and robots for specific tasks within the assembly process [60].

An example would be Schmidbauer et al.'s [3] adaptive task sharing method. A detailed description of this approach can be found in Section 2.1.2.

Simulation-supported task allocation

Petzold et al. [60] argue that approaches in this subcategory are also accounted for in static task allocation, but they are enhanced by a simulation to validate or optimize the assembly schedule [60]. An example of allocation optimization using a simulation tool can be found in the work of Bänziger et al. [61].

Furthermore, this approach makes it possible to compare different task allocation schedules before execution [60]. Tsarouchi et al. [62] show how a comparison of alternatives is possible [62].

Reactive and ad-hoc task allocation

In the next subcategory, Petzold et al. [60] describe reactive and ad-hoc task allocation, which share a central planning instance that allocates tasks to the human or robot. The authors distinguish between two sorts of approaches. The first starts with a plan that then can be subject to dynamic modifications in response to unforeseen disruptions. This may include revising task assignments (see Pupa et al. [63]) or initiating a replanning process for the remaining work (see Johannsmeier & Haddadin [64]). [60]

Another example is presented by Petzold et al. [65], which is described in more detail below.

The second approach is based on ad hoc decision logic performed dynamically during work allocation, whereas the decision of whether a task should be given to a human or robot can be based on characteristics such as task classifications or human fatigue [60]. For example, Makrini et al. [66] use an ad-hoc approach based on capability and ergonomics.

Proactive task allocation

Petzold et al. [60] describe that humans take charge of the assembly procedure in proactive approaches. The system anticipates and determines which tasks the robot can do in order to assist the human worker in a proactive manner. [60]

In this approach, the human selects and performs preferred tasks, while the robot proactively selects preparatory tasks that can be done in parallel to assist the human in reducing the overall process completion time [60]. An example is described by Cramer et al. [67].

After distinguishing between the approaches, Petzoldt et al. [60] propose a reference model for task allocation covering all subcategories except proactive task allocation due to its fundamentally different approach:

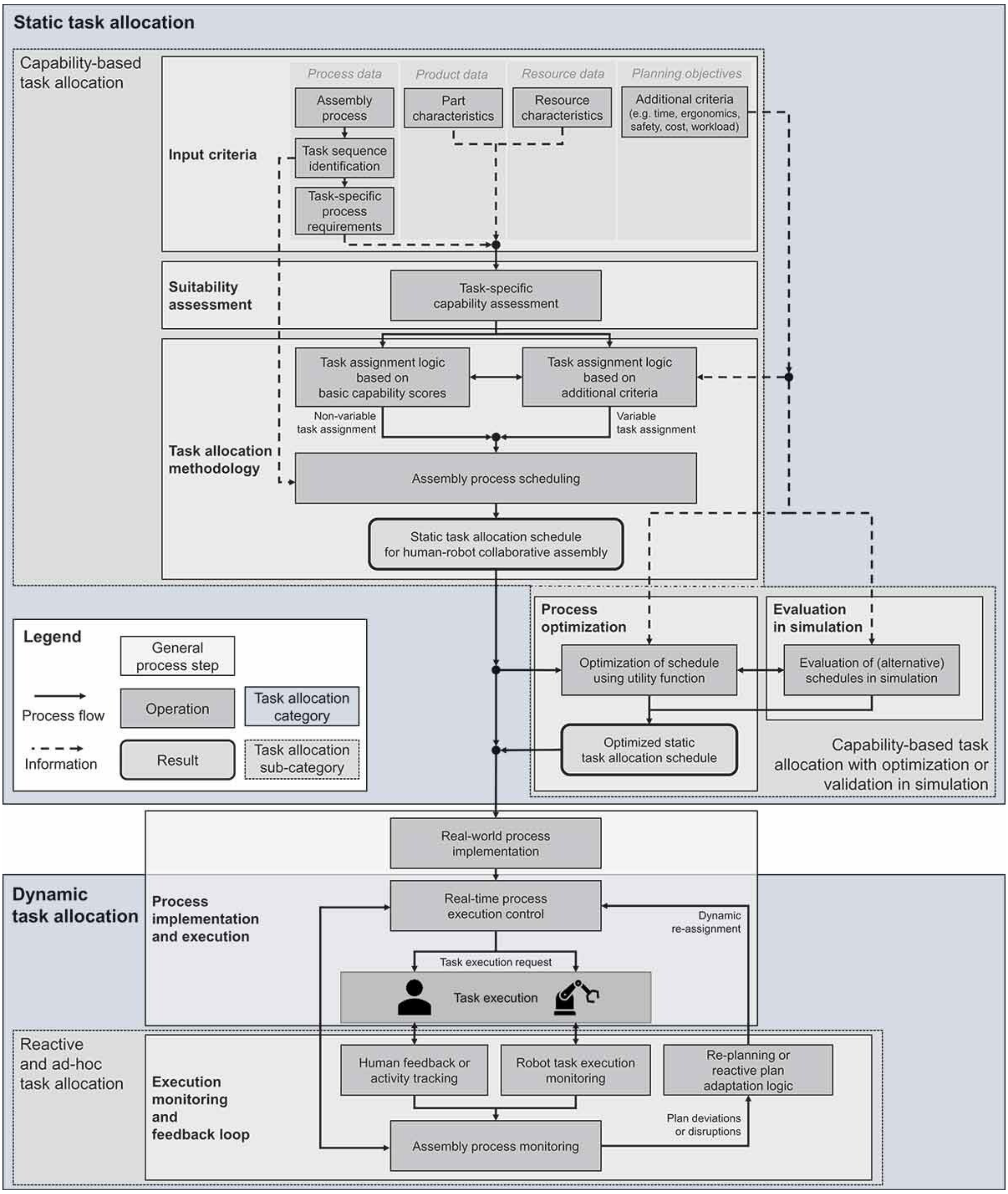

Figure 13: Reference model for task allocation [60]

The reference model consists of up to seven steps, depending on the allocation type [60]:

1. Input criteria provision: collection/recording of a set of criteria; an interface for accessing criteria needed by the task allocation method.
2. Suitability assessment: uses information about processes, products, and available resources to determine if and how well the agents can perform each task.
3. Task allocation methodology: first, assign static tasks using the capability assessment; if needed, rate of the meaningfulness with additional criteria; create a process schedule
4. Process optimization: optimize the schedule priory created using additional criteria.
5. Process evaluation in simulation: can be used for feasibility checks, to create alternative schedules, or to calculate parameters for process optimization.
6. Process implementation and execution: implementation and execution of process schedule at assembly system.
7. Execution monitoring and feedback loop: observation of progress and checking for deviations from the plan.

Applicable findings

Especially the article "Implementation and Evaluation of Dynamic Task Allocation for Human-Robot Collaboration in Assembly" by Petzold et al. [65] is very relevant to this thesis. Therefore, the approach and UI are covered in greater detail.

This article examines the effects of static versus dynamic task allocation in assembly processes. The researchers propose a system that minimizes planning effort by automatically assigning tasks based on task classification and assembly priority during execution. This system combines block-based programming with dynamic task assignment. The evaluation of the system is based on a user study that compares static and dynamic task allocation. [65]

In Figure 14, the UI of the worker assistance system and the process flow editor for the creation of the assembly process are presented [65].

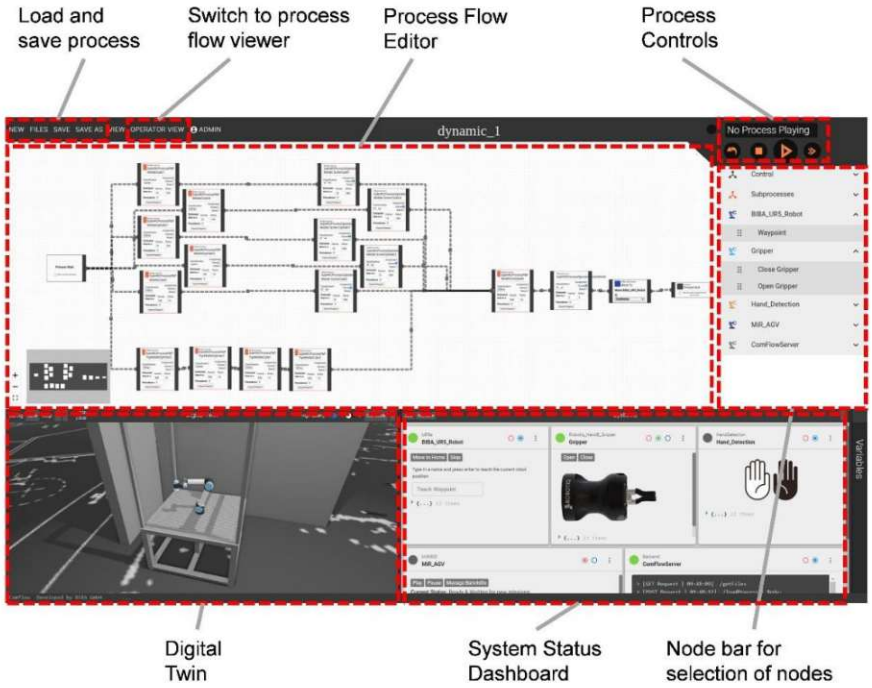

Figure 14: Process flow editor for creation of the assembly process [65]

A switch to the process flow viewer is necessary to start the process [65]. The UI is depicted in Figure 15. It has an integrated worker instruction system for the execution of the assembly process. A relevant finding for this thesis is the usage of a dashboard in this system.

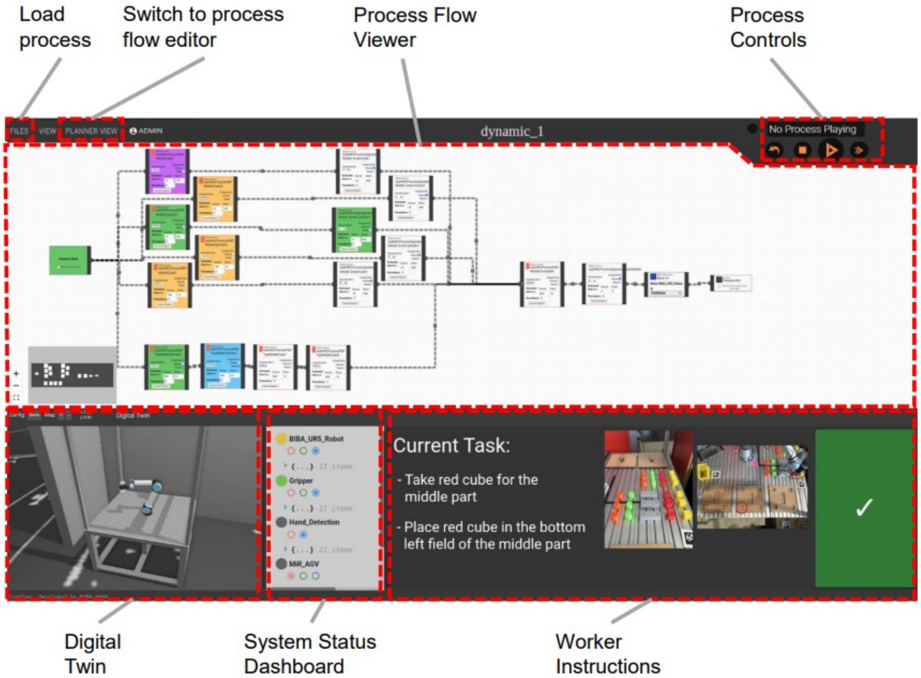

Figure 15: Process flow viewer for execution of the assembly process [65]

Learning curves in production management

As there is no previous research literature combining learning with task allocation between a human and robot, we look at learning curves in the context of production management.

We look at two state-of-the-art reviews: One focuses on the general overview of learning curves in production and operations management [68] and the other on learning curve models and their estimation methods for manual operations and processes [59].

The first review, published by Glock et al. [68], takes 457 articles into account. They highlight the rising trend of publications in this area, as shown in Figure 16.

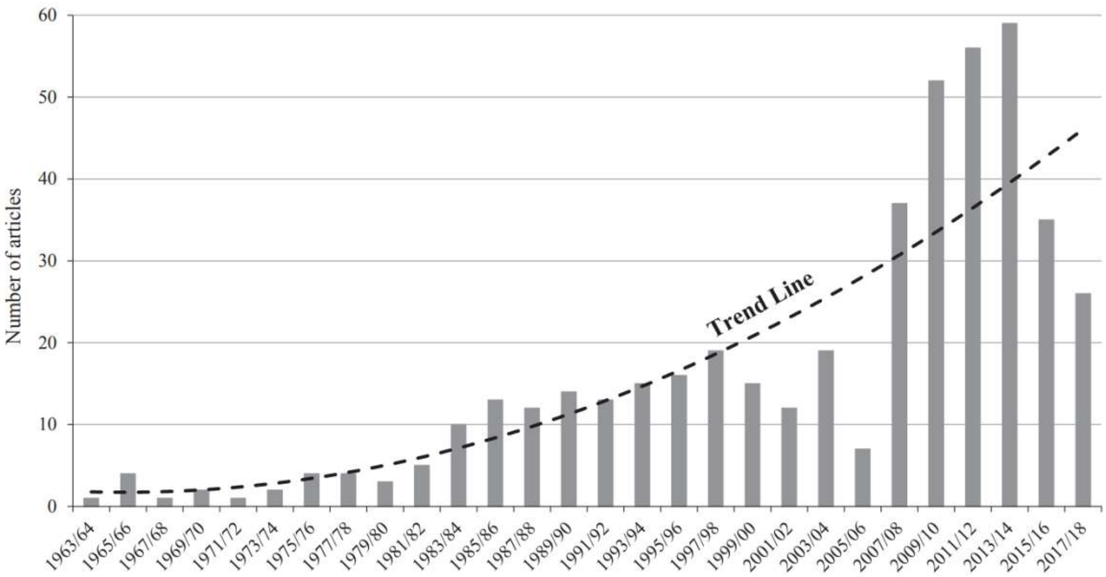

Figure 16: Trend line of publications in the area of learning curves in production and operations management [68]

Glock et al. divide the application of learning curves in production and operations management into the following areas [68]:

- Production management
- Inventory management, supply chain management
- Sustainability (e.g., returns/waste, green, emissions)
- Information technology and e-business
- Quality management (e.g., defects, inspections)
- Logistics
- Product management

An interesting finding was that the most common type of learning curve was the log-linear model based on the Wright learning curve, and the main area of use was production management; this focuses on the application of learning curves to different production-related subjects, including capacity management, assembly line planning, joint economic lot-size (JELS) models, and economic production quantity (EPQ) models. Moreover, the authors highlight the shortcomings of the Wright learning curve and emphasize conducting further research in industry-specific learning. [68]

In addition, Peña et al. [59] provide a summary of recent developments in the field of learning curve models for manually operated processes and operations [59].

In the review, five articles are highlighted. First, Tilindis and Kleiza [69] concentrate on the parameter estimation of two learning curve models (Wright [48] and Crawford [70]) with limited production data. The data from manual harness assembly in the automotive industry is used for validation. A comparison of the two models demonstrates that Wright's learning curve model provides a better fit than Crawford's approach. [59]

The second article, by Tamás and Koltai [71], delves into the potential of the traditional learning curve theory in industrial and service settings. This article explores several models of learning curves and shows how they can be applied to areas such as economic manufacturing quantity, break-even analysis, and assembly line balancing. The findings indicate that considering the learning effect in such domains can yield valuable insights at operational and strategic levels. Especially in the area of Industry 4.0 and Big Data, the article highlights the significance of incorporating the learning curve theory into decision-making processes. [59]

The third article, by Gao et al. [72], proposes a machine learning technique to forecast the shape of a surgeon's learning curve. They look at data on how many attempts it takes surgical trainees to reach a certain level of proficiency and then use a supervised machine learning model to extrapolate the number of tries and the result of their performance. Using data from the first 10 tries at surgical tasks, the researchers could accurately predict the learning curve's characteristics [59]. Peña et al. [59] suggest that even if this study is not directly related to manufacturing, the unique technique employed by Gao et al. [72] to estimate learning curves for manual surgical operations is easily transferable to manual assembly activities on a production line [59].

The fourth article, by Hogan et al. [73], discusses the limitations of conventional learning curve theory models from Wright [48] and Crawford [70] regarding the models having a constant learning rate (see Section 2.3.2) and introduces Boone's [74] learning curve, a model that accounts for decreasing learning rates as more units are produced [59].

The fifth article, by Di Luozzo et al. [75], investigates the effects of human performance, process configuration, and technical aspects on the early stages of implementing new automated or semi-automated production methods. They use Wright's learning curve model to examine how much time technicians spend learning new maintenance procedures [59].

The current state of the literature lacks discussion on the topic of learning and learning curves in the domain of task allocation between a human and robot. However, it is evident from the findings that learning and workforce development are crucial factors in the manufacturing industry, with a wide variety of application areas for learning curves in production. Another crucial observation is the limitation of classic learning curve models due to constant learning rates.

In the upcoming chapter, we explore the implementation of the learning curve model discussed in Sections 2.3.2 and 2.3.3, which does not suffer from the limitations described above.

4 Implementation

In this chapter, we look at the implementation phase. Now that we know the theoretical foundations and the current state of the art, we use this knowledge to meet the research questions. As seen in Chapters 2 and 3, a prototype enabling ATS by Schmidbauer et al. [3] is already available and serves as a basis for the implementation. At first, we derive requirements from the research questions and define respective design principles. After that, three implementation iterations are performed.

In the **first iteration,** the existing software architectures of the prototype are analyzed to get an overview of the prototype's code. After that, we attempt to meet the requirements in an iterative implementation method, including feedback loops, with the assistance of this thesis.

In the **second iteration,** a heuristic evaluation is performed by UI/Robotics experts to detect usability issues based on heuristic guidelines.

In the **third iteration,** the identified guideline violations by UI and robotics experts are resolved.

After the three iterations, the final artifact is tested and demonstrated with a real-world process.

Requirements and design parameters

In the first step, the requirements are defined. We look at the research questions, which we decompose to derive the requirements and define the design parameters:

> RQ2: What is a suitable design for a worker assistance system to improve decision support for adaptive task sharing and enable workplace learning?
>
> RQ3: What is a user-friendly way to implement these approaches?

In RQ2, the term "suitable design" refers to how improvement of decision support for a worker using the prototype with an ATS method can be reached and, secondly, how to include learning into the system. The term "user-friendly" was defined in Section 2.2. Based on this knowledge, the following requirements and design parameters are derived:

For the artifact specification, we chose design parameters based on the corresponding requirements, shown in Table 11.

Table 11: Requirements and design parameters

Requirements:	**Design Parameters:**	**No**
The prototype can assist in decision-making by pointing out the impacts when changing the task allocation.	Parameterization at the task level	1
	Dashboard, which visualizes the impact of task allocation	2
	Highlighting the cheapest/fastest process	3
The prototype enables learning in the workplace	Utilize the learning model for estimating learning times, and display the learning curve model	4
The prototype has a user-friendly design.	Implement the prototype with respect to usability and user experience (see UI/UX Section 2.2), i.e., simple design, validation with error messages, etc.	5

4.1 First Iteration

The prototype introduced in Section 2.1.2 serves as the starting point, and in Figure 17, the initial state is illustrated. The software architecture is analyzed to understand the existing implementation and determine where and how to initiate the changes.

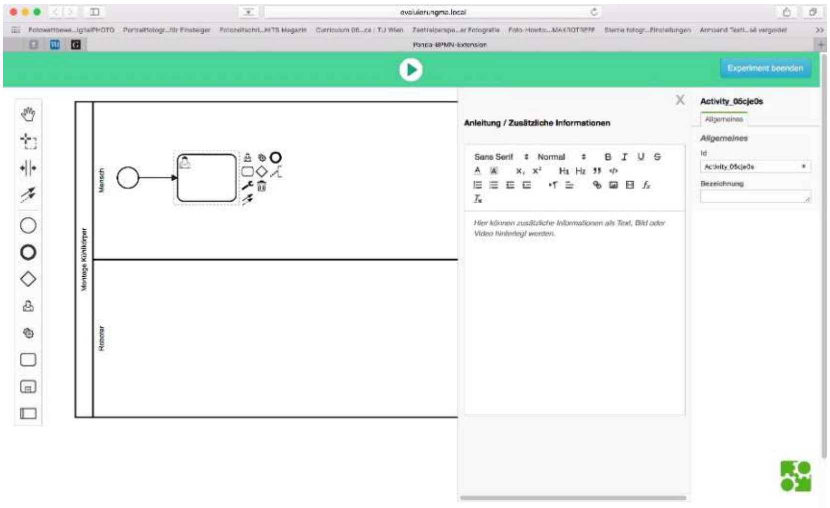

Figure 17: Initial state of the prototype[6]

To meet the previously determined requirements and design parameters, we use an iterative process for the implementation based on biweekly feedback cycles with the assistants of this thesis.

4.1.1 Parameterization

At first, we added parameters at the task level so that different values could be stored for each task (design parameter no. 1 - DP 1). Since the properties panel extension of bpmn.io was already used in the initial setup, it was expanded with additional input fields.

The properties panel appears when a task is selected, allowing the user to add or change the parameters. It is separated into distinct groups for simpler comprehension to guarantee a good user experience (DP 5). Table 12 provides a summary of all added groups and parameters.

Table 12: Groups and parameters of the properties panel

Group	Parameter	Description
General	Pre-allocation of task (feasibility)	Each task's feasibility can be set to worker, robot, or worker and robot. Implemented as a drop-down list; by default, worker and robot is selected.
Robot task parameters	Execution time robot	The duration of the robot execution. Value needs to be entered in seconds as a numeric value.
	Lot size	The lot size specifies how many identical components are produced [19]. Value needs to be entered in whole numbers.
	Set up time	Describes how long it takes a human to prepare the robot system[19]. Value needs to be entered in seconds as a numeric value.
Worker task parameters	Execution time worker	The duration of the worker execution. Value needs to be entered in seconds as a numeric value.
	Physical ergonomics	This can be assessed by the RULA score (Section 2.1.2.). Value needs to be 1, 2, 3, 4, 5, 6, or 7.
Learning parameters	Entropy of components	See Section 2.3.3. Value needs to be numeric.
	Entropy of the elements according to MTM-UAS	See chapter 2.3.3. Value needs to be numeric.

In order to give the user more guidance on how to enter the values and prevent incorrect inputs in advance, some of the input fields are provided with hints beneath the fields, as shown in Figure 18 and Figure 19.

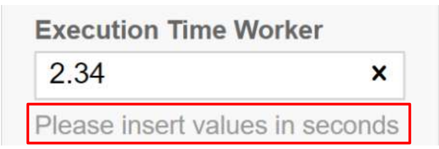

Figure 18: Additional information for input field "Execution Time Worker" (own screenshot)

Figure 19: Additional information for input field "Set up Time" (own screenshot)

To get an overview of the changes made, the old and new properties panels are compared.

The old properties panel is shown in Figure 20 along with its three parameters listed in Table 13:

Table 13: Initial parameters of the properties panel

Group	Parameter	Description
General	ID	Automatically generated ID
	Name	The name of the task. This is displayed directly for the task in the BPMN process.
Details	Name of the robot command	(Only for robot tasks) The robot command to be executed

The new properties panel with the added parameters from Table 12 is shown in Figure 21.

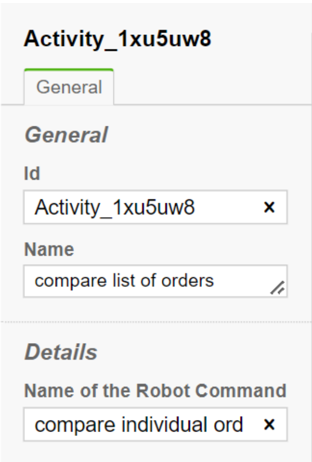

Figure 20: Old parameter panel (own screenshot)

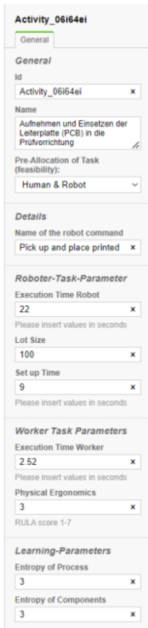

Figure 21: New parameter panel (own screenshot)

In the next step, a validation system for the input values of the parameters from Table 12 is implemented. If an input error occurs, a message appears under the input field, and the field is highlighted in red. This serves the purpose of providing the user with a clear error message showing them how to resolve the problem (DP 5).

In Table 14, all validation criteria are listed:

Table 14: Validation criteria for parameters

No	Parameter	Validation criteria
1	For all open input parameters	Input must be numeric.

2	For all parameters with whole number input	Check if the input is a whole number.
3	For all parameters with decimal input	Check if the decimal separator is ".".
4	Physical ergonomics - RULA score	RULA score must be one of the following values: 1, 2, 3, 4, 5, 6, or 7.

Figure 22Figure 23, Figure 24, andFigure 25 display a visual representation of the error messages resulting from the validation of the input fields.

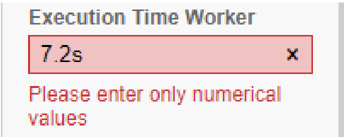

Figure 22: Input field with error #1 (own screenshot)

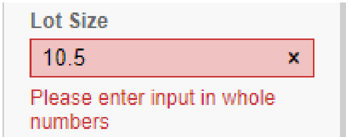

Figure 23: Input field with error #2 (own screenshot)

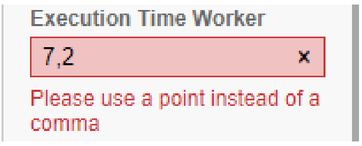

Figure 24: Input field with error #3 (own screenshot)

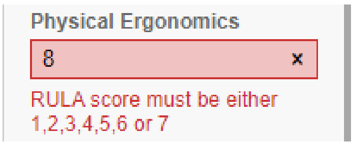

Figure 25: Input field with error #4 (own screenshot)

4.1.2 Dashboard

A dashboard that serves as a foundation for decision support was created through multiple development iterations (DP 2). To ensure good usability, the dashboard is implemented as a responsive CSS grid with aesthetics in mind (DP 5). As a result, the colors in the three sections were kept consistent by using complementary colors and colors from the same family.

Plotly.js [76] was utilized to develop the graphs in the dashboard; it is a JavaScript framework for plotting diagrams inside of a HTML content division element. With this framework, many of the graph's properties, such as color, legend, and appearance, can easily be changed. Another advantage of this framework is that the graphs are responsive, which means they will automatically rescale depending on the size of the parent container.

Figure 26 depicts the UI with the added dashboard:

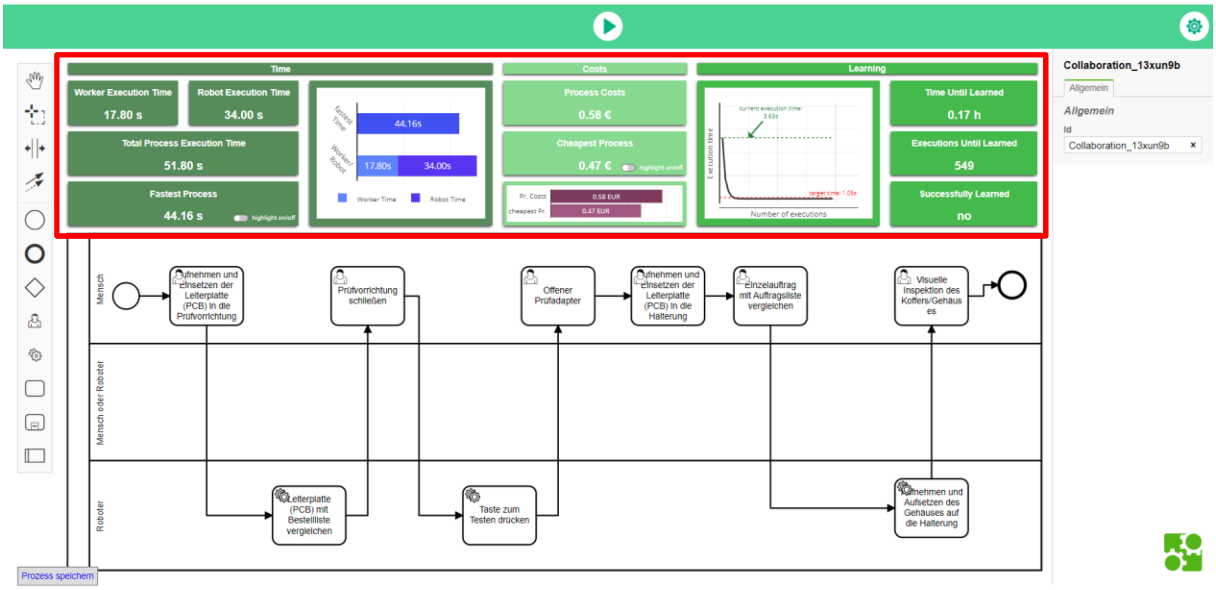

Figure 26: UI with added dashboard (own screenshot)

As can be seen, the dashboard is divided into three main areas:

- Time
- Costs
- Learning

Time area

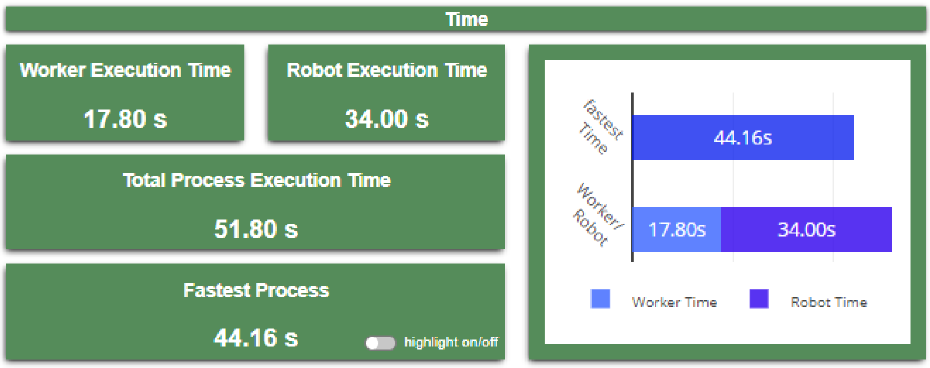

Figure 27: Time area of the dashboard (own screenshot)

The time area consists of five subelements. The worker execution time indicates the total amount of time required by the worker to complete the process, whereas the robot execution time represents the total amount of time required by the robot to complete the process.

The total process execution time shows the total time required for one process iteration, which is the sum of worker and robot time.

The fastest process element in the bottom section displays the shortest time possible to complete the process. We use Equation 3 to calculate the fastest process.

With the highlight on/off button, the tasks that need to be allocated to the worker/robot to result in the fastest process are highlighted. For further information, please refer to Section 4.1.4.

Furthermore, a simple diagram displaying the values is shown to enhance the overview of execution times and to compare the distribution between worker and robot execution times in a graphical way, as well as the possibility to compare it to the fastest possible process allocation.

Costs area

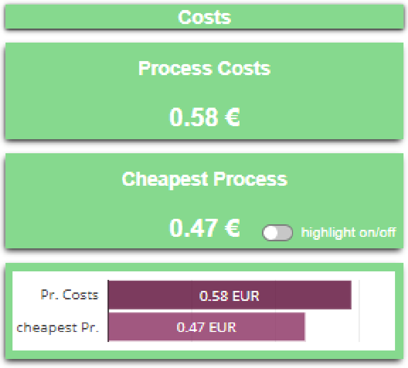

Figure 28: Costs area of the dashboard (own screenshot)

The cost section consists of three subelements: process costs, which indicates the total costs of the process in the current allocated task; the cheapest cost element, which is the minimal cost of the process if allocated accordingly; and the third element is a graph of the two previously mentioned elements for a simple and easy presentation of those values.

To calculate the process costs, we sum up the results of Equation 1 and Equation 2. To calculate the cheapest process, we utilize Equation 4.

With the highlight on/off button, the elements that need to be allocated to the worker/robot to result in the cheapest process are highlighted. For further information, please refer to Section 4.1.4.

Learning area

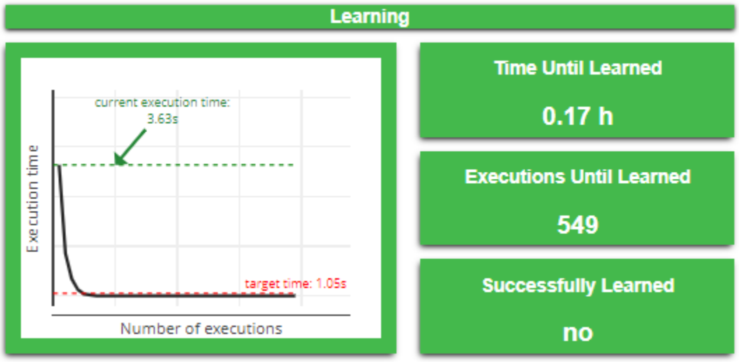

Figure 29: Learning area of the dashboard (own screenshot)

The learning curve model, described in Section 2.3.3, is implemented in the third main section – learning – to estimate the learning times and display the learning curve (DP 4).

As shown in Figure 29, the area is divided into four sub-elements. The first element is time until learned, which is calculated by the result of Equation 10 minus the sum of execution times of previously completed executions and indicates the remaining time required to learn the task. The second element is executions until learned, which is calculated by subtracting the number of already completed executions from Equation 9.

The third element, successfully learned, provides information on whether a task can be considered learned, determined by whether the worker meets the reference performance or so-called target time.

A graph representing the worker's learning curve can be seen in the fourth element, located on the section's left side. The x-axis in this graph represents the number of executions, while the y-axis represents the execution time.

The current learning progress is indicated by the intersection of the curve with a green dotted line representing the current execution time. A red dotted line represents the target time or reference performance. When the two lines intersect, it signifies that the reference performance is met, and the task can be considered learned.

The worker can use the values of the described elements as well as the graphical information as a decision parameter to determine where the worker stands in the learning process.

Moreover, the learning section allows distinguishing between tasks that have been learned and those that have not. As a result, tasks that have not been learned can be restricted to the worker and are only permitted to be assigned to the robot if the worker meets the reference performance.

Following the execution of the process, the dashboard values are recalculated to ensure that the correct values are displayed for each process iteration.

4.1.3 User Properties

To customize the learning curve model for user-specific calculation of the learning time and the expected number of executions until the worker has learned the task, it is necessary to gather user-specific data. For this purpose, the user properties panel was implemented using HTML/CSS/JS and JQuery (DP 4). The panel fades when the gear symbol is clicked, and the worker can enter the user-specific values.

The panel includes parameters from Section 2.3.3, namely the input fields aiming, wrist-finger speed, age, gender, work instructions, and experience, in a drop-down menu, which can be seen in Figure 30.

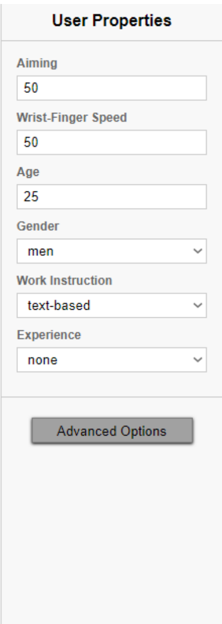

Figure 30: User properties (own screenshot)

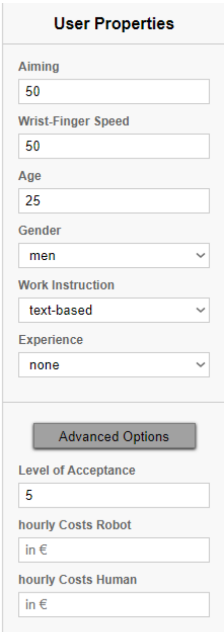

Figure 31: User properties with advanced options visible (own screenshot)

Furthermore, there is a button labeled "Advanced Options." When it is clicked, additional input fields for the level of acceptance and the hourly costs for the human and robot are displayed (see Figure 31), which should only be changed by users with further knowledge.

This improves the system's flexibility by enabling the entry of actual costs for both humans and robots as well as the appropriate level of acceptance for the learning model.

4.1.4 Functionality of Modeler

To improve decision support and usability of the application some additional functions were implemented.

Basic functions

At first, some basic functionalities were implemented. The dashboard's values and graphs are updated when a task is changed from a user task to a robot task or vice versa as well as after every input in the properties panel to ensure the dashboard is always up to date (DP 2).

Furthermore, if an agent is assigned a task that is not feasible, an error message is displayed in the properties panel to inform the user how to resolve the issue, as seen in Figure 32. (DP 5)

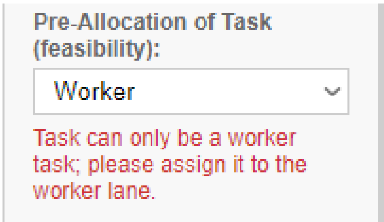

Figure 32: Error message for an unfeasible task allocation (own screenshot)

When this error or a validation error from Table 14 occurs, the tasks are highlighted in red in the modeler to make it more evident that an error has occurred.

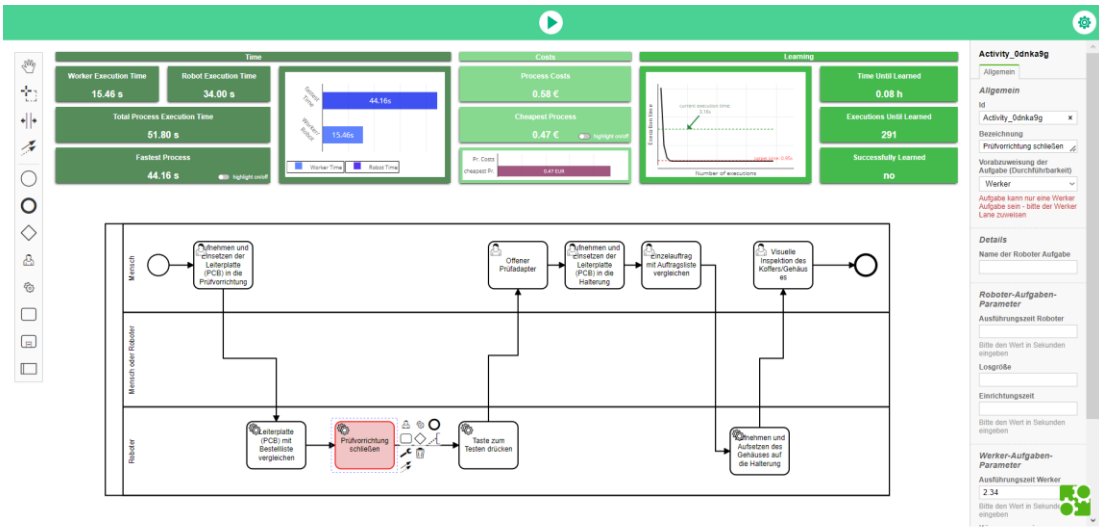

Figure 33: Example of a highlighted task, that is allocated to an agent for whom it is not feasible to carry out (own screenshot)

Deployment

Of course, the process can also be deployed as in the old prototype, which is highlighted by a moving green frame around an active task. Since the dashboard is only required for task allocation prior to execution, all other areas are hidden, as shown in Figure 34, to ensure that the user's attention is on the process execution, and distractions are minimized. (DP 5)

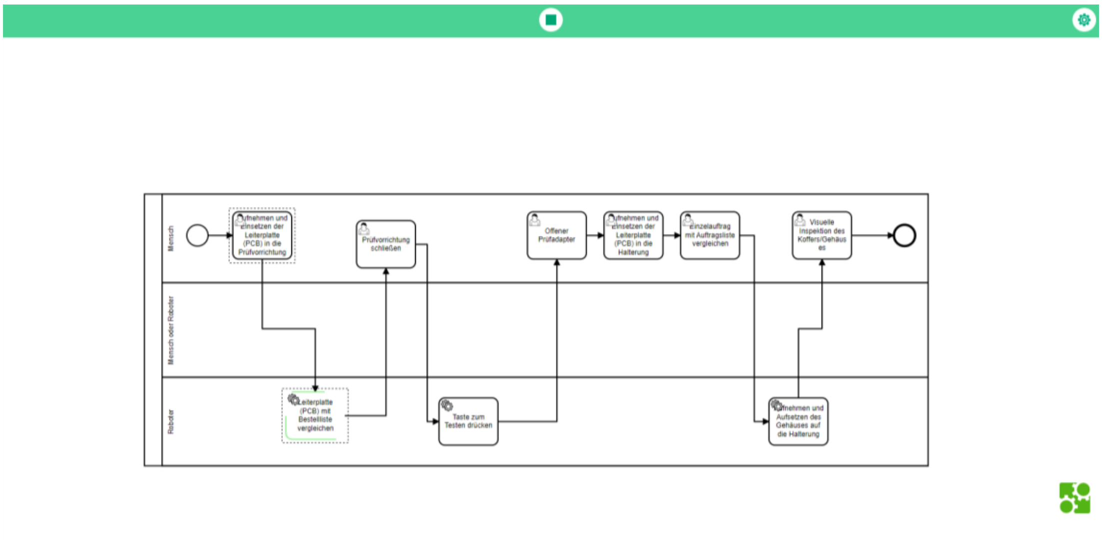

Figure 34: UI during execution of the process (own screenshot)

Before deploying the process, the system checks for active errors. If there are any errors, deployment of the process is not possible. Instead, after the play button is clicked, a pop-up appears with the message to resolve all errors in the red-highlighted tasks before deployment of the process is possible (see Figure 35). In addition to performing error checking, the system verifies if a task is allocated as shareable, which also results in a message to assign the task to either a worker or robot. (DP5)

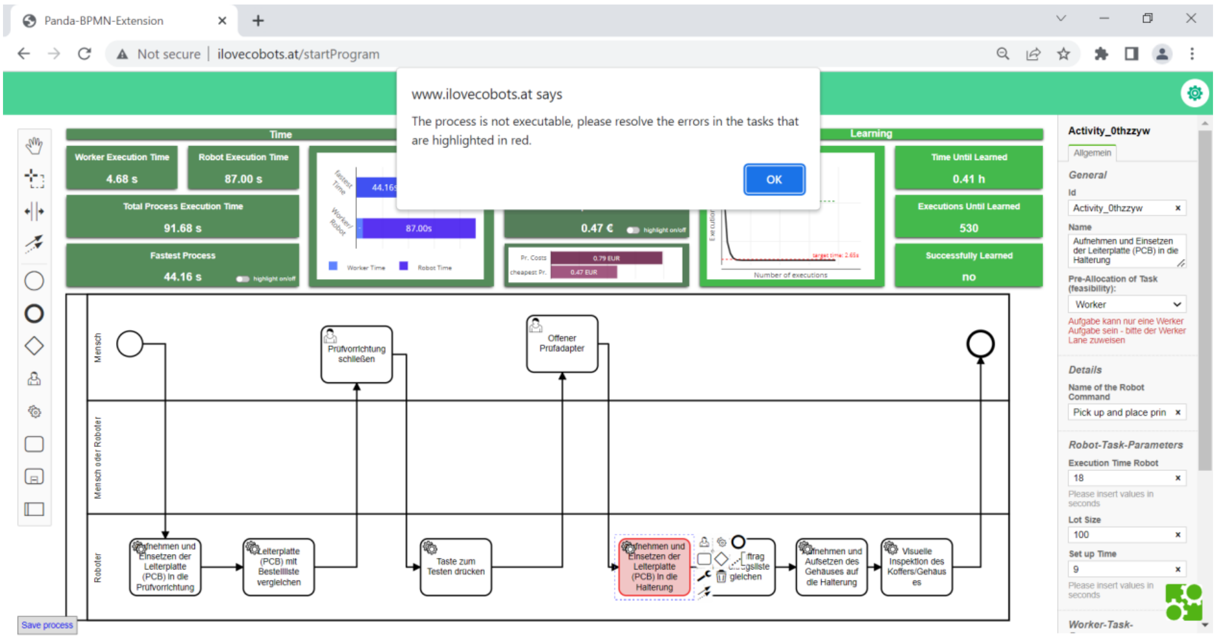

Figure 35: UI with error message if process is not executable (own screenshot)

Decision-making

A further important feature for decision-making is that tasks with the potential to be completed faster or at a lower cost can be highlighted (DP 3). This function can be accessed through the dashboard by enabling the highlight on/off button in the relevant areas. If the fastest process is desired, any task that the human or robot can execute in a shorter time is highlighted in purple.

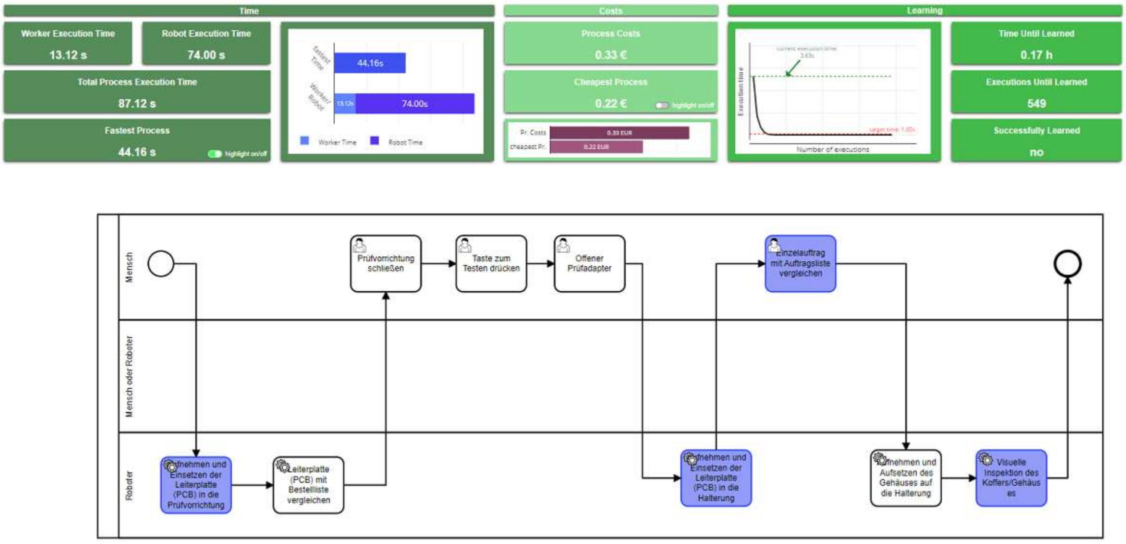

Figure 36: Highlighted tasks for fastest execution time (own screenshot)

Figure 37 shows the results after the tasks are allocated to the fastest possible execution time.

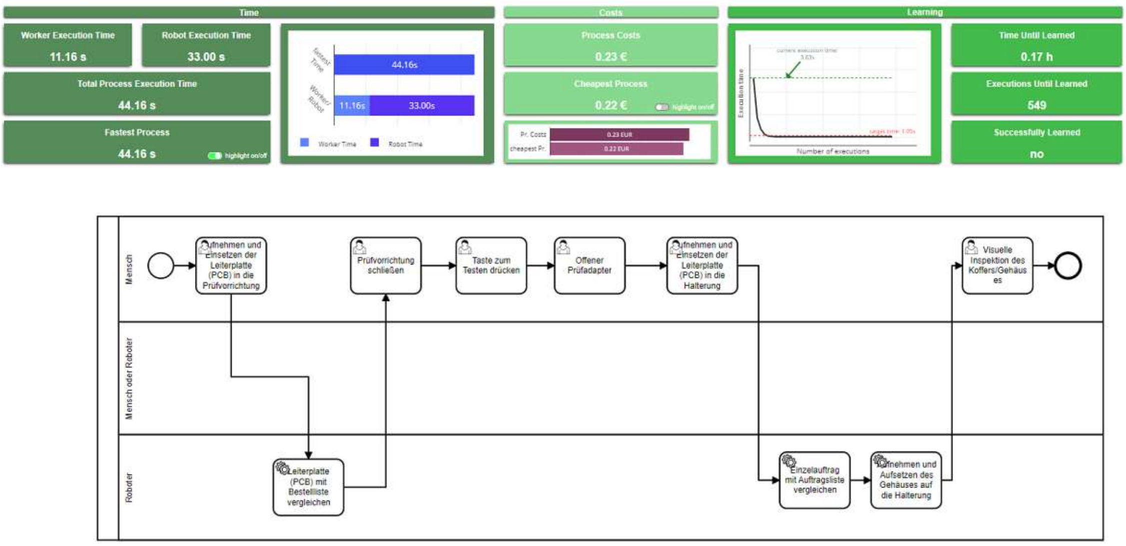

Figure 37: Fastest process allocation (own screenshot)

If the cheapest process is preferred, any task that can be completed at a lower cost by a human or robot is highlighted in red.

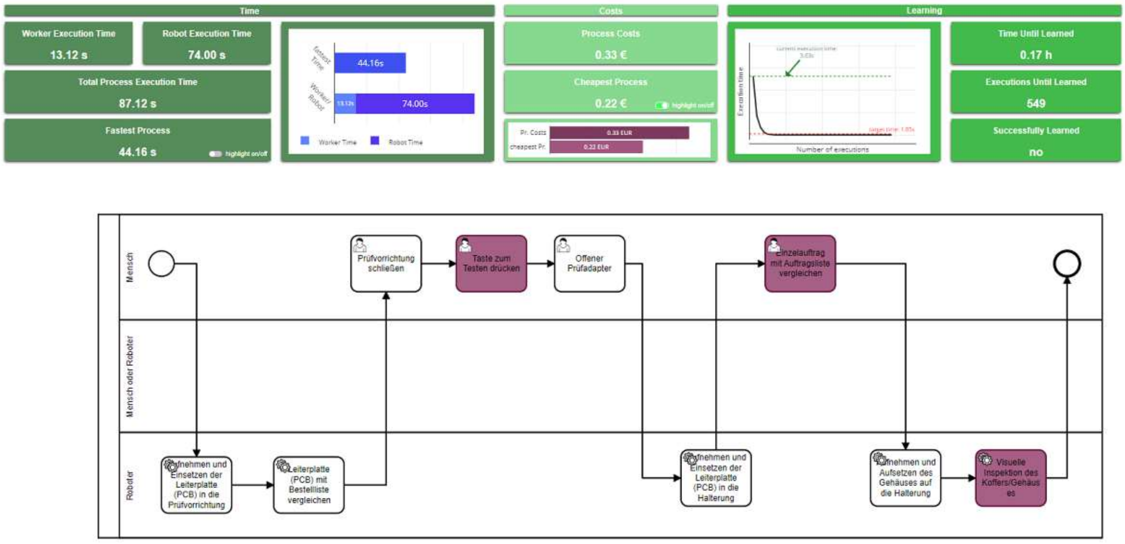

Figure 38: Highlighted tasks for cheapest execution (own screenshot)

The result of the cheapest task allocation is shown in Figure 39:

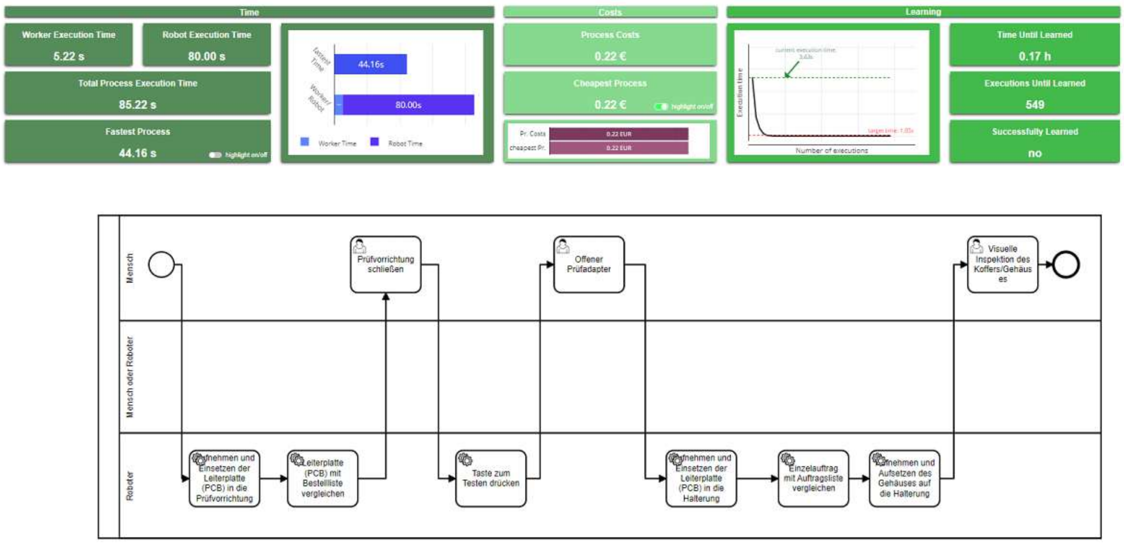

Figure 39: Cheapest process allocation (own screenshot)

An overview of the implemented design parameters 1–4 is depicted in Figure 40. As design parameter 5 has influence on the whole UI, it is not possible to highlight DP 5 as a single square in the Figure.

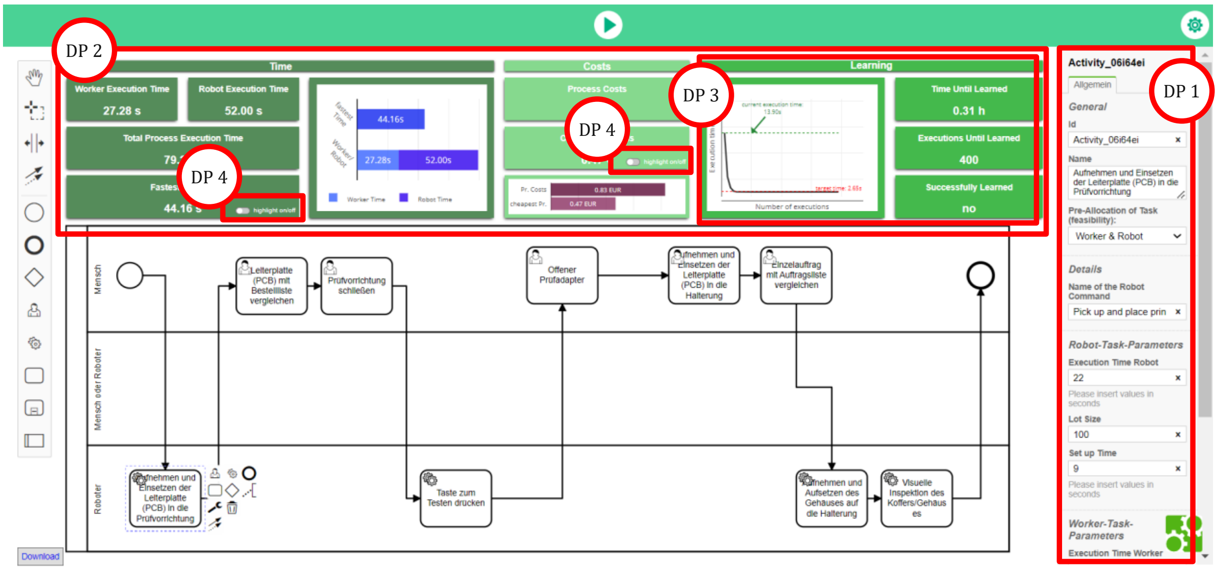

Figure 40: UI with DP 1–4 highlighted (own screenshot)

4.2 UI Expert Feedback Cycle (Second and Third Iteration)

In the next iteration, a heuristic evaluation was conducted. The evaluation consisted of a detailed examination of the interface by five evaluators. Its purpose was to detect any potential usability issues that may have been missed during previous development stages.

Once the issues were identified, they were addressed and resolved in another iteration based on the feedback from evaluators, which is discussed in detail in the following.

4.2.1 Heuristic Evaluation

Before delving into the details of the evaluation process, it is essential to understand what a heuristic evaluation is. Heuristic evaluation is defined as a method “for finding usability problems in a user interface design by having a small set of evaluators examine the interface and judge its compliance with recognized usability principles (the ‘heuristics’)” [77]. Nielsen [78] defines 10 heuristics with which an evaluation can be performed. Because the implemented system is intended to be used with cobots, particular heuristic guidelines are required [79]. Hence, we use the set of 24 design guidelines proposed by Frijns [79] to ensure having guidelines that are specific to cobot UI design in a manufacturing context [79].

An example of a guideline is listed in Table 15 [79]:

Table 15: Description of example guideline number 11

Guideline	Task progress: Communicate to the user which task is being executed.
Explanation	The GUI should make it easy for the user to follow task execution by indicating previous, current, and next steps.
Example	Example: When the robot is executing a series of actions, the current action that is being executed is highlighted on the GUI.

The full set of guidelines can be found in the Appendix (see Chapter 11.2).

The evaluation was conducted by five experts, either robotics experts, UI/UX experts, or both. Because of the COVID-19 pandemic and to get responses from a sufficient number of experts, the evaluation was conducted online. Prior to the evaluation, participants were asked to sign data protection information and consent forms.

The evaluation was performed in three steps:

1. Participants received standardized information about the system as well as online access to the system with a predefined sample process.
2. The participants were provided with the guidelines and asked to use them individually to identify and describe usability issues.
3. After collecting the written reviews, they were analyzed and examined for duplicates to derive a list of actions.

Results

The results for every guideline of the evaluations are shown in Table 16. In summary, 14 guidelines were violated. The number of issues found per guideline contains duplicates when different experts identified the same issues and does not differentiate if evaluators identified multiple issues.

Table 16: Results of heuristic evaluation

Guideline No.	Guideline title	Number of Issues found
1	System state awareness: Inform the user on the cobot's state.	3
2	Situation awareness: Inform the user regarding the cobot's environment	3
3	Accessibility of information: Allow users to access information required for the task	4
4	Feedback: The UI is responsive to user actions.	0
5	Affordances: Signify how the user can interact with the cobot.	3
6	Errors: Give clear explanations and steps to recover when errors occur.	2
7	Mental model: Support the user in understanding the way the system works.	2
8	Help and documentation: Provide contextual help and documentation.	5
9	Support user learning: Help the user solve their (automation) problem.	1

10	Efficiency: Avoid unnecessary work on the user's side.	0
11	Task progress: Communicate to the user which task is being executed.	0
12	Reuse: Enable reuse of previous work.	5
13	Human factors: Design cobot and UI with ergonomics and accessibility in mind.	1
14	Avoid cognitive overload: Reduce mental strain.	1
15	User attention: Support the user in directing their attention.	2
16	Level of automation: Let the user determine the level of human input.	1
17	Adaptable system architecture: Enable easy software integration after hardware exchange.	0*
18	Adaptable tasks: Support easy editing of robot programs.	2
19	Consistent behavior: Make sure cobot and UI behave in a consistent way.	0*
20	Multimodal UI: Consider the relation between different interaction modalities.	0*
21	Graphic design: Design GUI items with usability, accessibility, and aesthetics in mind.	7
22	Clarity of interface: Ensure the UI is easy and intuitive to use.	0
23	High vs. low complexity: Display programming functions at different levels of detail.	5
24	Customizability: Support user preferences.	0
Total issues found		**47**

*The experts determined guidelines 17, 19, and 20 as not applicable for this evaluation

Based on the reviews from the experts, a list of issues was created. At first, the results were cleaned of duplicates, as some evaluators detected similar issues. After further analysis of the feedback, some recommendations were determined to be not feasible in a certain amount of time or fell outside the scope of this thesis. For instance, an evaluator suggested for guideline 2 to add a live 3D model of the cobot to the UI, which is not feasible due to the limited time frame. Furthermore, the experts determine guidelines 17, 19, and 20 as not applicable for this evaluation. The final list of issues is presented in Table 17.

Table 17: List of issues

Guideline No.	Guideline title	Issues
1	System state awareness: Inform the user on the cobot's state.	The system state is not clear enough (unclear if the process is being executed at the time).

3a	Accessibility of information: Allow users to access information required for the task.	The values for “Worker Execution Time”, “Robot Execution Time”, “Total Process Execution Time” are initially 0.00 and are only initialized when the play button or the first box for a worker task is clicked.
3b	Accessibility of information: Allow users to access information required for the task.	The download button was not initialized from the beginning, only after clicking the play button or a worker task.
12	Reuse: Enable reuse of previous work.	The download/upload button could not be found.
13	Human factors: Design cobot and UI with ergonomics and accessibility in mind.	Difficult visibility with long processes when the process is executed.
21a	Graphic design: Design GUI items with usability, accessibility, and aesthetics in mind.	Some contrast-ratio issues in parts of the UI.
21b	Graphic design: Design GUI items with usability, accessibility, and aesthetics in mind.	The headings “Time,” “Cost,” and “Learning” in the dashboard are too small.

Furthermore, the recommendations of the evaluators for the respective issues are presented in Table 18.

Table 18: List of issues and respective recommendations

Guideline No.	**Issues**	**Recommendation**
1	The system state is not clear enough (unclear if the process is being executed at the time).	Make the global state clearer, for example, by placing “running” “paused/stopped” beside the play/stop button.
3a	The values for “Worker Execution Time”, “Robot Execution Time”, “Total Process Execution Time” are initially 0.00 and are only initialized when the play button or the first box for a worker task is clicked.	Make sure the dashboard is initialized properly from the beginning.
3b	The download button was not initialized from the beginning, only after clicking the play button or a worker task.	Initialize the download button from the beginning.
12	The download/upload button could not be found.	There should be a small area with two actions: download diagram and upload diagram.
13	Difficult visibility with long processes when the process is executed.	Focus on the active process to avoid manual scrolling.
21a	Some contrast-ratio issues in parts of the UI.	Fix contrast issues using the lighthouse test.
21b	The headings “Time,” “Cost,” and “Learning” in the dashboard are too small.	Make the headings “Time”, “Cost,” and “Learning” more readable.

4.2.2 Resolution of Guideline Violations

Following the creation of the list of issues, the issues were resolved using the evaluators' recommendations, resulting in the following changes:

Guideline 1

A text was added next to the play button after a few evaluators determined that the global system state lacked sufficient clarity. It displays "Click the play button to start" if the process is not currently running (see Figure 41) and "Running..." if it is currently being executed (see Figure 42).

Figure 41: Label if process is not currently being executed (own screenshot)

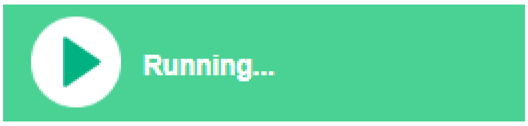

Figure 42: Label if process is currently being executed (own screenshot)

Guideline 3a

Another guideline violation was detected, namely that labels for worker execution time, robot execution time, and total process execution time were initially displayed as 0.00 (see Figure 43). These elements would only be updated to their starting values if the play button or a worker task was clicked. In order to address this issue, the initialization of the process parameters was modified so that the initial values are displayed on the dashboard. (see Figure 44).

Figure 43: Dashboard does not load on start (own screenshot)

Figure 44: Dashboard loads on start (own screenshot)

Guideline 3b

Similar to Guideline 3a, the download button was not properly initialized. This issue was resolved by modifying the initialization, resulting in the download button functioning correctly when the system loads.

Guideline 12

One of the reviewers mentioned that they were unable to locate the function to download the process. According to the recommendation, there should be a simple and small area with two actions, save process and upload process, which should make it easier to reuse diagrams.

For this recommendation, the button download button is renamed "Save process" to make it more intuitive that the process can be saved. Unfortunately, the upload function in this bpmn.io version is restricted to dragging and dropping the saved file into the browser. However, the save and load buttons will be standard functionality in a newer version of bpmn.io and can be used in future research projects.

Guideline 13

To address the evaluator's concern about the difficulty of manual scrolling during lengthy processes, the active element is now centered when the process is executed, eliminating the need for manual scrolling.

Guideline 21a

When an evaluator conducted a lighthouse accessibility test, it was discovered that the costs section of the dashboard had contrast ratio issues, making it difficult for users to read and comprehend the text. In particular, the contrast ratio for the cost section was determined to be 3.9:1, falling short of the WCAG-recommended minimum [80] of 4.5:1. This was resolved by changing the section's color from #86D98E to #557E5A, which resulted in a contrast ratio of 4.6:1 and improved readability. A comparison of the colors is displayed in Figure 45 and Figure 46.

Figure 45: Color of cost section before (own screenshot)

Figure 46: Color of cost section after (own screenshot)

Guideline 21b

Feedback was received indicating that the headings "Time," "Costs," and "Learning" were too small and thus difficult to read. To address this issue, the font size for these headings was increased to 14px, and a padding top of 2px was added to center the elements.

4.3 Proof of Concept

After implementing the changes from the UI feedback cycle in Section 4.2, the final demonstrator appears as follows:

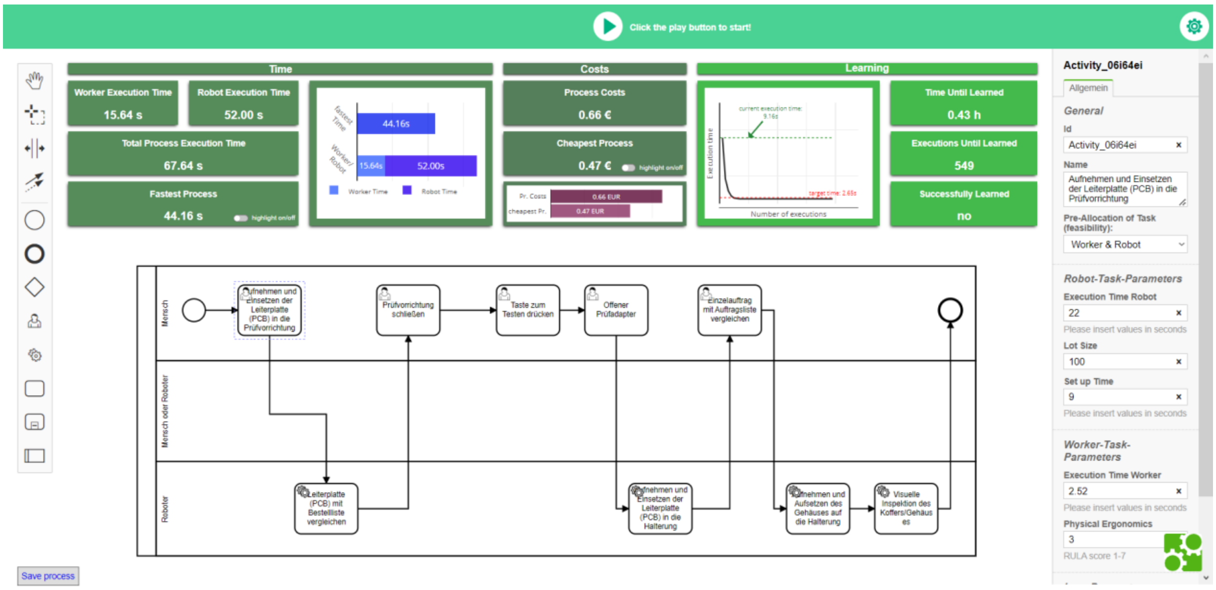

Figure 47: Final prototype (own screenshot)

This prototype can be downloaded at https://gitlab.tuwien.ac.at/e330-03-research-unit-of-human-machine-interaction/public/bpmn-prototyp-including-workplaced-learning

To demonstrate the applicability of the proposed solution, an assembly process of a timing relay is used. This process is replicated from a real-world setup from TELE Haase Steurgeräte Ges.m.b.H Vienna, Austria [19]. The process includes completing 18 tasks in total, as shown in Table 19.

Table 19: Task list of assembly process [19]

No.	Task	Assembly Function	Function specified
1	Pick up and place printed circuit board (PCB) in test fixture	Handling	Moving
2	Compare printed circuit board (PCB) with order list	Checking	Comparing
3	Close test fixture	Handling	Moving
4	Press button for testing	Checking	Confirming
5	Open test fixture	Handling	Moving

6	Pick up and place printed circuit board (PCB) on fixture	Handling	Moving
7	Compare individual order with order list	Checking	Comparing
8	Pick up and place case/housing on fixture	Handling	Moving
9	Visual inspection of case/housing	Checking	Comparing
10	Pick up and place PCB in case/housing	Joining	Putting together
11	Pick up and place cover on case/housing	Handling	Moving
12	Tighten/fix the cover	Joining	Putting together
13	Visual inspection of the component	Checking	Inspecting
14	Pick up and place component for labeling	Handling	Moving
15	Press button for labeling	Checking	Confirming
16	Visual inspection of the component	Checking	Inspecting
17	Updating the order list	Special	Marking
18	Placing the product for transport	Handling	Saving

The results of a task analysis conducted by Schmidbauer [19] on this process are presented in Table 20.

Table 20: Task analysis of example process [19]

Criteria										
Task number		**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**
Hourly costs human [€]	cH	38.04								
Hourly costs robot [€]	cR	6.00								
Execution time human [s]	tH	2.52	.54	2.34	.9	2.34	2.52	.54	NA	.54
Execution time robot [s]	tR	22	11	NA	7	NA	18	6	16	7
Setup costs [€]	csetup	9.17	9.17	NA	9.17	NA	9.17	9.17	9.17	9.17
Repetition rate	f	100	100	NA	100	NA	100	100	100	100
Setup costs/ repetition rate [€]	csetup/ f	.09	.09	NA	.09	NA	.09	.09	.09	.09
Costs per part human [€]	cPH	.04	.01	.04	.01	.04	.04	.01	NA	.01
Costs per part robot [€]	cPR	.13	.11	NA	.01	NA	.12	.10	.12	.10
Task number		**10**	**11**	**12**	**12**	**14**	**15**	**16**	**17**	**18**
Execution time human [s]	tH	2.52	2.5	.72	.54	2.52	.9	.54	2.34	1.8
Execution time robot [s]	tR	11	14	NA	7	18	7	7	8	NA
Setup costs [€]	csetup	9.17	9.17	NA	9.17	9.17	9.17	9.17	9.17	NA
Repetition rate	f	100	100	NA	100	100	100	100	100	NA
Setup costs/ repetition rate [€]	csetup/f	.09	.09	NA	.09	0	.09	.09	.09	NA
Costs per part human [€]	cPH	.04	.04	.01	.01	.04	.01	.01	.04	.03
Costs per part robot [€]	cPR	.11	.12	NA	.10	.12	.10	.10	.11	NA

To define pre-allocation of tasks, a robot feasibility check as well as a physical ergonomic evaluation are performed by Schmidbauer [19]. The robot feasibility check indicates if a task can be executed by the robot, which is indicated by the result “B” [19]. If tasks are not feasible for the robot, they should be done by the human, which is indicated by an “H” [19]. Additionally, physical ergonomics are evaluated [19]. If the ergonomics are insufficient for humans, the task should be assigned to the robot, which is indicated by the letter “R” [19]. The results are displayed in Table 21.

Table 21: Task assignment of example process [19]

Criteria									
Task number	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**
Robot feasibility check	B	B	H	B	H	B	B	B	B
Physical ergonomics	B	B	H	B	H	B	B	R	B
	10	**11**	**12**	**13**	**14**	**15**	**16**	**17**	**18**
Robot feasibility check	B	B	H	B	B	B	B	B	H
Physical ergonomics	B	B	H	B	B	B	B	B	H

This results in the following possible starting allocation:

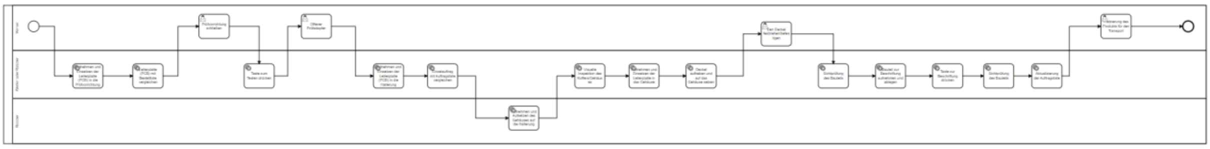

Figure 48: BPMN process chart of example process (own screenshot)

The fastest and cheapest possible process allocations are shown in Table 22.

Table 22: Economic efficiency evaluation [19]

Economic Efficiency Evaluation									
Task number	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**
Fastest, most time-efficient process	H	H	H	H	H	H	H	R	H
Cheapest, most cost-efficient process	H	H	H	H	H	H	H	R	H
	10	**11**	**12**	**13**	**14**	**15**	**16**	**17**	**18**
Fastest, most time-efficient process	H	H	H	H	H	H	H	H	H
Cheapest, most cost-efficient process	H	H	H	H	H	H	H	H	H

To test the prototype, we calculate the fastest and cheapest process and compare it to the values calculated by the prototype and displayed on the dashboard.

To determine the fastest process, we use Equation 4 from Section 2.1.2 with the variables from Table 20, which results in 42.62 s. For calculating the cheapest process, we use Equation 3 with the variables from Table 20, which results in €0,53.

Now we assign the tasks to the cheapest process and examine which values are displayed on the dashboard:

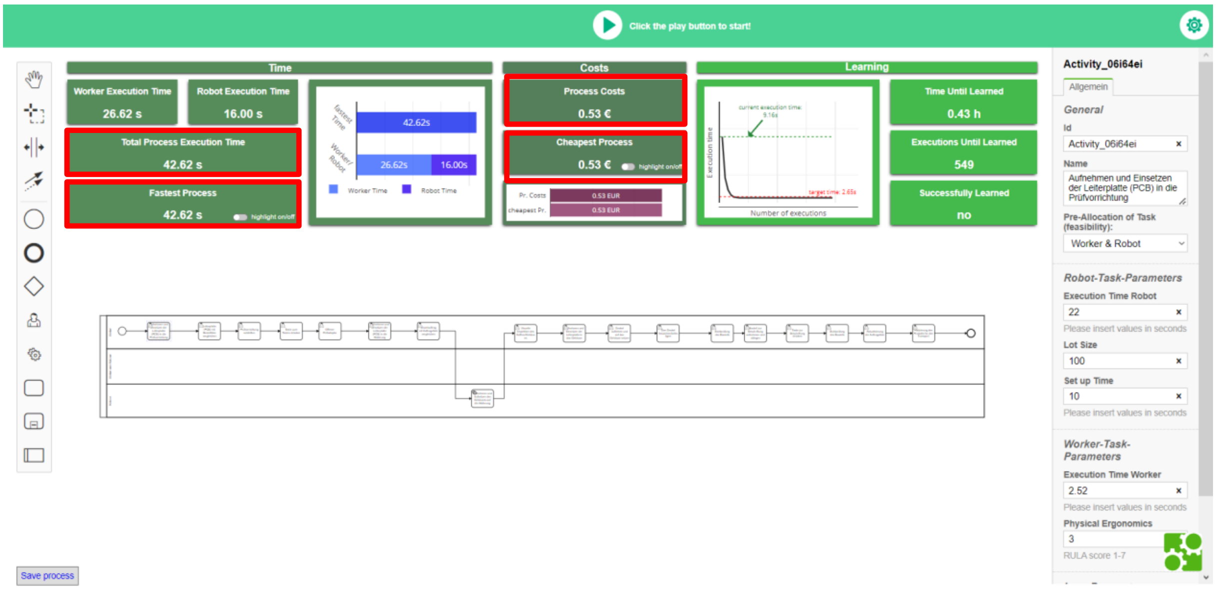

Figure 49: Test case, cheapest and fastest process, zoomed out to see the whole process at once (own screenshot)

As shown in Figure 49, the process allocation as depicted in Table 22 is shown with the dashboard displaying the same values as those calculated above for the fastest and most cost-efficient processes.

The next step is to calculate the value for the most expensive process allocation and compare it to the value displayed on the dashboard. The following equation is used to calculate the most expensive process:

$$c_{max} = \sum_{i=1}^{n} \max\left(c_{PH,i}, c_{PR,i}\right) \qquad \text{Equation 11}$$

Using Equation 11 and the parameters listed in Table 20, we can calculate the most expensive process allocation, which results in a value of €1.66. The distribution of the tasks for the most expensive allocation is shown in Table 23:

Table 23: Most expensive task allocation

Economic Efficiency Evaluation									
Task number	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**
Most expensive process	R	R	H	R	H	R	R	R	R
	10	**11**	**12**	**13**	**14**	**15**	**16**	**17**	**18**
Most expensive process	R	R	H	R	R	R	R	R	H

Now we assign the tasks to the most expensive process and examine which values are displayed on the dashboard:

Figure 50: Test case - most expensive process allocation

As shown in Figure 50, the process depicted in Table 23 is allocated, with the dashboard displaying €1,66, which is the same value as calculated above for the most expensive process.

Lastly, we demonstrate and test the proper calculation for the executions needed to learn the task and the expected learning time for Task 1 of the example process. We chose the following user properties as a test case:

Table 24: Test case properties for learning prediction

Properties	**Value**
Aim	60
Arm-wrist speed	55
Age	21
Gender	Men
Work instruction	Text based
Experience	None
Entropy of components	1
Entropy of process	1

By using the test parameters from Table 24 and the parameters from Task 1 to solve Equation 9, we can approximate the number of repetitions required to reach the reference performance for Task 1, which in this case is 1,963 times.

Next, we use Equation 10 to compute the expected learning time for Task 1, which in this example is 1.51 h.

Now, we enter the test case parameters in the user properties panel in the prototype and compare the values in the dashboard to the calculated numbers.

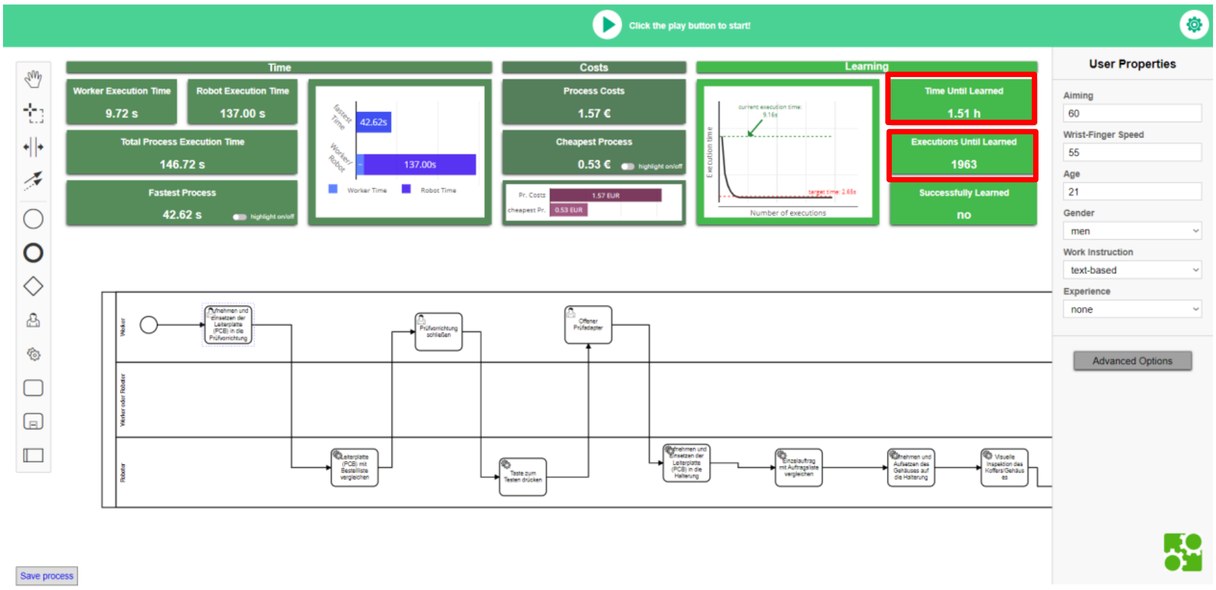

Figure 51: Test case for learning time and executions until learned

As shown in Figure 51, the number of repetitions required and the expected learning time to reach the reference performance for Task 1 are 1,963 and 1.51 h, respectively the same as the numbers calculated above.

In summary, we have presented the implementation and testing of the prototype in this chapter. The next chapter focuses on the user-friendliness evaluation.

5 Evaluation

This thesis aims to enhance decision support and enable learning opportunities for adaptive task sharing in a user-friendly way. Whether the implemented changes for the prototype that enable adaptive task sharing can be called user friendly is clarified in this chapter.

5.1 Study Design

Due to the coronavirus pandemic, the study was conducted online to protect the participants. That is why we used the website https://ilovecobots.at to conduct the study.

Because of the online format, a few things must be kept in mind:

- The cobot was not present during the study. To demonstrate the execution of cobot tasks, a five-second delay was implemented during the execution of the process.
- The participants were not able to ask questions during the evaluation. To prevent a negative impact on the evaluation, clear instructions were essential. Therefore, the participants were provided with a video tutorial to receive clear introductions and overview of the task. Additionally, the study was tested by volunteers before launching the evaluation in order to eliminate any ambiguity and prevent errors in advance.

5.1.1 Hypothesis

To clarify whether the proposed solution is user friendly, the following hypothesis is tested:

Hypothesis H1: The enhanced prototype exhibits good usability and sufficient user experience to be considered user friendly.

5.1.2 Study Procedure

The study was conducted online and divided into the following steps:

1. Introduction to the study/data protection
2. Task explanation
3. Solving the task
4. Questionnaire

Introduction/data protection

On the landing page, participants first received a brief overview of the study's purpose and how the procedure of the evaluation will be conducted, as can be seen in Figure 52, Figure 53, and Figure 54. Following the introduction, participants were informed about the personal data that would be collected and processed and their rights according to the GDPR (see Figure 55). After accepting the voluntariness of participation and the data privacy policy, the participants were forwarded to the next step.

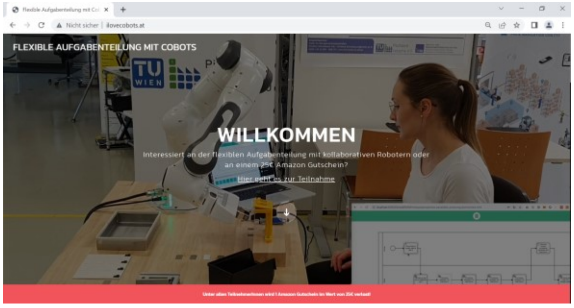

Figure 52: Landing page (own screenshot)

Figure 53: Explanation of study (own screenshot)

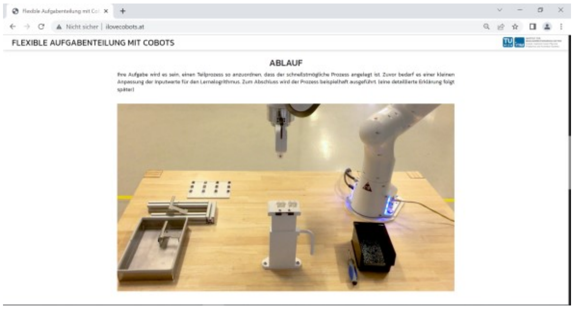

Figure 54: Brief explanation of procedure (own screenshot)

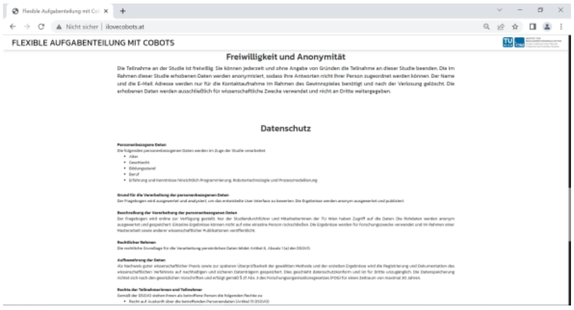

Figure 55: Voluntariness, anonymity and data protection (own screenshot)

Task explanation

At the top of the next page, two videos were shown. The first one displayed how the process looks in an experimental setting. This gave the participants a quick overview of the process. For the example process, we took the first nine tasks of the TELE Haase Steurgeräte Ges.m.b.H process, as discussed in Chapter 4.3, with all the task parameters already filled in.

The second video provided an explanation of the UI and its functionalities along with a description of the tasks to be completed by the participants during the evaluation.

Below the second video, the task description was also presented in written form, providing participants with a detailed explanation that eliminated the need to memorize the information shown in the video.

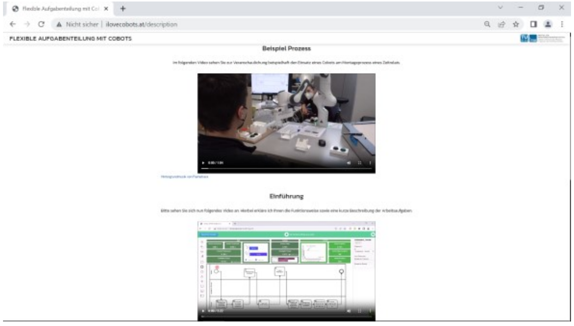

Figure 56: Videos of example task and description to solve the task (own screenshot)

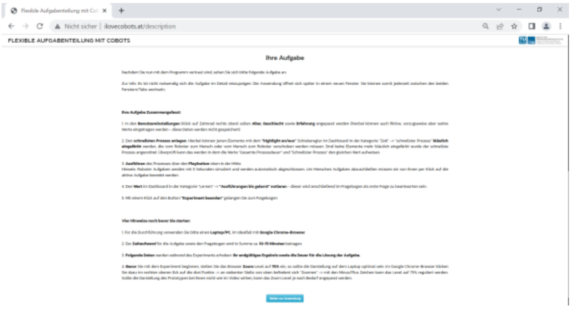

Figure 57: Written task description for the participants (own screenshot)

The tasks to be completed by the participants were as follows:

- Personalization: The participants had to enter their age, gender, and experience to individualize the learning curve.
- Allocation of the fastest process: The participants had to assign the tasks to get the fastest possible process allocation. The highlight on/off button was recommended for assistance and was explained in the description video.
- Execution of the process: The participants had to click on the play button to execute the process.
- Note a value: The participants had to note or remember the value from the dashboard element "Executions until learned" after execution of the task. This value differed for most participants based on the personalization task.
- The participants were then instructed to click the "End Experiment" button in the upper left corner to be redirected to the questionnaire.

At the bottom of the page, the participants were informed about the required presettings before proceeding to the practical part.

The study could only be completed with a laptop or PC System, and the Google Chrome web browser was recommended. Furthermore, a zoom level of 75% was suggested as it is ideal for laptop displays. The users also received instructions on how to adjust the zoom level in Google Chrome.

To proceed to the practical part, the participants had to click the button at the bottom of the page. The prototype opened in a new browser tab, which ensured that the participants had access to the instructions during the evaluation.

Solving the task

The prototype for the evaluation is shown in Figure 58.

To complete the first task, the participants had to click on the gear wheel in the upper right corner to open the user properties panel (see Figure 59) and enter the required data.

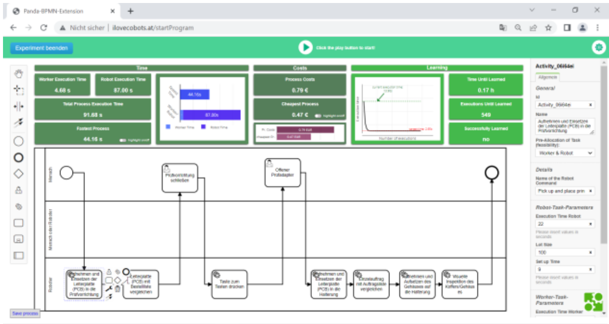

Figure 58: Prototype for evaluation (own screenshot)

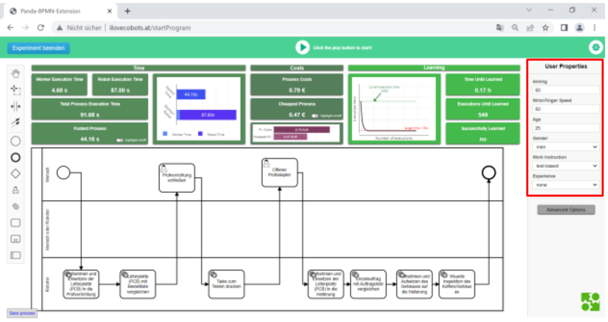

Figure 59: First task - personalization of user properties (own screenshot)

To allocate the fastest process, it was recommended to use the highlight on/off functionality, which colors every task that needs to be allocated to another agent (see Figure 60). After allocating the highlighted tasks, the fastest possible process is assigned, as shown in Figure 61.

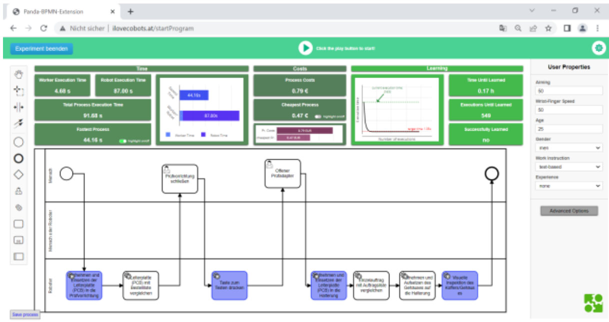

Figure 60: Second task - highlight the fastest process (own screenshot)

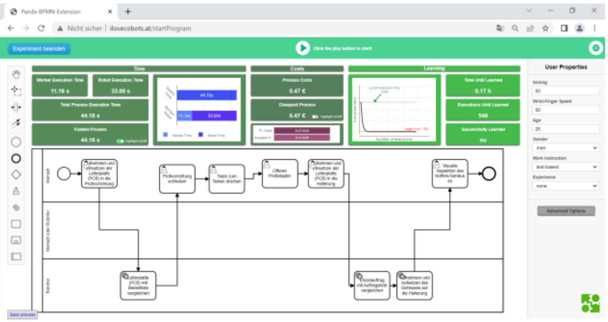

Figure 61: Second task - allocation of the fastest process (own screenshot)

With a click on the play button in the upper middle, the users started the execution of the process. The starting process is shown in Figure 62. With a click on an active worker task, the participants could finish the active tasks, which triggered the start of the next task in line.

Due to the absence of the cobot, the cobot tasks were simulated with a five-second delay instead of real-time execution. After completing the final task, the execution ended, and the dashboard and all other UI elements became visible again.

To finish the practical part, the participants had to note or remember the value of executions until learned from the dashboard (see Figure 63).

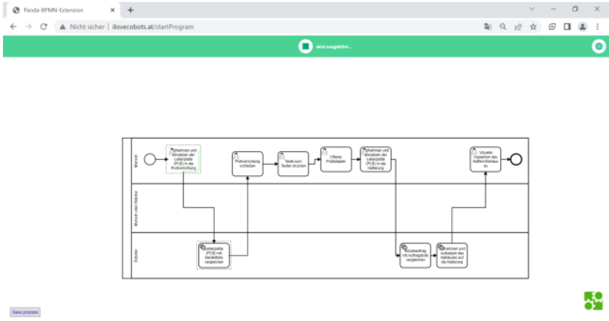

Figure 62: Third task - execution of the process (own screenshot)

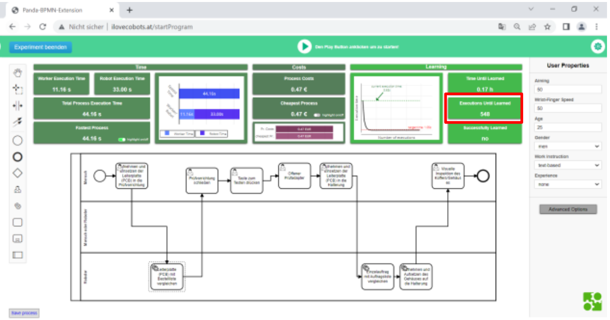

Figure 63: Fourth task - notation of value executions until learned (own screenshot)

The participants had to finish the experiment by clicking the “End Experiment” button in the upper left corner, after which they were redirected to the questionnaire.

Questionnaire

The questionnaire is composed of three parts:

- User task solution
- Demographic data
- Evaluation of the application

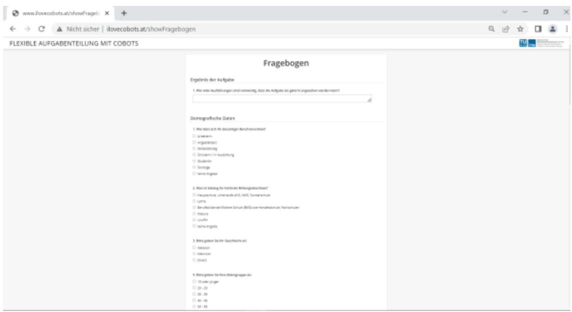

Figure 64: Questionnaire - the first and second parts (own screenshot)

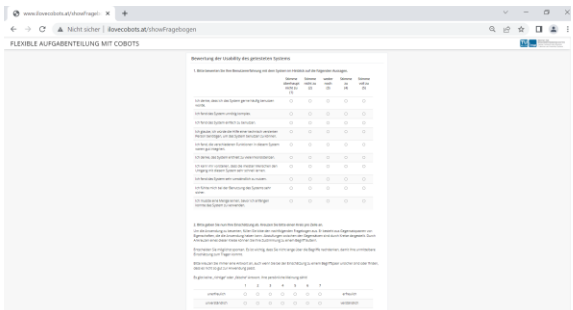

Figure 65: Questionnaire - the third part (own screenshot)

User task solution

In this section, the participants were asked to enter the value they noted in the last step of the practical evaluation part. This value was then compared to the value that should have been entered to see how many participants were able to find the right solution.

Demographic data

In the next part, the participants were asked to enter the following demographic data:

- Occupation

- Education
- Gender
- Age
- Technology affinity
- Experience with collaborative robots
- Programming knowledge
- Experience with process modeling/BPMN

Evaluation of the application

Three questionnaires were used for the assessment of the user friendliness of the application The System Usability Scale (SUS) [10], and the User Experience Questionnaire (UEQ) [39].

System Usability Scale

John Brooke's System Usability Scale (SUS) [10] is a simple tool for measuring a system's usability. Because of its cost effectiveness and flexibility, it is used in various applications, from research projects to industrial evaluations. The questionnaire consists of 10 items on a five-point Likert scale ranging from "strongly disagree" to "strongly agree" (see Appendix Chapter 11.1). It results in the SUS score, which ranges from 0 to 100, with 0 indicating extremely poor usability and 100 indicating extremely good usability. [10]

User Experience Questionnaire

The User Experience Questionnaire (UEQ) is designed to efficiently measure user experience [11]. Because of its ease of use and standardization, it is widely used for evaluating UX in interactive products [32]. In order to achieve stable results, at least 30 participants need to complete the survey [32]. It only considers aspects of pragmatic quality; it also considers hedonic quality [39]. The questionnaire consists of six dimensions: attractiveness, perspicuity, ,efficiency, dependability, stimulation, and novelty (see Section 2.2.2), with 26 items on a seven-point Likert scale (see Appendix Chapter 11.1) [39]. As a result, the UEQ provides a mean value for each dimension [11]. Furthermore, a benchmark for UEQ results is available to facilitate the interpretation of the outcome; it consists of a large sample of results from 468 studies [32]. The UEQ is available in many languages. In this study, we utilized the German version of the UEQ. For analyzing the results, we used a data analysis tool (Version 10) in the form of an Excel file, which can be downloaded from www.ueq-online.org.

Qualitative feedback questions

Two text areas were provided, one for feedback on any issues encountered during the evaluation and another for additional comments.

5.1.3 Technical Implementation

The online user study was conducted using the website https://ilovecobots.at, which was previously used for a master's thesis by Hader [6]. The website was originally built using the Laravel framework and the PHP programming language. HTML, CSS, and JavaScript were used for the front end, and the data was stored in a MySQL database. Amazon Web Services was used to run the Camunda BPMN engine and the task client required to execute the process. The questionnaire page also pre-existed and was built with HTML, CSS, JavaScript, PHP, and MySQL; it had to be modified for this thesis because different questionnaires were used. The self-hosted website had the advantage of increased data protection (GDPR) and better evaluation integrity as it was only possible for users who performed the evaluation tasks to answer the questionnaire. [6]

5.2 Participants

5.2.1 Data Cleansing

The database consisted of 104 entries of participants who clicked the link to the online study, of whom 51 completed the online task and the questionnaire. To count the participants, a unique session ID was saved on the user's device. With this ID, the website could retrieve the user's information when the site was closed and opened again. This ensured that the user was provided with the option to return and finish the evaluation later. In addition, this reduced the chance that a participant would engage in the research multiple times. When a participant wanted to access the website after submitting the questionnaire, the user was redirected to a thank-you page. Unfortunately, it is not possible to completely rule out the possibility of multiple engagements by a participant, because they might, for instance, use a different web browser or electronic device or engage in a private session.

A possible explanation for the large divergence between visits to the website and users who finished the study and questionnaire could be that some participants used mobile devices to open the landing page, while the study could only be completed using a laptop/PC system. This would trigger an entry for the mobile device in the database without the user being able to participate in the study. Therefore, a second visit with a laptop/PC system was necessary. Furthermore, web crawlers or visitors by accident could have created entries. Another explanation could be that users exited the evaluation prior to opening the task or submitting the questionnaire.

Because it is not entirely clear why these 53 visitors did not finish the study, these entries were deleted from the database.

5.2.2 Demographic Data

In total, there were 51 participants, of whom 25 were female, 24 were male, and two did not specify their gender (see Figure 66). Most participants were white-collar workers (34; 66.67%), while the rest (17; 33.33%) were students, blue-collar workers, self-employed, and others, as seen in Figure 67. Most participants were between 20–29 years (24) and 30–39 years old (14). Furthermore, seven participants of age 60+ completed the study, while the rest of the participants were 40–49 years (3) and 50–59 years (2) old, and only one participant was 19 or younger (see Figure 68).

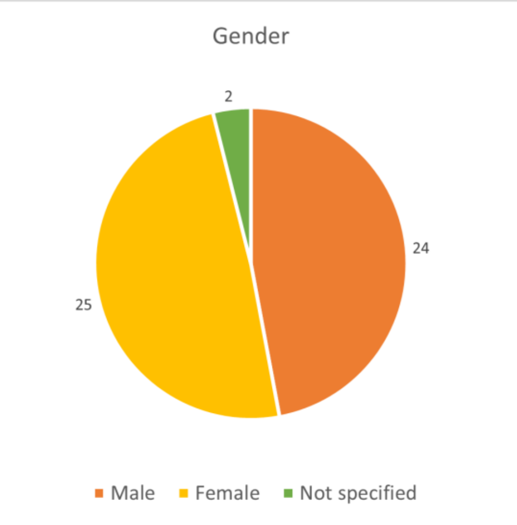

Figure 66: Gender (n=51; own figure)

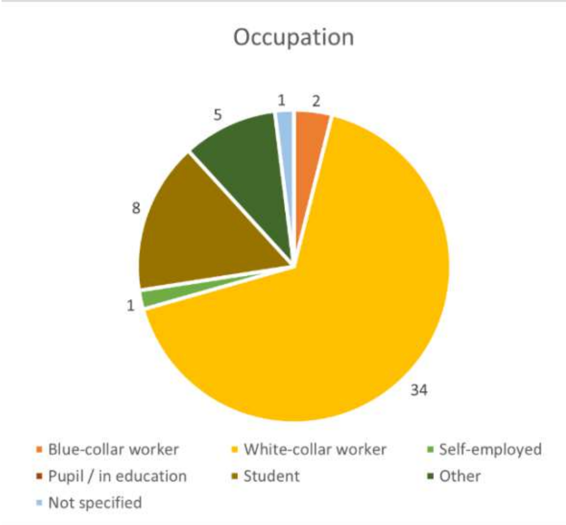

Figure 67: Occupation (n=51; own figure)

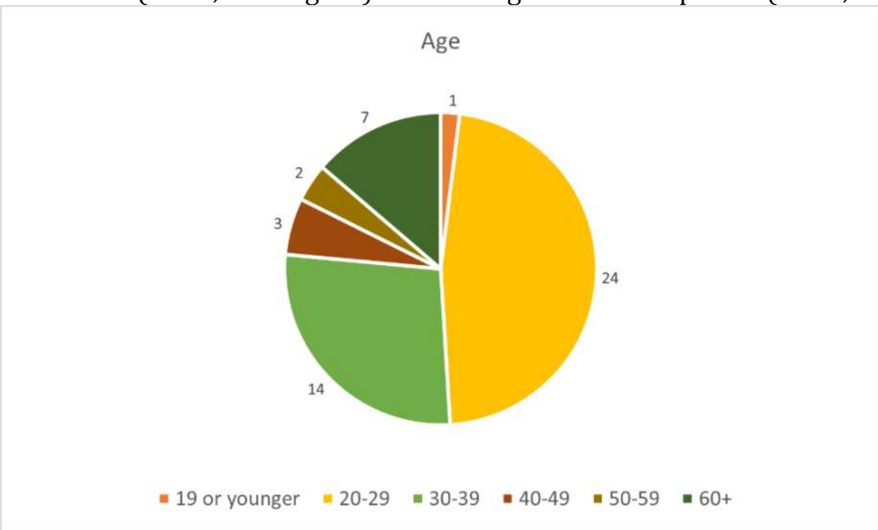

Figure 68: Age (n=51; own figure)

Experience

The majority of participants (98%) had never interacted with a cobot before; the one participant (2%) who had interacted with a cobot had only done so once (see Figure 69).

Furthermore, 35 participants (68.6%) had no coding skills at all, while nine participants considered themselves beginners, four intermediate, and three experts (see Figure 70).

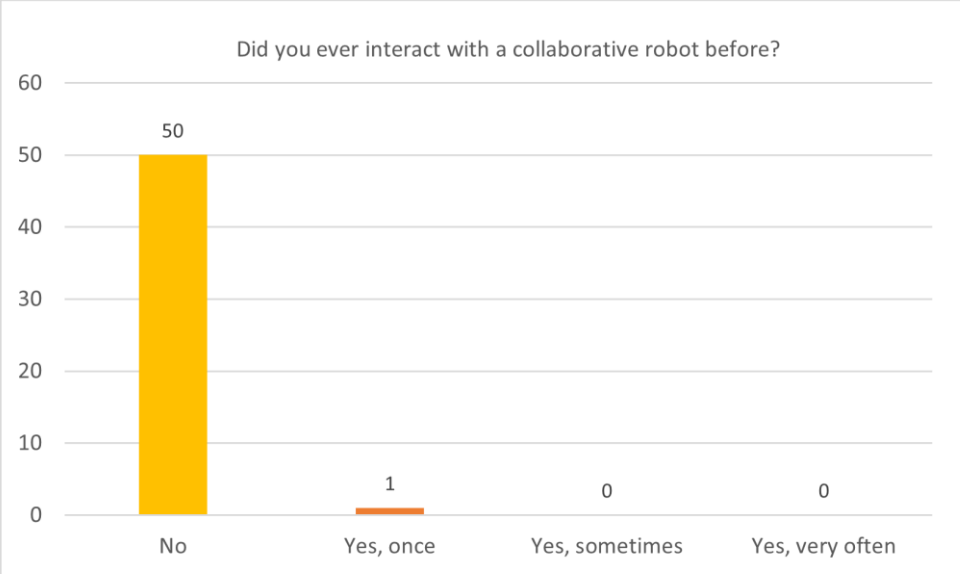

Figure 69: Interaction with collaborative robots (n=51; own figure)

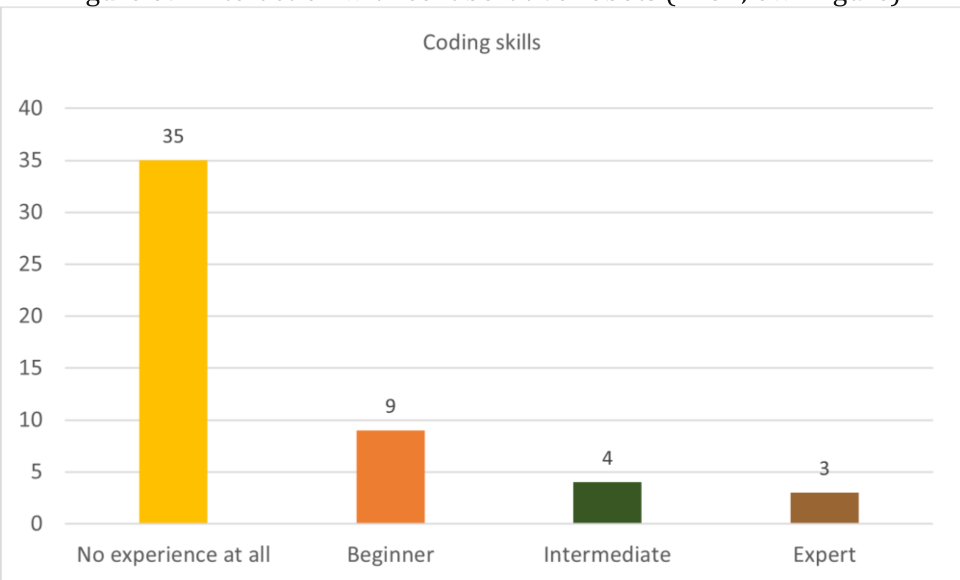

Figure 70: Coding skills (n=51; own figure)

Figure 71 shows that only 11 participants had sometimes worked with process modeling languages; 38 had no experience with them, and two had worked once with process modeling languages. Further, two out of the 11 participants, who had experience with process modeling had only some experience with BPMN. Most participants (49) had no experience at all with BPMN (see Figure 72).

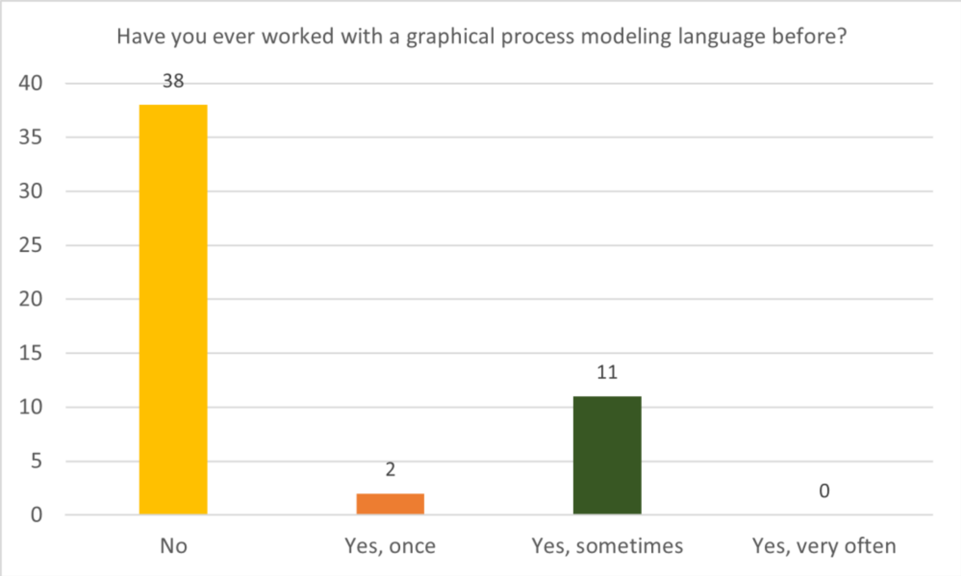

Figure 71: Experience with graphical process modeling languages (n=51; own figure)

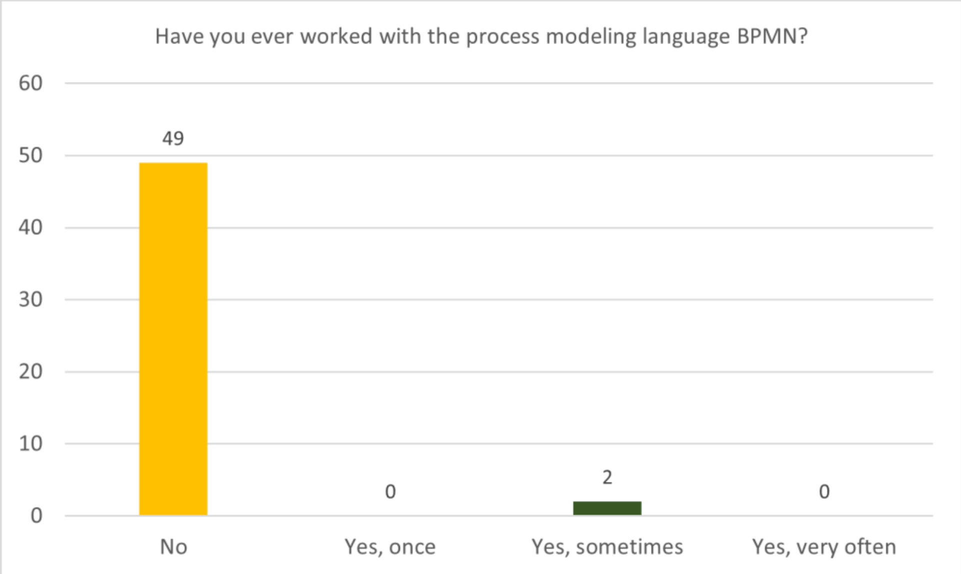

Figure 72: Experience with BPMN (n=51; own figure)

5.3 Results

5.3.1 Correct Solving of the Task

Figure 73 shows that 41 participants (80.4%) solved the task correctly, and 10 participants (19.6%) did not enter the correct value in the questionnaire. These wrong solutions were further analyzed. Two participants did not enter a value for the first question, resulting in null values in the database for this entry. During the practical evaluation, one participant entered an illegal character in an input field while performing the personalization task. This led to an error message in the dashboard at the "Executions

Until Learned" element. Because the participant did not delete this illegal character, the displayed error message was entered in the questionnaire instead of the correct value.

Three of the seven remaining results were incorrect by one value (e.g., instead of the correct answer 50, the result was 51), while the remaining four were entirely incorrect values. The completely incorrect values most likely occurred when the participants did not click on the first task, but it is difficult to know in detail why the wrong answers were given.

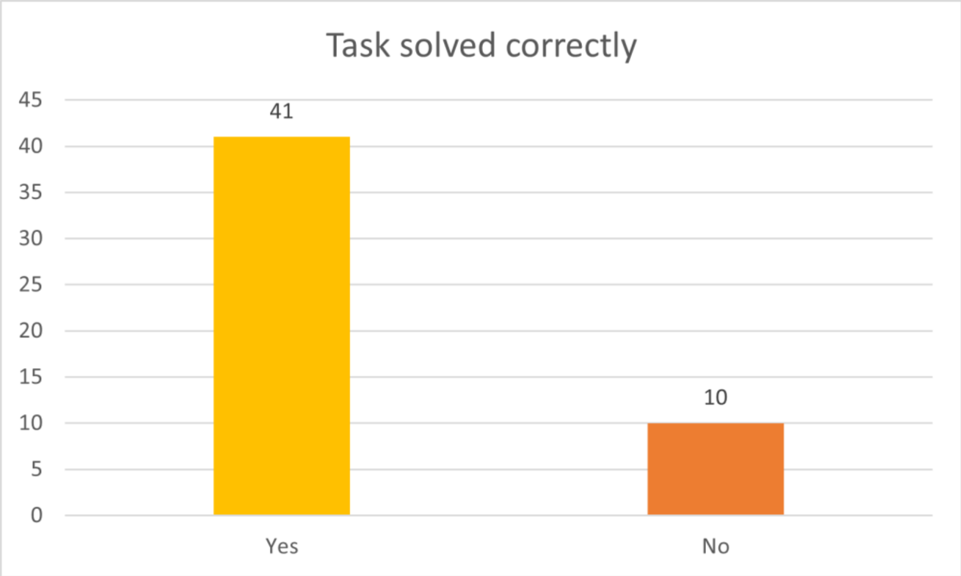

Figure 73: Correct/no correct solution of task (n=51; own figure)

5.3.2 Usability/User Experience

System Usability Scale (SUS)

The System Usability Scale (SUS) developed by John Brooke [10] was used to evaluate the usability of the proposed solution. The achieved score of the system was 76 (see Figure 74).

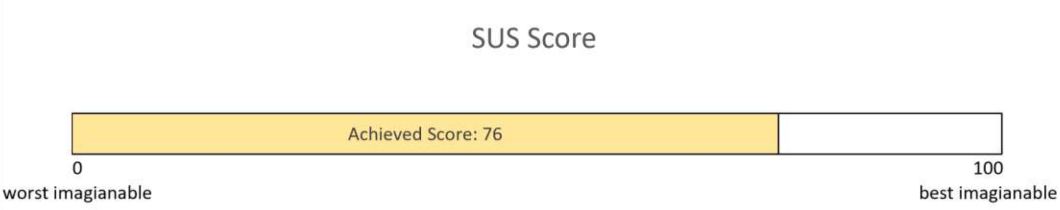

Figure 74: System Usability Score result (own figure)

At the base of this work is the BPMN prototype by Hader [6], which was also evaluated using a SUS so the scores could compared. The developed prototype by Hader received a score of 86, which shows that the usability of our prototype slightly decreased in comparison. This is most likely a result of the increased complexity.

An additional interpretation of the SUS score is needed to provide a more in-depth analysis. Bangor et al. [81] developed a classification for systems based on the SUS score that determines whether the system has an acceptable level of usability. Furthermore, Bangor et al. [82] introduced an adjective rating. Both classifications are displayed in Figure 75.

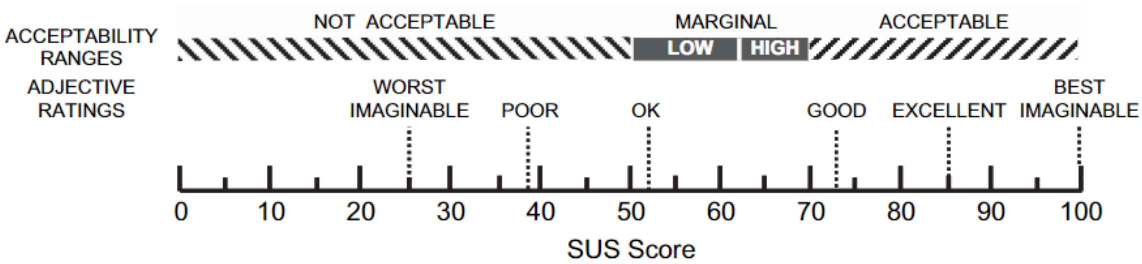

Figure 75: Acceptability ranges and adjective ratings of SUS scores [81]

Based on these classifications, it is apparent that the SUS score of 76 is considered acceptable as scores roughly above 70 are considered acceptable. Moreover, the score falls within the range of 73 to 85, which is classified as "good."

User Experience Questionnaire (UEQ)

For analyzing the outcomes of the User Experience Questionnaire, the UEQ Data Analysis Tool V10 was used, which can be downloaded from www.ueq-online.org. In the first step, we used the tool to check for inconsistencies in the answers. In fact, one data line showed inconsistencies across all scales, which indicates that the answers of this participant were given randomly or not seriously. Therefore, this data line was removed, which resulted in a total participant number of 50 for the UEQ.

The results of the UEQ are six mean values, where the scales can range from -3 to +3 as follows: -3 represents the most negative answer, +3 the most positive answer, and 0 a neutral answer. However, it is rare to observe values above +2 or below -2. This results from the calculation of means over a large number of individuals with different opinions and answering tendencies, such as avoidance of extremes. [83]

The results of the UEQ of this evaluation are displayed in Figure 76.

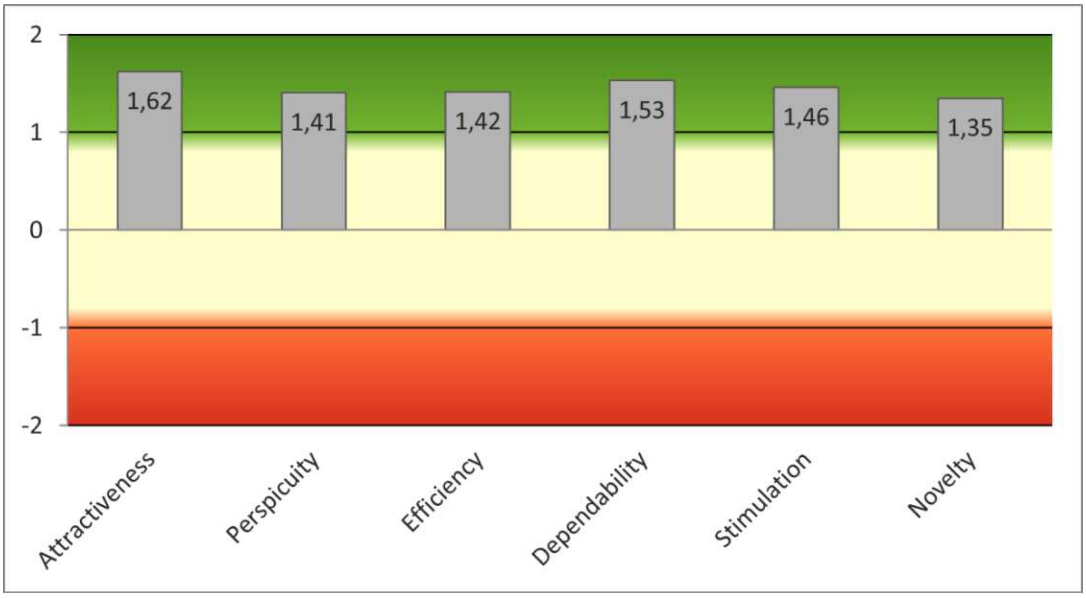

Figure 76: Result of UEQ (n=50; own screenshot)

What do these values mean? Generally, values from -0.8 to +0.8 indicate a more or less neutral evaluation of the corresponding scale, whereas values greater than 0.8 indicate a positive evaluation, and values less than -0.8 indicate a negative evaluation [84]. Therefore, the result of the evaluation indicates a good user experience.

Additionally, it is necessary to determine if there is sufficient UX [32].

It would be an easy task if there existed a previous version to which it could be compared using a statistical test [32]. In our case, we did not have an older version for comparison. For such purposes, a benchmark for the UEQ is provided [32]. When an evaluated system has a high score compared to the benchmark, it indicates sufficient UX [32].

Therefore, intervals with adjectives are provided, as displayed in Table 25.

Table 25: Benchmark intervals for UEQ [32]

	Att.	**Eff.**	**Per.**	**Dep.**	**Stim.**	**No**
Excellent	≥ 1.75	≥ 1.78	≥ 1.9	≥ 1.65	≥ 1.55	≥ 1.4
Good	≥ 1.52 < 1.75	≥ 1.47 < 1.78	≥ 1.56 < 1.9	≥ 1.48 < 1.65	≥ 1.31 < 1.55	≥ 1.05 < 1.4
Above Average	≥ 1.17 < 1.52	≥ 0.98 < 1.47	≥ 1.08 < 1.56	≥ 1.14 < 1.48	≥ 0.99 < 1.31	≥ 0.71 < 1.05
Below Average	≥ 0.7 < 1.17	≥ 0.54 < 0.98	≥ 0.64 < 1.08	≥ 0.78 < 1.14	≥ 0.50 < 0.99	≥ 0.3 < 0.71
Bad	< 0.7	< 0.54	< 0.64	< 0.78	< 0.5	< 0.3

The results of this evaluation compared to the benchmark along with interpretations are shown in Table 26:

Table 26: UEQ benchmark and interpretations

Scale	Mean	Comparison to benchmark	Interpretation
Attractiveness	1.62	**Good**	10% of results better, 75% of results worse
Perspicuity	1.41	**Above Average**	25% of results better, 50% of results worse
Efficiency	1.42	**Above Average**	25% of results better, 50% of results worse
Dependability	1.53	**Good**	10% of results better, 75% of results worse
Stimulation	1.46	**Good**	10% of results better, 75% of results worse
Novelty	1.35	**Good**	10% of results better, 75% of results worse

A graphical representation of the results is shown in Figure 77.

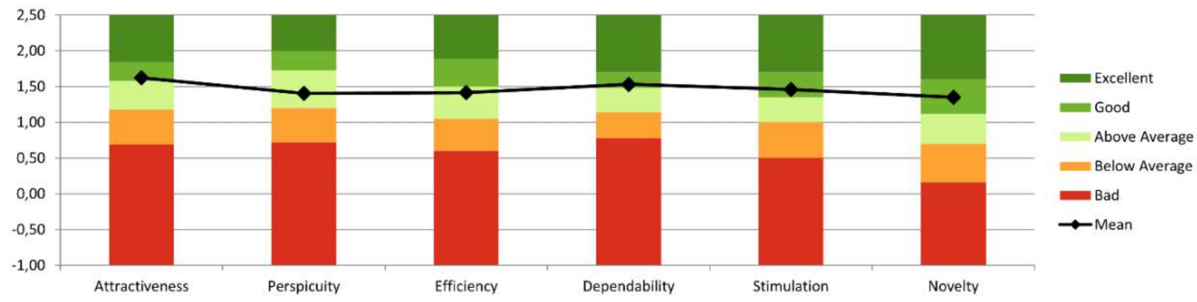

Figure 77: Results of UEQ in comparison to benchmark (own screenshot)

As can be seen, four scales are labeled "good", and two are labeled "above average”. Only “perspicuity” falls in the middle of the above-average category, while the efficiency scale is at the upper boundary of the above-average category. Hence, the comparison to the benchmark exhibits sufficient UX for the system.

In summary, the result of the usability evaluation using the SUS indicates acceptable and good usability for the prototype. Furthermore, the UEQ results for evaluating the user experience are positive and achieved a high score compared to the benchmark, which indicates the UX is sufficient. **Based on this evaluation, hypothesis H1 is supported.**

5.3.3 Qualitative Data

Analyzing the two open questions showed that most participants had no issues with using the tool.

Very few participants provided feedback that they did not clearly understand the tool's intended purpose or did not understand the provided example process. This might be because the study was conducted online, where participants could not ask questions if something was unclear. Despite the introduction videos and explanations given for this lack of feedback regarding not understanding the tool’s purpose, it cannot be completely ruled out that some misunderstandings occurred. The total time needed to read the instructions and watch the videos was about 10 to 15 minutes. This was the most extensive explanation that could be given while still keeping the study as brief as possible.

Another participant was overwhelmed by the functions of the system, while another was insecure because of functions in the system that were not needed during the evaluation.

One participant detected a spelling error in the instructions. Another mentioned that explanatory texts in pop-up windows would be helpful. Some issued feedback that the estimated time was too short.

Furthermore, the majority of comments were positive for example the following (these answers were given in German during the evaluation and translated for this section):

- “Nice color scheme and very clearly arranged.”
- “Very creative and neatly done!”
- “Process management solved extremely creatively and effectively.”
- “Very well done :)”
- “Very creative matter with a lot of future potential in many areas and industries.”

6 Conclusion, Discussion, and Outlook

6.1 Conclusion

The main problem is that although the existing prototype enables adaptive task sharing [3], it lacks in providing learning opportunities and decision-support information. In this technologically dominated world, learning is paramount for workers. Furthermore, the worker should be provided with information about the effects of allocating tasks. Therefore, this thesis aimed to enhance the existing prototype to provide decision support and consider workplace learning in combination with task allocation.

The research questions to be answered in this work were as follows:

- RQ1: Which approach is suitable to realize adaptive task sharing between a human and robot considering learning opportunities regarding learning new tasks?
- RQ2: What is a suitable design for a worker assistance system to improve decision support for adaptive task sharing and enable workplace learning?
- RQ3: What is a user-friendly way to implement this approach?

Based on the literature review in Chapter 3, no work currently combines workplace learning with task allocation between a human and a robot. Thus, a learning curve model and forecasting for learning times for industrial settings is needed. Therefore, we proposed integrating a learning curve model and Jeske's devised prediction method for sensorimotor tasks [55] to the existing prototype [3] and add decision-support information.

This resulted in the introduction of task-level parameters and the dashboard, which allow the worker to consider various factors when determining task allocation. The worker can select time, costs, and learning as decision-making criteria. Furthermore, it is possible to highlight the fastest and cheapest process allocation possible.

The final prototype (Figure 78) enables:

- An overview of the current costs and execution times of the process as well as for each agent,
- Easier decision-making regarding costs and execution times (fastest/cheapest process highlighting), and
- Learning opportunities, clarity of the current skill level, and predicting the time and number of executions to learn a task.

The design science research methodology by Peffers [7] was used to conduct the research. The prototype was implemented in multiple iterations, including a heuristic evaluation performed by experts. A detailed description of the implementation process can be found in Chapter 4.

A crucial part of the work was the user-friendliness evaluation. An online user study was performed using the SUS and UEQ to conduct the evaluation. The evaluation results showed that the final prototype is considered user friendly.

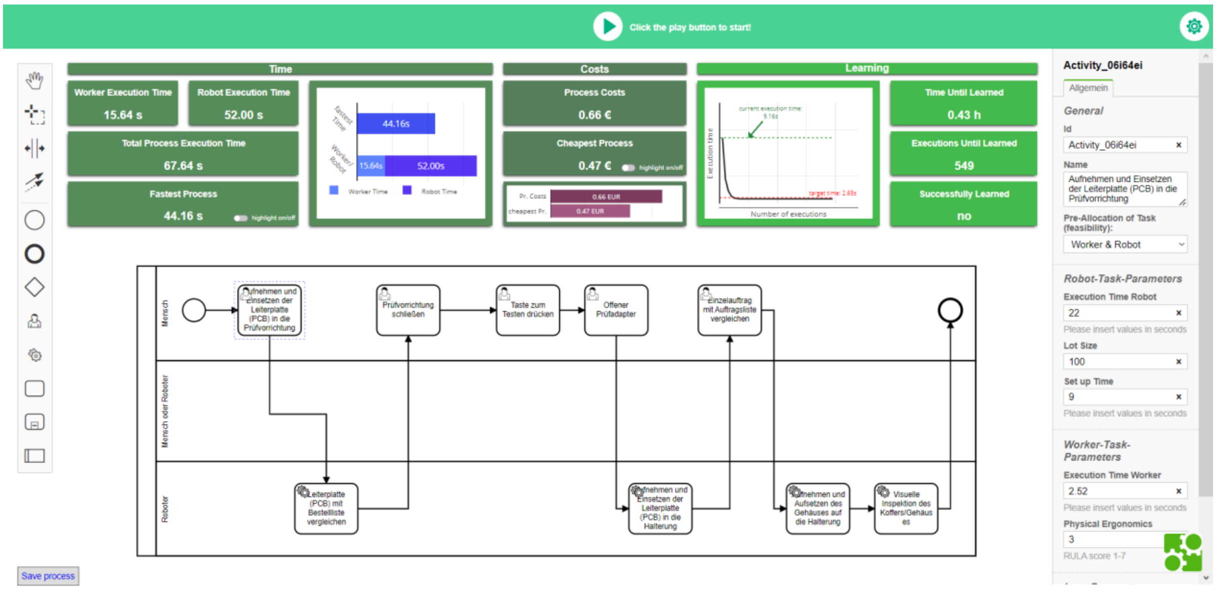

Figure 78: Final GUI of the prototype (own screenshot)

6.2 Discussion and Outlook

One limitation of this thesis is that the user study was conducted in an online format due to the coronavirus pandemic. For the evaluation, there was no cobot present. Therefore, the evaluation could be done in person in the future and compared to the findings in this thesis.

Additionally, the demographic data analysis showed that the main participants were employees, students, and self-employed. Only two workers participated in the user study. Therefore, a study should be conducted among this user group in the future.

Within the heuristic evaluation, some recommendations by the experts were out of this thesis's scope or not possible to implement due to a restricted time frame. Future research ideas derived from these recommendations are as follows:

- **Upgrade the BPMN.io version**: The BPMN.io version could be upgraded to the newest release. This would enable new functionalities such as a save/load button, zoom in/out button, and version control, enabling undo and redo operations.
- **Work instruction parameter for learning curve model at task level**: The parameter for the learning curve model used in this thesis considers the work instruction type. This can only be changed globally. If a process has multiple work instructions for different tasks, it would be more precise if the parameter could be changed at the task level.
- **Animation of the robot:** To illustrate the whole system and the individual steps, an animation of the robot would help people with little experience better understand the robot and the system.
- **Beginner/expert mode**: A beginner mode could be introduced, that hides currently unneeded blocks. A switch to expert mode needs to be possible. This could also be an optional feature for every area, allowing hiding areas with a button click.
- **Help menu/documentation:** This would help users when they need to understand a function of the software. This could be in the form of documentation or with individual question-mark buttons in selected areas.

Other possible future research ideas are as follows:

- **User management:** Add user management that enables the worker to save/load their preferences.
- **Further criteria:** Additional criteria, such as physical/cognitive workload, could be added as decision support for the allocation of a task based on these criteria.
- **Change from offline to online task allocation:** Make it possible to change the task allocation during the execution to be able to react to occurrences when the process is executed.

7 List of References

[1] J. Zimmermann, *Collaborative robots (COBOTS): Safe co-operation between human beings and robots.* [Online]. Available: https://www.dguv.de/ifa/fachinfos/kollaborierende-roboter/index-2.jsp (accessed: Sep. 7 2021).

[2] C. Schmidbauer, B. Hader, and S. Schlund, "Evaluation of a Digital Worker Assistance System to enable Adaptive Task Sharing between Humans and Cobots in Manufacturing," 2021 (forthcoming).

[3] C. Schmidbauer, S. Schlund, T. B. Ionescu, and B. Hader, "Adaptive Task Sharing in Human-Robot Interaction in Assembly," 2020, doi: 10.1109/IEEM45057.2020.9309971.

[4] J. Bischoff *et al.,* "Erschließen der Potenziale der Anwendung von Industrie 4.0 im Mittelstand," Mülheim an der Ruhr.

[5] H.-J. Buxbaum, S. Sen, and R. Häusler, "Theses on the Future Design of Human-Robot Collaboration," DOI: 10.1007/978-3-030-49062-1_38, 2020, doi: 10.1007/978-3-030-49062-1_38.

[6] B. Hader, *Intuitive programming of collaborative human robot processes*. Wien, 2021. [Online]. Available: http://media.obvsg.at/AC16178411-2001

[7] Ken Peffers, Tuure Tuunanen, Charles E. Gengler, and Matti Rossi, *The design science research process: a model for producing and presenting information systems research*, 2006. [Online]. Available: https://www.researchgate.net/publication/238077290_The_design_science_research_process_a_model_for_producing_and_presenting_information_systems_research

[8] B. Kitchenham and S. Charters, "Guidelines for performing Systematic Literature Reviews in Software Engineering," vol. 2, 2007.

[9] Hevner, March, Park, and Ram, "Design Science in Information Systems Research," *MIS Quarterly*, vol. 28, no. 1, p. 75, 2004, doi: 10.2307/25148625.

[10] J. Brooke, "SUS: A quick and dirty usability scale," *Usability Eval. Ind.*, vol. 189, 1995.

[11] M. Rauschenberger, M. Schrepp, and J. Thomaschewski, *User Experience mit Fragebögen messen – Durchführung und Auswertung am Beispiel des UEQ*: German UPA e.V, 2013. [Online]. Available: https://dl.gi.de/handle/20.500.12116/5653

[12] M. A. Goodrich and A. C. Schultz, "Human-Robot Interaction: A Survey," *FNT in Human-Computer Interaction*, vol. 1, no. 3, pp. 203–275, 2007, doi: 10.1561/1100000005.

[13] T. Fong, C. Thorpe, and C. Baur, "Collaboration, Dialogue, Human-Robot Interaction," in *Robotics Research*: Springer, Berlin, Heidelberg, 2003, pp. 255–266. [Online]. Available: https://link.springer.com/chapter/10.1007/3-540-36460-9_17

[14] Linda Onnasch, Xenia Maier, and Thomas Jürgensohn, "Mensch-Roboter-Interaktion - Eine Taxonomie für alle Anwendungsfälle," Dortmund, 2016.

[15] A. Tausch, “Aufgabenallokation in der Mensch-Roboter-Interaktion: Eine psychologische Betrachtung von Aufgabenzuteilungs-Prozessen zur Gestaltung menschengerechter Zusammenarbeit von Mensch und Roboter,” Ruhr-Universität Bochum, 2021.

[16] E. Matheson, R. Minto, E. G. G. Zampieri, M. Faccio, and G. Rosati, “Human–Robot Collaboration in Manufacturing Applications: A Review,” *Robotics*, vol. 8, no. 4, p. 100, 2019, doi: 10.3390/robotics8040100.

[17] F. Ranz, V. Hummel, and W. Sihn, “Capability-based Task Allocation in Human-robot Collaboration,” *Procedia Manufacturing*, vol. 9, pp. 182–189, 2017, doi: 10.1016/j.promfg.2017.04.011.

[18] J. Edward Colgate, Witaya Wannasuphoprasit, and Michael A. Peshkin, “Cobots: robots for collaboration with human operators,” (in English (US)), *Proceedings of the 1996 ASME International Mechanical Engineering Congress and Exposition*, 1995. [Online]. Available: https://www.scholars.northwestern.edu/en/publications/cobots-robots-for-collaboration-with-human-operators

[19] Christina Schmidbauer, *Adaptive Task Sharing between Humans and Cobots in Assembly*, 2022.

[20] A. Vysocky and P. Novak, “Human-Robot Collabroation in Industry,” *MM SJ*, vol. 2016, no. 02, pp. 903–906, 2016, doi: 10.17973/MMSJ.2016_06_201611.

[21] M. Knudsen and J. Kaivo-Oja, “Collaborative Robots: Frontiers of Current Literature,” *Journal of Intelligent Systems: Theory and Applications*, pp. 13–20, 2020, doi: 10.38016/jista.682479.

[22] W. Bauer, M. Bender, M. Braun, P. Rally, and O. Scholtz, *Leichtbauroboter in der manuellen Montage – einfach EINFACH anfangen.: Erste Erfahrungen von Anwenderunternehmen*, 2016.

[23] R. Challenger, C. W. Clegg, and C. Shepherd, “Function allocation in complex systems: reframing an old problem,” *Ergonomics*, vol. 56, no. 7, pp. 1051–1069, 2013, doi: 10.1080/00140139.2013.790482.

[24] F. Ansari, P. Hold, W. Mayrhofer, S. Schlund, and W. Sihn, “AUTODIDACT: Introducing the Concept of Mutual Learning into a Smart Factory Industry 4.0,” *International Association for Development of the Information Society*, 2018. [Online]. Available: https://eric.ed.gov/?id=ED600612

[25] *Arbeitskosten - STATISTIK AUSTRIA - Die Informationsmanager.* [Online]. Available: https://www.statistik.at/statistiken/arbeitsmarkt/arbeitskosten-und-tariflohnindex/arbeitskosten/arbeitskosten (accessed: Dec. 1 2022).

[26] D. Schröter, “Entwicklung einer Methodik zur Planung von Arbeitssystemen in Mensch-Roboter-Kooperation,” 2018.

[27] L. McAtamney and E. Nigel Corlett, “RULA: a survey method for the investigation of work-related upper limb disorders,” *Applied ergonomics*, vol. 24, no. 2, pp. 91–99, 1993, doi: 10.1016/0003-6870(93)90080-s.

[28] Bpmn.io, *Web-based tooling for BPMN, DMN, CMMN, and Forms | bpmn.io.* [Online]. Available: https://bpmn.io/ (accessed: Nov. 29 2022).

[29] Camunda, *The Universal Process Orchestrator | Camunda.* [Online]. Available: https://camunda.com/ (accessed: Nov. 29 2022).

[30] J. Nielsen, "CHAPTER 1 - What Is Usability?," in *The Morgan Kaufmann series in interactive technologies, User Experience Re-Mastered: Your Guide to Getting the Right Design*, C. Wilson, Ed., 1st ed., s.l.: Elsevier monographs, 2009, pp. 3–22. [Online]. Available: https://www.sciencedirect.com/science/article/pii/B9780123751140000049

[31] P. Christensson, *User-Friendly Definition.* [Online]. Available: https://techterms.com/definition/user-friendly (accessed: Feb. 9 2022).

[32] Martin Schrepp, Jörg Thomaschewski, and Andreas Hinderks, "Construction of a Benchmark for the User Experience Questionnaire (UEQ)," *International Journal of Interactive Multimedia and Artificial Intelligence*, vol. 4, Regular Issue, 2017. [Online]. Available: https://www.ijimai.org/journal/bibcite/reference/2604

[33] H. M. Hassan and G. H. Galal-Edeen, "From usability to user experience," in *2017 International Conference on Intelligent Informatics and Biomedical Sciences (ICIIBMS)*, 2017, pp. 216–222.

[34] J. S. Dumas and J. C. Redish, *A practical guide to usability testing*. Exeter: Intellect, 1999. [Online]. Available: https://books.google.at/books?id=4lge5k_F9EwC

[35] Jakob Nielsen, *Usability 101: Introduction to Usability.* [Online]. Available: https://www.nngroup.com/articles/usability-101-introduction-to-usability/ (accessed: Sep. 24 2022).

[36] *Ergonomics of human-system interaction - Part 11: Usability: Definitions and concepts (ISO 9241-11:2018); German version,* International Organisation of Standardisation (ISO), Berlin.

[37] Nielsen Norman Group, *The Definition of User Experience (UX).* [Online]. Available: https://www.nngroup.com/articles/definition-user-experience/ (accessed: Sep. 22 2022).

[38] *Ergonomics of human-system interaction – Part 210: Human-centred design for interactive systems (ISO 9241-210:2019); German version,* International Organisation of Standardisation (ISO), Berlin.

[39] M. Schrepp, A. Hinderks, and J. Thomaschewski, "Applying the User Experience Questionnaire (UEQ) in Different Evaluation Scenarios," in *Lecture Notes in Computer Science*, vol. 8517, *Design, User Experience, and Usability. Theories, Methods, and Tools for Designing the User Experience*, D. Hutchison et al., Eds., Cham: Springer International Publishing, 2014, pp. 383–392.

[40] N. Sasmita and R. H. Kumar, "Exigency of re-skilling for organization and employees growth," *International Journal of Business*, pp. 65–67, 2018.

[41] Gartner Inc, *Gartner HR Research Finds 58% of the Workforce Will Need New Skill Sets to Do Their Jobs Successfully.* [Online]. Available: https://www.gartner.com/en/newsroom/press-releases/2021-02-03-gartner-hr-research-finds-fifty-eight-percent-of-the-workforce-will-need-new-skill-sets-to-do-their-jobs-successfully (accessed: Sep. 20 2022).
[42] T. Komenda, C. Schmidbauer, D. Kames, and S. Schlund, *Learning to Share - Teaching the Impact of Flexible Task Allocation in Human-cobot Teams*, 2021.
[43] W. Laurig, *Grundzüge der Ergonomie: Erkenntnisse und Prinzipien*.
[44] S. Nixdorf, F. Ansari, and W. Sihn, "Work-Based Learning in Smart Manufacturing: Current State and Future Perspectives," *Proceedings of the Conference on Learning Factories (CLF)*, 2021, doi: 10.2139/ssrn.3858379.
[45] C. Schlick, R. Bruder, and H. Luczak, *Arbeitswissenschaft,* 4th ed. Berlin, Heidelberg: Springer Berlin Heidelberg, 2018. [Online]. Available: http://nbn-resolving.org/urn:nbn:de:bsz:31-epflicht-1521821
[46] W. Rohmert, E. Ulich, and J. Rutenfranz, *Das Anlernen sensumotorischer Fertigkeiten*. Frankfurt am Main: Europäische Verl.-Anst, 1971.
[47] F. Kuhlenbäumer, P. Przybysz, S. Mütze-Niewöhner, and C. M. Schlick, "Age-Differentiated Analysis of the Influence of Task Descriptions on Learning Sensorimotor Tasks," in *Advances in Ergonomic Design of Systems, Products and Processes*: Springer, Berlin, Heidelberg, 2017, pp. 159–175. [Online]. Available: https://link.springer.com/chapter/10.1007/978-3-662-53305-5_12
[48] T. P. WRIGHT, "Factors Affecting the Cost of Airplanes," *Journal of the Aeronautical Sciences*, vol. 3, no. 4, pp. 122–128, 1936, doi: 10.2514/8.155.
[49] J. De Jong, "Die Auswirkungen zunehmender Fertigkeiten," *REFA Nachrichten*, pp. 155–161, 1960.
[50] W. L. Hieber, "Lern- und Erfahrungskurveneffekte und ihre Bestimmung in der flexibel automatisierten Produktion: Lernkurveneffekte / Controlling-Praxis," Zugl.: Stuttgart, Univ., Diss., 1990 München : Vahlen, München. [Online]. Available: https://permalink.catalogplus.tuwien.at/AC00411972
[51] M. de Greiff, *Die Prognose von Lernkurven in der manuellen Montage unter besonderer Berücksichtigung der Lernkurven von Grundbewegungen*. Zugl.: Duisburg, Univ.-Gesamthochsch., Diss., 2001. Düsseldorf: VDI-Verl., 2001.
[52] A. Laarmann, *Lerneffekte in der Produktion*. Zugl.: Bochum, Ruhr-Univ., Diss., 2005, 1st ed. Wiesbaden: Dt. Univ.-Verl., 2005.
[53] P. Hold, "Vorgehensmodell zur Planung und Evaluierung digitaler Assistenzsysteme in der Montage," *Vorgehensmodell zur Planu*, 2020. [Online]. Available: https://permalink.catalogplus.tuwien.at/AC15763980
[54] T. Jeske and C. M. Schlick, "A New Method for Forecasting the Learning Time of Sensorimotor Tasks," in *Advances in human factors and ergonomics series, Advances*

in ergonomics in manufacturing, S. Trzcieliński and W. Karwowski, Eds., Boca Raton, Fla: CRC Press, 2012, pp. 241–250.
[55] T. Jeske, C. M. Schlick, and S. Mütze-Niewöhner, “Unterstützung von Lernprozessen bei Montageaufgaben,” in *Flexible Produktionskapazität innovativ managen*, C. M. Schlick, K. Moser, and M. Schenk, Eds., Berlin, Heidelberg: Springer Berlin Heidelberg, 2014, pp. 163–192.
[56] T. Jeske, *Entwicklung einer Methode zur Prognose der Anlernzeit sensumotorischer Tätigkeiten*. Zugl.: Aachen, Techn. Hochsch., Diss., 2013. Aachen: Shaker, 2013.
[57] B. Aunkofer, “MTM – Universelles Analysier System,” *Der Wirtschaftsingenieur.de*, 04 May., 2012. https://www.der-wirtschaftsingenieur.de/index.php/mtm-universelles-analysier-system/ (accessed: Aug. 7 2023).
[58] E. A. Fleishman, “A factor analysis of fine manipulative tests,” *Journal of applied psychology*, vol. 46, no. 2, pp. 96–105, 1962.
[59] C. Peña, D. Romero, and J. Noguez, “Workforce Learning Curves for Human-Based Assembly Operations: A State-of-the-Art Review,” *Applied Sciences*, vol. 12, no. 19, p. 9608, 2022, doi: 10.3390/app12199608.
[60] C. Petzoldt, M. Harms, and M. Freitag, “Review of task allocation for human-robot collaboration in assembly,” *International Journal of Computer Integrated Manufacturing*, pp. 1–41, 2023, doi: 10.1080/0951192X.2023.2204467.
[61] T. Bänziger, A. Kunz, and K. Wegener, “Optimizing human–robot task allocation using a simulation tool based on standardized work descriptions,” (in En;en), *J Intell Manuf*, vol. 31, no. 7, pp. 1635–1648, 2020, doi: 10.1007/s10845-018-1411-1.
[62] P. Tsarouchi, G. Michalos, S. Makris, T. Athanasatos, K. Dimoulas, and G. Chryssolouris, “On a human–robot workplace design and task allocation system,” *International Journal of Computer Integrated Manufacturing*, vol. 30, no. 12, pp. 1272–1279, 2017, doi: 10.1080/0951192X.2017.1307524.
[63] A. Pupa and C. Secchi, “A Safety-Aware Architecture for Task Scheduling and Execution for Human-Robot Collaboration,” in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2021, pp. 1895–1902.
[64] L. Johannsmeier and S. Haddadin, “A Hierarchical Human-Robot Interaction-Planning Framework for Task Allocation in Collaborative Industrial Assembly Processes,” *IEEE Robotics and Automation Letters*, vol. 2, no. 1, pp. 41–48, 2017, doi: 10.1109/LRA.2016.2535907.
[65] C. Petzoldt, D. Niermann, E. Maack, M. Sontopski, B. Vur, and M. Freitag, “Implementation and Evaluation of Dynamic Task Allocation for Human–Robot Collaboration in Assembly,” *Applied Sciences*, vol. 12, no. 24, p. 12645, 2022, doi: 10.3390/app122412645.
[66] I. E. Makrini, K. Merckaert, J. de Winter, D. Lefeber, and B. Vanderborght, “Human Robot Collaborative Intelligence,” *Interaction Studies*, vol. 20, no. 1, pp. 102–133, 2019, doi: 10.1075/is.18018.mak.

[67] M. Cramer, K. Kellens, and E. Demeester, “Probabilistic Decision Model for Adaptive Task Planning in Human-Robot Collaborative Assembly Based on Designer and Operator Intents,” *IEEE Robotics and Automation Letters*, vol. 6, no. 4, pp. 7325–7332, 2021, doi: 10.1109/LRA.2021.3095513.

[68] C. H. Glock, E. H. Grosse, M. Y. Jaber, and T. L. Smunt, “Applications of learning curves in production and operations management: A systematic literature review,” *Computers & Industrial Engineering*, vol. 131, pp. 422–441, 2019, doi: 10.1016/j.cie.2018.10.030.

[69] J. Tilindis and V. Kleiza, “Learning curve parameter estimation beyond traditional statistics,” *Applied Mathematical Modelling*, vol. 45, pp. 768–783, 2017, doi: 10.1016/j.apm.2017.01.025.

[70] J. R. Crawford, “Learning curve, ship curve, ratios, related data,” *Lockheed Aircraft Corporation*, vol. 144, 1944.

[71] A. Tamás and T. Koltai, “Application of Learning Curves in Operations Management Decisions,” *Period. Polytech. Soc. Man. Sci.*, vol. 28, no. 1, pp. 81–90, 2020, doi: 10.3311/PPso.14136.

[72] Y. Gao, U. Kruger, X. Intes, S. Schwaitzberg, and S. De, “A machine learning approach to predict surgical learning curves,” *Surgery*, vol. 167, no. 2, pp. 321–327, 2020, doi: 10.1016/j.surg.2019.10.008.

[73] D. Hogan, J. Elshaw, C. Koschnick, J. Ritschel, A. Badiru, and S. Valentine, “Cost Estimating Using a New Learning Curve Theory for Non-Constant Production Rates,” *Forecasting*, vol. 2, no. 4, pp. 429–451, 2020, doi: 10.3390/forecast2040023.

[74] Evan R. Boone, A. Badiru, C. Koschnick, John J. Elshaw, and J. Ritschel, “A Learning Curve Model Accounting for the Flattening Effect in Production Cycles,” 2021. [Online]. Available: https://www.semanticscholar.org/paper/A-Learning-Curve-Model-Accounting-for-the-Effect-in-Ritschel-Badiru/4466911160bb325bc815c7161ff6acabe9a1e3f7

[75] S. Di Luozzo, G. R. Pop, and M. M. Schiraldi, “The Human Performance Impact on OEE in the Adoption of New Production Technologies,” *Applied Sciences*, vol. 11, no. 18, p. 8620, 2021, doi: 10.3390/app11188620.

[76] Plotly.js, *Plotly JavaScript Open Source Graphing Library.* [Online]. Available: https://plotly.com/javascript/ (accessed: Feb. 22 2023).

[77] J. Nielsen, “Finding usability problems through heuristic evaluation,” in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, Monterey, California, United States, 1992, pp. 373–380.

[78] J. Nielsen, *Ten usability heuristics.* [Online]. Available: https://www.nngroup.com/articles/ten-usability-heuristics/ (accessed: Jan. 20 2023).

[79] H. A. Frijns and C. Schmidbauer, “Design Guidelines for Collaborative Industrial Robot User Interfaces,” in *Lecture Notes in Computer Science, Human-Computer*

Interaction – INTERACT 2021, C. Ardito et al., Eds., Cham: Springer International Publishing, 2021, pp. 407–427.

[80] *Understanding Success Criterion 1.4.3: Contrast (Minimum).* [Online]. Available: https://www.w3.org/WAI/WCAG21/Understanding/contrast-minimum.html (accessed: May 15 2023).

[81] A. Bangor, P. T. Kortum, and J. T. Miller, “An Empirical Evaluation of the System Usability Scale,” *International Journal of Human–Computer Interaction*, vol. 24, no. 6, pp. 574–594, 2008, doi: 10.1080/10447310802205776.

[82] Aaron Bangor, Philip Kortum, and James T. Miller, “Determining what individual SUS scores mean: adding an adjective rating scale,” *Journal of Usability Studies archive*, 2009. [Online]. Available: https://www.scinapse.io/papers/172711948

[83] Maria Rauschenberger, Martin Schrepp, Manuel Perez-Cota, Siegfried Olschner, and Jörg Thomaschewski, “Efficient Measurement of the User Experience of Interactive Products. How to use the User Experience Questionnaire (UEQ).Example: Spanish Language Version,” *International Journal of Interactive Multimedia and Artificial Intelligence*, vol. 2, Special Issue on Artificial Intelligence and Social Application, 2013. [Online]. Available: https://www.ijimai.org/journal/bibcite/reference/2403

[84] Martin Schrepp, “User Experience Questionnaire Handbook,” 2015.

8 List of Figures

Figure 1: User Interface of existing prototype for ATS (own screenshot) 1
Figure 2: Draft of expected user interface (own figure) 3
Figure 3: DSRM process model [7] 4
Figure 4: Different collaborative robots from different manufacturers [20] 9
Figure 5: Definition of the level of collaboration by Bauer et al. [22] 9
Figure 6: Static task allocation vs adaptive task sharing [3] 10
Figure 7: UI of the prototype composed of the three areas[2] 13
Figure 8: ATS prototype architecture [3] 14
Figure 9: Scale structure of the dimensions [39] 17
Figure 10: Schematic representation of learning time [47] 19
Figure 11: Representation of the learning curve and the model parameters to be determined [56] 20
Figure 12: Categorization of task allocation approaches [60] 25
Figure 13: Reference model for task allocation [60] 27
Figure 14: Process flow editor for creation of the assembly process [65] 29
Figure 15: Process flow viewer for execution of the assembly process [65] 29
Figure 16: Trend line of publications in the area of learning curves in production and operations management [68] 30
Figure 17: Initial state of the prototype[6] 34
Figure 18: Additional information for input field “Execution Time Worker” (own screenshot) 35
Figure 19: Additional information for input field “Set up Time” (own screenshot) 35
Figure 20: Old parameter panel (own screenshot) 36
Figure 21: New parameter panel (own screenshot) 36
Figure 22: Input field with error #1 (own screenshot) 37
Figure 23: Input field with error #2 (own screenshot) 37
Figure 24: Input field with error #3 (own screenshot) 37
Figure 25: Input field with error #4 (own screenshot) 37
Figure 26: UI with added dashboard (own screenshot) 38
Figure 27: Time area of the dashboard (own screenshot) 38
Figure 28: Costs area of the dashboard (own screenshot) 39
Figure 29: Learning area of the dashboard (own screenshot) 40
Figure 30: User properties (own screenshot) 41
Figure 31: User properties with advanced options visible (own screenshot) 41
Figure 32: Error message for an unfeasible task allocation (own screenshot) 42
Figure 33: Example of a highlighted task, that is allocated to an agent for whom it is not feasible to carry out (own screenshot) 42
Figure 34: UI during execution of the process (own screenshot) 43

Figure 35: UI with error message if process is not executable (own screenshot) 43
Figure 36: Highlighted tasks for fastest execution time (own screenshot) 44
Figure 37: Fastest process allocation (own screenshot) 44
Figure 38: Highlighted tasks for cheapest execution (own screenshot) 45
Figure 39: Cheapest process allocation (own screenshot) 45
Figure 40: UI with DP 1–4 highlighted (own screenshot) 46
Figure 41: Label if process is not currently being executed (own screenshot) 50
Figure 42: Label if process is currently being executed (own screenshot) 50
Figure 43: Dashboard does not load on start (own screenshot) 50
Figure 44: Dashboard loads on start (own screenshot) 50
Figure 45: Color of cost section before (own screenshot) 51
Figure 46: Color of cost section after (own screenshot) 51
Figure 47: Final prototype (own screenshot) 52
Figure 48: BPMN process chart of example process (own screenshot) 54
Figure 49: Test case, cheapest and fastest process, zoomed out to see the whole process at once (own screenshot) 55
Figure 50: Test case - most expensive process allocation 56
Figure 51: Test case for learning time and executions until learned 57
Figure 52: Landing page (own screenshot) 59
Figure 53: Explanation of study (own screenshot) 59
Figure 54: Brief explanation of procedure (own screenshot) 59
Figure 55: Voluntariness, anonymity and data protection (own screenshot) 59
Figure 56: Videos of example task and description to solve the task (own screenshot) . 60
Figure 57: Written task description for the participants (own screenshot) 60
Figure 58: Prototype for evaluation (own screenshot) 61
Figure 59: First task - personalization of user properties (own screenshot) 61
Figure 60: Second task - highlight the fastest process (own screenshot) 61
Figure 61: Second task - allocation of the fastest process (own screenshot) 61
Figure 62: Third task - execution of the process (own screenshot) 62
Figure 63: Fourth task - notation of value executions until learned (own screenshot) ... 62
Figure 64: Questionnaire - the first and second parts (own screenshot) 62
Figure 65: Questionnaire - the third part (own screenshot) 62
Figure 66: Gender (n=51; own figure) 65
Figure 67: Occupation (n=51; own figure) 65
Figure 68: Age (n=51; own figure) 65
Figure 69: Interaction with collaborative robots (n=51; own figure) 66
Figure 70: Coding skills (n=51; own figure) 66
Figure 71: Experience with graphical process modeling languages (n=51; own figure) . 67
Figure 72: Experience with BPMN (n=51; own figure) 67
Figure 73: Correct/no correct solution of task (n=51; own figure) 68

Figure 74: System Usability Score result (own figure) 68
Figure 75: Acceptability ranges and adjective ratings of SUS scores [81] 69
Figure 76: Result of UEQ (n=50; own screenshot) 70
Figure 77: Results of UEQ in comparison to benchmark (own screenshot) 71
Figure 78: Final GUI of the prototype (own screenshot) 74
Figure 79: Design guidelines for collaborative industrial robot user interfaces 1-12 [79] 91
Figure 80: Design guidelines for collaborative industrial robot user interfaces 13-24 [79] 92

9 Equation Directory

Equation 1 12
Equation 2 12
Equation 3 12
Equation 4 12
Equation 5 20
Equation 6 20
Equation 7 20
Equation 8 21
Equation 9 21
Equation 10 21
Equation 11 55

10 List of Tables

Table 1: Variables for economic efficiency calculation [19] 11
Table 2: RULA scores and recommended actions [27] 12
Table 3: Decision criteria for human suitability evaluation [19] 13
Table 4: Six dimensions for user experience [39] 17
Table 5: Keywords for task allocation between a human and robot 22
Table 6: Exclusion terms for task allocation between human and robot 23
Table 7: Search queries for task allocation between a human and robot 23
Table 8: Keywords for learning curves [59] 23
Table 9: Query string for learning curves in assembly operations [59] 24
Table 10: Search queries including the keyword "learning" 24
Table 11: Requirements and design parameters 34
Table 12: Groups and parameters of the properties panel 35
Table 13: Initial parameters of the properties panel 36
Table 14: Validation criteria for parameters 36
Table 15: Description of example guideline number 11 46
Table 16: Results of heuristic evaluation 47
Table 17: List of issues 48
Table 18: List of issues and respective recommendations 49
Table 19: Task list of assembly process [19] 52
Table 20: Task analysis of example process [19] 53
Table 21: Task assignment of example process [19] 54
Table 22: Economic efficiency evaluation [19] 54
Table 23: Most expensive task allocation 56
Table 24: Test case properties for learning prediction 56
Table 25: Benchmark intervals for UEQ [32] 70
Table 26: UEQ benchmark and interpretations 71

11 Appendix

11.1 Questionnaire

Fragebogen

Ergebnis der Aufgabe

1. Wie viele Ausführungen sind notwendig, dass die Aufgabe als gelernt angesehen werden kann?

Demografische Daten

1. Wie lässt sich Ihr derzeitiger Beruf einordnen?

- ○ ArbeiterIn
- ○ Angestellte(r)
- ○ Selbstständig
- ○ SchülerIn / in Ausbildung
- ○ StudentIn
- ○ Sonstige
- ○ keine Angabe

2. Was ist bislang Ihr höchster Bildungsabschluss?

- ○ Hauptschule, Unterstufe AHS, NMS, Sonderschule
- ○ Lehre
- ○ Berufsbildende Mittlere Schule (BMS) wie Handelsschule, Fachschulen
- ○ Matura
- ○ Uni/FH
- ○ keine Angabe

3. Bitte geben Sie Ihr Geschlecht an:

- ○ Weiblich
- ○ Männlich
- ○ Divers

4. Bitte geben Sie Ihre Altersgruppe an:

- ○ 19 oder jünger
- ○ 20 - 29
- ○ 30 - 39
- ○ 40 - 49
- ○ 50 - 59
- ○ 60+

5. Wie schätzen Sie sich in Bezug auf Technik ein?

	Stimme überhaupt nicht zu	Stimme nicht zu	weder noch	Stimme zu	Stimme voll zu
Ich habe ein gutes Verhältnis zu Technik und Maschinen.	○	○	○	○	○
Das Erlernen von neuen Technologien fällt mir leicht.	○	○	○	○	○
Ich weiß, wie man mit technischen Problemen umgeht.	○	○	○	○	○
Ich löse gerne technische Probleme, sehe dies als Herausforderung und habe Spaß daran.	○	○	○	○	○
Ich bin der Meinung, dass die meisten Technologien leicht zu erlernen sind.	○	○	○	○	○
In Bezug auf neue Technologien bin ich Up-to-date.	○	○	○	○	○

6. Haben Sie schon einmal mit einem kollaborativen Roboter gearbeitet?

- ○ Ja, sehr oft
- ○ Ja, einige Male
- ○ Ja, einmal
- ○ Nein

7. Bitte bewerten Sie Ihre Programmierkenntnisse anhand der folgenden Skala:

- ○ Experte (Ich beherrsche zumindest eine Programmiersprache auf einem Expertenlevel)
- ○ Fortgeschrittene Kenntnisse (Ich kann programmieren, bin aber kein Experte)
- ○ Anfänger (Ich habe einen Programmierkurs besucht und/oder verfüge über Basiskenntnisse)
- ○ Ich verfüge über keine Programmierkenntnisse

8. Haben Sie bereits mit einer grafischen Prozessmodellierungssprache wie zB UML Aktivitätsdiagrammen, Erweiterte Prozessketten, BPMN oder Flussdiagrammen gearbeitet?

- ○ Ja, sehr oft
- ○ Ja, einige Male
- ○ Ja, einmal
- ○ Nein

Bewertung der Usability des getesteten Systems

1. Bitte bewerten Sie Ihre Benutzererfahrung mit dem System im Hinblick auf die folgenden Aussagen.

	Stimme überhaupt nicht zu (1)	Stimme nicht zu (2)	weder noch (3)	Stimme zu (4)	Stimme voll zu (5)
Ich denke, dass ich das System gerne häufig benutzen würde.	○	○	○	○	○
Ich fand das System unnötig komplex.	○	○	○	○	○
Ich fand das System einfach zu benutzen.	○	○	○	○	○
Ich glaube, ich würde die Hilfe einer technisch versierten Person benötigen, um das System benutzen zu können.	○	○	○	○	○
Ich fand, die verschiedenen Funktionen in diesem System waren gut integriert.	○	○	○	○	○
Ich denke, das System enthielt zu viele Inkonsistenzen.	○	○	○	○	○
Ich kann mir vorstellen, dass die meisten Menschen den Umgang mit diesem System sehr schnell lernen.	○	○	○	○	○
Ich fand das System sehr umständlich zu nutzen.	○	○	○	○	○
Ich fühlte mich bei der Benutzung des Systems sehr sicher.	○	○	○	○	○
Ich musste eine Menge lernen, bevor ich anfangen konnte das System zu verwenden.	○	○	○	○	○

2. Bitte geben Sie nun Ihre Einschätzung ab. Kreuzen Sie bitte einen Kreis pro Zeile an.

Um die Anwendung zu bewerten, füllen Sie bitte den nachfolgenden Fragebogen aus. Er besteht aus Gegensatzpaaren von Eigenschaften, die die Anwendung haben kann. Abstufungen zwischen den Gegensätzen sind durch Kreise dargestellt. Durch Ankreuzen eines dieser Kreise können Sie Ihre Zustimmung zu einem Begriff äußern.

Entscheiden Sie möglichst spontan. Es ist wichtig, dass Sie nicht lange über die Begriffe nachdenken, damit Ihre unmittelbare Einschätzung zum Tragen kommt.

Bitte kreuzen Sie immer eine Antwort an, auch wenn Sie bei der Einschätzung zu einem Begriffspaar unsicher sind oder finden, dass es nicht so gut zur Anwendung passt.

Es gibt keine „richtige" oder „falsche" Antwort. Ihre persönliche Meinung zählt!

Es gibt keine „richtige“ oder „falsche“ Antwort. Ihre persönliche Meinung zählt!

	1	2	3	4	5	6	7	
unerfreulich	○	○	○	○	○	○	○	erfreulich
unverständlich	○	○	○	○	○	○	○	verständlich
kreativ	○	○	○	○	○	○	○	phantasielos
leicht zu lernen	○	○	○	○	○	○	○	schwer zu lernen
wertvoll	○	○	○	○	○	○	○	minderwertig
langweilig	○	○	○	○	○	○	○	spannend
uninteressant	○	○	○	○	○	○	○	interessant
unberechenbar	○	○	○	○	○	○	○	voraussagbar
schnell	○	○	○	○	○	○	○	langsam
originell	○	○	○	○	○	○	○	konventionell
behindernd	○	○	○	○	○	○	○	unterstützend
gut	○	○	○	○	○	○	○	schlecht
kompliziert	○	○	○	○	○	○	○	einfach
abstoßend	○	○	○	○	○	○	○	anziehend
herkömmlich	○	○	○	○	○	○	○	neuartig
unangenehm	○	○	○	○	○	○	○	angenehm
sicher	○	○	○	○	○	○	○	unsicher
aktivierend	○	○	○	○	○	○	○	einschläfernd
erwartungskonform	○	○	○	○	○	○	○	nicht erwartungskonform
ineffizient	○	○	○	○	○	○	○	effizient
übersichtlich	○	○	○	○	○	○	○	verwirrend
unpragmatisch	○	○	○	○	○	○	○	pragmatisch
aufgeräumt	○	○	○	○	○	○	○	überladen
attraktiv	○	○	○	○	○	○	○	unattraktiv
sympathisch	○	○	○	○	○	○	○	unsympathisch
konservativ	○	○	○	○	○	○	○	innovativ

3. Welche Probleme konnten Sie während der Benutzung des Systems feststellen ?

4. Haben Sie sonst noch Anmerkungen?

11.2 UI guidelines

Situation awareness
1. System state awareness: Inform the user on the cobot's state. The interface should support the user in maintaining appropriate awareness of the system's state. *Example: The cobot informs the user via lights on the robotic arm about the current state, for instance a green light means a program is running.*
2. Situation awareness: Inform the user regarding the cobot's environment and configuration. Help the user in understanding the configuration of the cobot in its environment, as well as other sensor inputs. *Example: The GUI shows a 3D model of the robot, which indicates the cobot's configuration and end effector position.*
3. Accessibility of information: Allow users to access information required for the task. Make sure the information the user needs for the task is available and accessible, considering possible restrictions. *Example: When editing a trajectories, a user can access information about the exact end effector location on the GUI.*
System understanding
4. Feedback: The UI is responsive to user actions. Respond to user actions so the user can follow task progress and understand the effects of their actions. *Example: The UI is responsive when buttons are clicked, when the user navigates to a different menu, or when values are updated.*
5. Affordances: Signify how the user can interact with the cobot. The interface should indicate which actions are currently possible and which ones are not. *Example: An icon of a trash bin next to a stored point indicates the point can be deleted by clicking on the icon.*
6. Errors: Give clear explanations and steps to recover when errors occur. Tolerate minor user errors, prevent critical system errors, support undo and redo. *Example: When a trajectory cannot be executed, a popup appears with an explanation why the error occurred and steps to recover from the error.*
7. Mental model: Support the user in understanding the way the system works. Support the user in understanding the connection between user actions and system response, for instance by providing feedback and using appropriate terminology. *Example: The user can play a programmed trajectory as an animation on the GUI before execution by the cobot.*
8. Help and documentation: Provide contextual help and documentation. Give users clear explanations of functionality and errors. *Example: When the help icon is clicked, the help menu displays help items that relate to the functions that are currently on the display.*
9. Support user learning: Help the user solve their (automation) problem. Support trial-and-error behavior and provide templates, contextual instructions or other clues that indicate how the cobot can be interacted with. *Example: Templates for robot tasks are provided, so the user has an idea what a program should look like.*
Task efficiency
10. Efficiency: Avoid unnecessary work on the user's side. Minimize the number of steps required to achieve goals and provide shortcuts. *Example: The user does not have to set the speed and acceleration for each point in a trajectory, but can specify these values for the whole trajectory.*
11. Task progress: Communicate to the user which task is being executed. The GUI should make it easy for the user to follow task execution by indicating previous, current and next steps. *Example: When the robot is executing a series of actions, the current action that is being executed is highlighted on the GUI.*
12. Reuse: Enable reuse of previous work. Support users in reusing their work or the work of others. *Example: Previous programs can be copied and edited.*

Figure 79: Design guidelines for collaborative industrial robot user interfaces 1-12 [79]

Human factors
13. Human factors: Design cobot and UI with ergonomics and accessibility in mind. Ensure the cobot UI is comfortable to work with for the necessary duration. *Example: The teach pendant is light to carry or can be placed on a table, so it will be comfortable to work with it for a few hours.*
14. Avoid cognitive overload: Reduce mental strain. Support recognition instead of requiring users to recall information, and limit the number of options that are presented. *Example: A function name such as "Wait" is easy to remember and indicates its function.*
15. User attention: Support the user in directing their attention. Make menu items that need attention visually salient. Do not attract attention unnecessarily. *Example: When the user is in the submenu for editing a trajectory, the menu items for editing points are the largest items.*
Configurability
16. Level of automation: Let the user determine the level of human input. The user can decide to integrate human input or to make the program fully automatic. *Example: There is a function for integrating human input, which requires the operator to press a button before the robot continues its task.*
17. Adaptable system architecture: Enable easy software integration after hardware exchange. The system architecture should allow for adapting the system to different types of tasks and application scenarios. *Example: It is easy to exchange the gripper and add sensors to the system.*
18. Adaptable tasks: Support easy editing of robot programs. Robot programs, trajectories, configurations should be editable by the user. *Example: Points in a previously stored trajectory can be deleted or changed.*
Interaction design of the UI
19. Consistent behavior: Make sure cobot and UI behave in a consistent way. Cobot behaviors, movement, and responses are predictable. *Example: The cobot always executes the same motion trajectory the same way.*
20. Multimodal UI: Consider the relation between different interaction modalities. Manage user attention across modalities and ensure the way information is presented via different modalities is consistent. *Example: The system provides feedback with LED lights on the cobot, which matches specific events on the UI.*
21. Graphic design: Design GUI items with usability, accessibility, and aesthetics in mind. Make sure information is presented in a clear and structured way, and use color, contrast and salience appropriately. *Example: Fonts are legible and the interface has appropriate contrast.*
22. Clarity of interface: Ensure the UI is easy and intuitive to use. Avoid a complex UI design; make use of simple graphics and icons. *Example: When selecting an action for the cobot, a sub menu for editing this action opens automatically.*
23. High vs. low complexity: Display programming functions at different levels of detail. Allow users to switch between simple and more complex ways of programming the cobot. *Example: There is the possibility to change between a simple version of the UI and a more complex version that provides more options.*
24. Customizability: Support user preferences. Enable users to change the interface according to their wishes and needs. *Example: It is possible to adapt different features based on user preference, such as the size of windows on the UI.*

Figure 80: Design guidelines for collaborative industrial robot user interfaces 13-24 [79]