

How to Teach Your Robot in 5 Minutes: Applying UX Paradigms to Human-Robot-Interaction

Martin Kraft and Markus Rickert

Abstract—When creating modern and visually appealing user experiences for the interaction with industrial robots, previously known and universally applicable paradigms in app and web design can be utilized to increase accessibility and usability of the to be created service. This is especially the case when the expected user group consists of untrained and inexperienced users and therefore system interaction focus is laid more on build progress overview, safety for human and robot, as well as overall simplification of complicated features. In this paper, we present four of the most important paradigms of modern graphical user experiences in web and app design that can be used to forward the concept of interacting with an industrial robot without any experience-related thresholds. By redesigning an existing interaction concept of a working robot cell system for assembly tasks in a small and medium-sized enterprise environment the presented paradigms are being utilized. The achieved improvements are then examined in a before-after user study to analyze the paradigm’s success in suiting the user’s expectation and anticipation using the redesigned service.

I. INTRODUCTION

In the modern era of customizable robot usage in production lines, mainly starting with the introduction of early predecessors from teach pendant-like devices in the mid to late 1980s, programming an industrial robot is done by entering its moves one step after the other. A trained expert utilizes the available input device connected to the robot, guides it to certain positions in its available workcell, and selects an iterative sequence of commands to form a sequence plan after his or her best effort and knowledge. However, this sort of input strategy is highly biased and objective to each expert. It is affected by an increasing number of aspects connected to the human operator [1], e.g. his skill level in using the input device, his knowledge in operating the system’s capabilities and level of education in analyzing and finding the ideal sequence strategy for the task. This sequential style of programming a robot is far off from being an ideal input strategy [2].

An ideal input strategy demands several characteristics missing in the *teaching pendant* approach. As a start, the required usage of proprietary hard-connected input devices, like the teach pendant, is slow, requires significant mastery and exceptional training before being usable in a professional environment, and demonstrates a threshold between human teaching and translation to robot movement in general [3]. It completely lacks the option to document and explain single moves in the created sequence, therefore making each input a unique order of moves incomparable and incomprehensible

Martin Kraft and Markus Rickert are with fortiss, An-Institut Technische Universität München, Guerickestr. 25, 80805 Munich, Germany



Fig. 1: The redesigned user interface running on an industrial touchscreen next to the robot workcell. The main focus is on ease-of-use for workers in small and medium-sized enterprises with small lot sizes.

to an untrained person and unanalyzable for general usage. Programs based on this classic concept do not provide a semantic description of the process and therefore fail to connect the seemingly arbitrary motions and I/O commands of a robot to their intended meaning. Another programmer will often have difficulty in understanding the reasons behind parts of the program.

In small and medium-sized enterprises (SME), process knowledge is often in the mind of one single domain expert. By simple observation, this knowledge is hard to master, as the expert will base his individual steps on certain assumptions, common knowledge of his trade, and special tricks he learned over the years. The ability to make this information accessible to robot systems while increasing usability for the worker is a major challenge in the area of Internet of Things and *Industrie 4.0* [4]. This is especially important since SMEs represent more than 99% of the businesses in the European Union [5]. Currently, only a very small percentage of them use robots in their production, especially in the manufacturing domain [6].

When the general interaction approach to a system and in particular the needed skill set necessary for using it is identified as a major threshold, the consequential next step for an enhanced usability should be the simplification and improvement of system accessibility toward user groups previously left out of the system’s access range. If a wider group of potential users is able to operate a system profoundly, safely, and intuitively using modern, universally applicable,

and understood paradigms, human-robot interaction is able to perform a significant step forward in its general accessibility [7], [8] (Fig. 1).

II. RELATED WORK

Published software in HRI-related fields can be broadly separated into two different categories. First, complex and professional CAD-based software [9] is used by experts to create and control robot-supported production. Second, lightweight and modernized tools try to implement new approaches in software-interaction. However, these *lighter* software tools are oftentimes associated with a playful character, like Lego's Mindstorms software or animated presentation of robots in general [10], and are missing the profound skill set necessary for usage in professional environments.

CAD-based program GUIs (Graphical User Interface) oftentimes orient themselves on conservative menu structures to compensate for their massive amount of functionality and options in usability. Deep menus are not unusual and demand extensive training before a user is able to use them correctly [11]. When describing the usage of GUIs in HRI research and science, problems appear to be even more drastic. Most GUIs described in papers are proof-of-concepts or simple visualizations of robotic activity and are not meant to be used in an actual working environment outside of the limited test parameters and builds [12]. Yet the presentation of GUIs is described as progressive and a beneficial addition to the actual research and work, although most of the time the GUI itself is in fact not deeply connected to the work, but rather created as an optional extra.

III. TASK-BASED GUI AND PROBLEM ANALYZATION

The foundation for this work is the first iteration of an intuitive workcell for small lot production as it is typically used in small and medium-sized enterprises [13], [14]. A video demonstration of this system with a comparison to a classical programming approach using a teach pendant can be found online¹. Compared to the direct approach of (low-level) coding or the functional wrapping of identical low-level strategies into new concepts like the teach pendant does, this new interaction strategy focuses on objects and corresponding tasks with a more abstract and higher-level approach that is common and understandable for a domain expert. The complex calculation and translation of user input to robot moves is still inevitable, yet the user is not burdened with its imminent importance and is able to focus on his task. Underspecified processes are handled through logic and inference in combination with an ontology of common and domain-specific knowledge. Therefore, users are able to achieve progress easier than before and with less complicated tools and overhead. The task-based GUI and described workflow can be seen in Fig. 2. In terms of performance, the new approach works significantly better in user-crucial areas like input speed and offers an overall better programming paradigm. In a direct task comparison

between a teach pendant-like device and the new GUI-based approach, the time needed by a domain professional to enter all necessary task-related inputs was reduced from 48 to 13 minutes [8]. So, while providing a step forward in the technical foundation and input-wise for domain experts alike, the project did not offer any form of improvement in the ground laying user experience concept since it does not provide feedback on the correctness of the process specification.

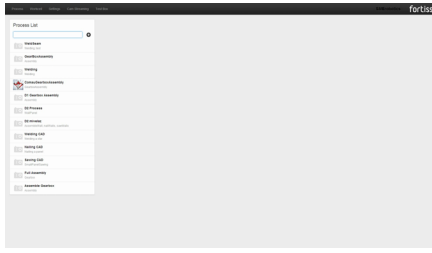
We began the redesign process by analyzing the task-based GUI's workflow in a user study with a focus on system usability, the finding of interaction thresholds, and perceived satisfaction of used input types. We discuss some of the study's findings in this section, while more information about the study and the comparison between the task-based GUI and its redesign approaches can be found in Section V.

Participants were able to use the presented service and finish the building task without the need of profound and prior knowledge, yet ranked the system's workflow at a low 2.82 out of 5 points as well as the satisfaction of interacting with the visual presentation at 2.72 out of 5 points. These ratings indicated a suffering usability from a discontinuous interaction concept, offering too complicated and complex functionality (Fig. 2a and Fig. 2c) to the inexperienced user as well as demanding mandatory inputs with technically challenging definitions to initialize the task instead of immediately starting with item selection and assembly definition (Fig. 2b). Furthermore, participants stated they received the feeling of achieving progress only at some point in the middle of the assignment, indicating an existing and increasing gap between the start of the assignment and the user's feeling of creating progress. Participants experienced unnecessary breaks to double-check performed decisions and oftentimes stopped to ascertain what has been done and what could be the next step in solving the assignment due to the missing confirmation and visual representation of correctly solved subtasks (Fig. 2e). In combination with the challenging functionality, these interaction flaws could potentially create severe thresholds when being used by inexperienced users. 67% of participants stated that they required additional help on several occurrences during the assignment, demonstrating the functional, yet too complex service design of the GUI (Fig. 2i), with 55% of participants experiencing more than 15 misclicks or accidental and wrongly chosen decisions (Fig. 2f) operating the system. A more detailed analysis of the task-based GUI's issues can be found in the thesis by Kraft [15].

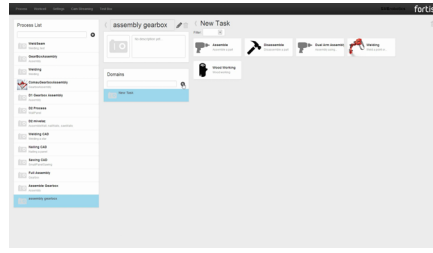
IV. UNIVERSALLY APPLICABLE DESIGN PRINCIPLES AND PROJECT REDESIGN

To increase the usability of the system, efforts were made to implement content visualization strategies and interaction paradigms which are easy-to-understand and universally applicable by all kinds of users. These principles can be found in today's design of internet services, websites, and—even more clearly—app design for touch-input devices like smartphones and tablets. Design principles applied in this field

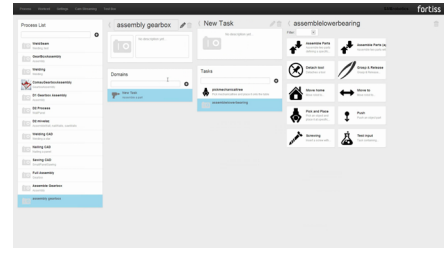
¹<https://www.youtube.com/watch?v=B1Qu8Mt3WtQ>



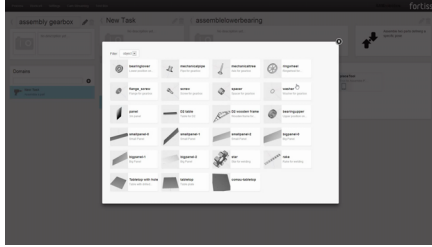
(a) Process creation is achieved via a small plus sign in the top right corner of the list.



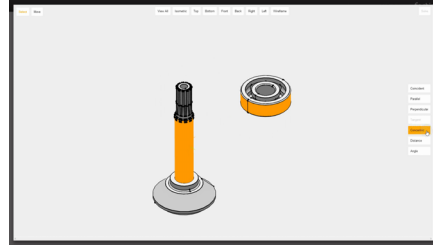
(b) A new process must have a *domain* as well as 1 out of 5 predefined purposes.



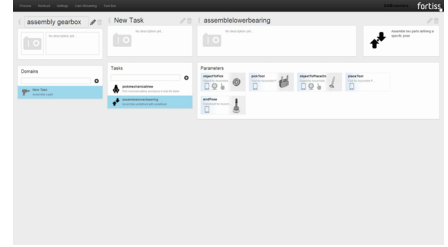
(c) Task creation is again done via a small plus sign. A task must have 1 of 10 predefined purposes.



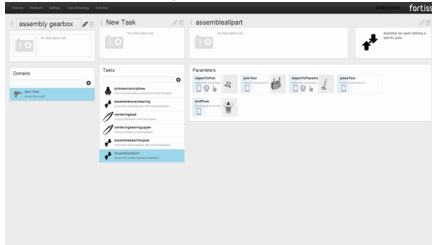
(d) The parts to be combined must be selected from a large list.



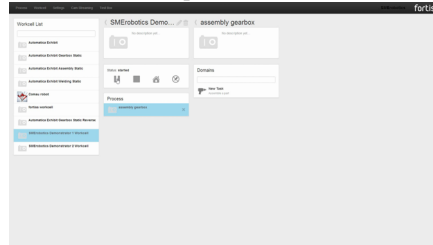
(e) Constraints between two items are defined via touch input and a button.



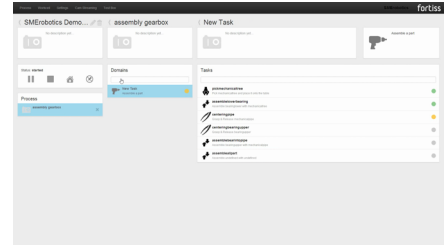
(f) A created task (with its constraints) is added to the task list view.



(g) The completed assignment with all necessary tasks.



(h) Running the process is controlled via a small start-stop panel.



(i) Process execution is visualized via colorful dots in a list view.

Fig. 2: The status quo of the GUI with a pane-based grid of list views to represent (a) a process, (b) a domain list, (c) a task list, and (d) selector options. Constraints between items are (e) surface-based and created tasks are displayed in (f)–(g) the task list. Executing the task is done via (h)–(i) list-based options.

of usage are exposed to immense amounts of active users and have proven themselves accessible and user-supportive in presenting and solving tasks. By applying these paradigms to the context of our innovative technical solution, we gain the potential of raising the technical system into a more complete and immersive user experience.

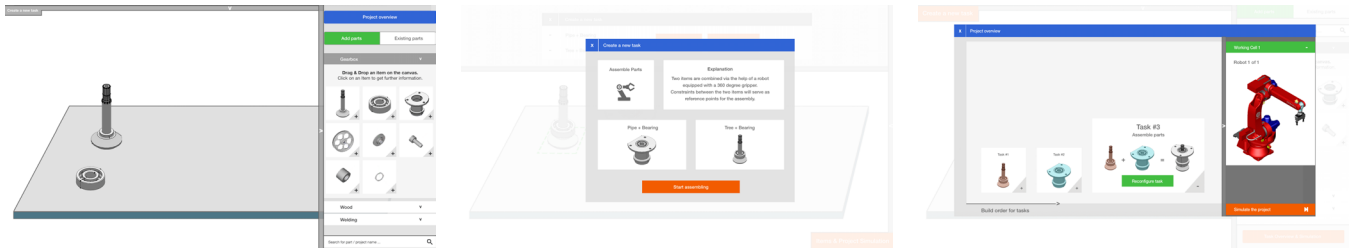
A. Reintroduction of WYSIWYG

The newly created and experimental introduction of standards in the rise of smartphones and tablets in the mid 2000s introduced a shift in the static approach to interaction concepts of services. While previous concepts implemented complex and WIMP-like (Windows Icon Menu Pointer) features, new services needed to fit on smaller devices and lead to a step-wise content presentation and on-screen controls.

This newly created universal approach has proven itself viable and usable by millions of devices and users in all age groups and resembles the previously known WYSIWYG (What You See Is What You Get) approach. This paradigm-approach has changed the way people consume information as well as software interfaces and raise user

expectations on how a service should be build, which elements should be used, and how different sequences should be triggered [16]. The application of the WYSIWYG approach is best seen in our redesigned GUI by looking at the item selection (Fig. 4b) and item usage on the canvas (Fig. 4c). Choosing a preferred item via a small object model in a presorted list view, dragging and placing it onto the canvas where the user prefers it to be placed, and then directly clicking on it to start an interaction is the essential representation of this direct interaction paradigm. Compared to the status quo (Fig. 2a, Fig. 2b, and Fig. 2c), the user involvement is much more rewarding. Actions are clearer using drag and drop capabilities and instant visual feedback (Fig. 4e).

General trends in project development of the last years include the software-wise focus on content, general minimalism and context reduction with focus on clarity in motion, and reaction of software behavior. The *action and reaction* approach is more viable than before [17]. It is observable, that some users find the similarity in software and interface design boring, however they still get irritated by designs varying from the standard by introducing alternative interac-



(a) Introduction of a virtual working canvas with item libraries (b) Test of additional confirmation screens throughout the workflow. (c) New interface for mapping a process to a workcell.

Fig. 3: The clickdummy workflow was designed to center around a central (a) virtual working canvas. (b) Additional confirmation screens were tested, and a new (c) style of mapping a process to a workcell was evaluated.

tion strategies [18], therefore leading to a very conservative conclusion: Human-robot interaction must not reinvent the way people use systems and it must not reinvent itself in revolutionary interaction disciplines. It needs to adapt proven, valid, and already existing universally applicable paradigms to align itself to the way users expect it to work in these (modern) times. In general, this approach is called *user-centered design*. User experience design and its paradigms present an existing field of research with a lot of different approaches and expectations that can be applied to human-robot interaction. HRI can orient itself along these lines to deliver a smooth, interesting, and visually appealing experience to help people engage in a topic they would previously have deemed too difficult to use [19].

B. User Interface Complexity & Elements

In general, problem-solving UX paradigms for the field of HRI can be summarized in three important and interaction-improving points, especially when creating software to be used by non-expert user groups with a focus on widely available hardware.

1) *Simplification*: The instruction set of a robot builds upon the complex foundation of robot actions with Euler angle-based pose definitions, different motion types, kinematic singularities, reachability based on joint limits and configuration, or handling of uncertainties. This difficult and rich functionality presents a constant threat of overexerting the user, who needs to ultimately handle and master the available options correctly regardless of his skill level. However, this complexity must not be visualized onscreen to the (novice) user [20]. If a user wants to connect two parts, he only needs to be presented with the rendering of two parts. The interaction must not be made overcomplicated [21] and the complex computation can be executed in the background. Therefore, simplification reduces complexity in favor of focused usability and narrowed functionality. As a strong point in case, the entry to create a new process in our redesigned GUI is displayed in a focused and simplified manner (Fig. 4a) and lacks the previously mandatory process description choices (Fig. 2a and Fig. 2b) in favor of a more direct approach to begin the assignment. Because the process description is dependent on choice of items and subsequent item combinations, this step is solvable by the system and

does not need to be declared by the user.

2) *User-Supporting Software*: When operating expensive machines with little or no knowledge of pre-existing patterns, the existing sense of security felt by the user in terms of actual physical safety as well as safety in software-side inputs must be present at all times and has to be well-defined [22]. Entering potential catastrophic data must be intercepted by the system with visual support like warnings and simulations. Again, these calculation-heavy security mechanisms can run in the background, unseen by the user, yet are indispensable for recognizing wrong calculations and errors [17]. If such critical implementations are possible to occur, the interaction threshold rises immensely and may be too high to be acceptable for untrained user groups [23].

While redesigning the GUI, such paradigms are presented in the updated constraints GUI and subsequent renderings of connection choices (Fig. 4d). By giving the user a preview of what to expect when executing his item combination, the user is able to take an important decision in a complex and wording-wise difficult scenario. The same goes for the finished process and newly implemented simulation (Fig. 4i), where the user is able to visually check the created process for possible mistakes and unwanted behavior before starting the process on the actual system.

3) *Multimodality and Abstraction*: When analyzing input strategies and user behavior for modern projects, the approach for *user-centered design* is inevitable. Software should adapt on how users approach the to-be-fulfilled tasks, and human-robot interaction, due to its complexity and novelty, is struggling to find an adequate categorization of functionality-to-complexity ratio [20].

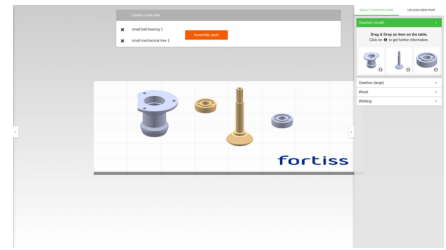
The paradigm of *different ways lead to the finished result* can only be applied partially to all HRI software in general due to its many and complex dependencies and usages. However, it is important to analyze the existing sub-parts of programs or services responsible for handling user-inputs and enabling progress by interacting with the user. The analysis of such key system interaction points, which can be affected and improved by user-centered design, must be found and subsequently updated. In our case, the selection of parts in a build project and their connection points share identical dependencies in every finished process, yet the chronology of build steps themselves can be massively different from



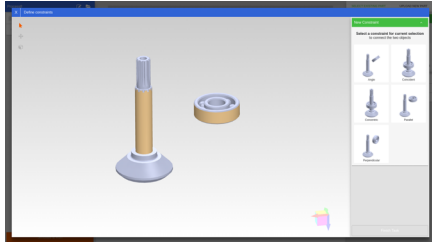
(a) A new process is created via a focused menu and large button.



(b) Item selection is achieved by dragging items from the right menu onto the canvas.



(c) By selecting the items to be combined, the top menu offers logical and available task creation options to the user.



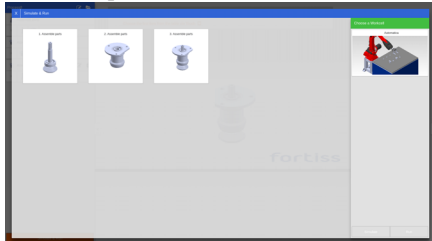
(d) Constraints between two items are defined via touch input and a rendered simulation of options to choose from.



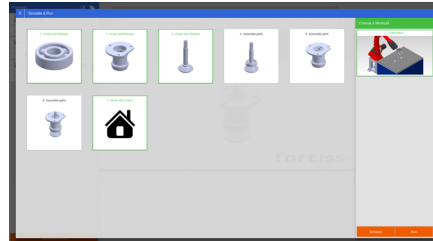
(e) A created task (with its constraints) is added to the task list view as well as the visual combination of items on the table.



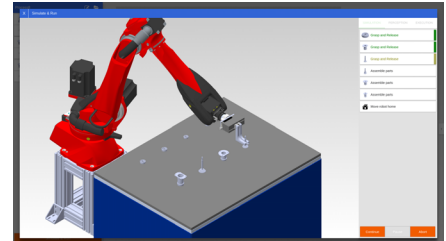
(f) The completed assignment with all necessary tasks.



(g) The abstract process description can now be mapped to an existing robot workcell.



(h) When mapped, all additional workcell-specific dependencies are added to the process.



(i) The user-generated process can now be simulated, checked for errors, executed, and supervised.

Fig. 4: The canvas-based GUI where (a) a new process is created by presenting the user with a focused menu of two options. Creating a new process mimics (b) a digital and empty working canvas acting as an overview, offering the possibility to drag and drop the required items on the canvas. A new task is created by (c) selecting items, with the task menu automatically showing the executable task options for the selected number and type of items. Constraints between items are (d) surface-based and offer user-supportive renderings of connection options. The newly created tasks are being displayed (e) in a task list as well as (f) being visualized as building progress on the canvas. The abstract process is now (g)–(h) mappable to different robot workcells and offers the possibility to (i) simulate robot execution.

user to user.

The abstraction of *real* or 1:1 visualization can be considered one of the strongest and potentially most important improvements in usability of future HRI GUIs. While professional CAD software must limit itself to a realistic visualization, this is not the case for interfaces optimized for untrained user groups. Since the interaction is more focused on the previous task, safety, or progress-related issues than realism itself [24], selected items can be made bigger than others. The relation of items to each other does not have to be correct if this offers a visual benefit to the user, e.g., when trying to highlight or select small points in scale.

C. Content Visualization

The visual representation of achieved progress can be displayed to the user in different approaches. WIMP-wise

small icons and texts in a list of tiles [25] are a possibility. However, using the mentioned post-WIMP paradigms in this section, we emphasized a F Pattern Layout—a form of content structuring in navigation design—by keeping an upper navigation bar as well as additional navigation elements on the left and right side of the screen. This progress overview visualizes the user-achieved progress by showing the selected and used parts as well as already performed combinations (Fig. 4e), resulting in a more centered focus on the actual task and workflow rather than calculations and complexity (Fig. 4f). This is aided by the navigational structure being more interactive and logically sequenced in its appearance and being displayed only when its functionality is needed in the expected workflow [26].

Again, one of the key aspects of user-centered design focuses around the idea of creating user-interface elements

styled with purpose and meaning [17]. Color and shape of interaction elements are able to improve usability in terms of navigation flow as well as program structure. These elements are re-usable throughout the service interaction, thus aiding users in recognizing important interaction points [23].

In our project, the visualization mimics the canvas of the actual workcell (Fig. 4d), therefore aiding overall system immersion while offering the abstraction of extended and enlarged items presented to the user. The support for technological additions like touchscreen usage offers a more convenient way to expect a successful user selection on a bigger potential target area. By implementing a drag and drop style input strategy, we are able to place the GUI closer to the mentioned app-like paradigms of *action and reaction* [27]. This newly introduced overview canvas also serves as the main interaction point with additional layers popping up as separate windows. The presentation of selected items and achieved progress in form of 3D models complements the presented paradigms. We are able to present a visually appealing, easy-to-use, and abstracted version of the complex and difficult realism which runs in the background. All this culminates in the user-selectable functionality to choose two big objects, select a combination method, and see the finished result update itself (two objects merge into each other) on the process overview canvas, therefore triggering the *action to reaction* paradigm representing the user-achieved progress. When differentiating between professional and inexperienced users, the expectation of behavior and program usage as well as the strategic approach to performing and finishing a task can differ immensely. One of the most important aspects of visualization is called *Where does a user expect what interaction?* [28]. This type of simplified visualization can also be found in our redesigned constraints view. By laying the focus on affected and selected icons with reduced option functionality, the general user understanding of existing options and part-related requirements is improved, therefore severely aiding the feeling of progress achievement.

D. Improving the Program's Sequential Logic

When using a complex system, one of the most important aspects of paradigm shifts includes the splitting or simplification of functionality. Increasing the importance of this aspect even more, usable interface element groups of the service's functionality should respond to the proceeded input and react accordingly and expectedly. Displaying only valid parts of the available functionality when needed supports users in focusing attention on the required inputs thus increasing the workflow even more [29]. As an example: At first the user is going to select the actual objects to work with, therefore displaying the corresponding item selection menu structures, then afterwards displaying the menu structures to choose and connect constraint options and last, offering interface options for finishing or simulating the achieved progress. Updating and displaying required interface parts as screen overlays above the existing content increases the sequential logic even further, again helping to set the focus on functionality and optimal user flow.

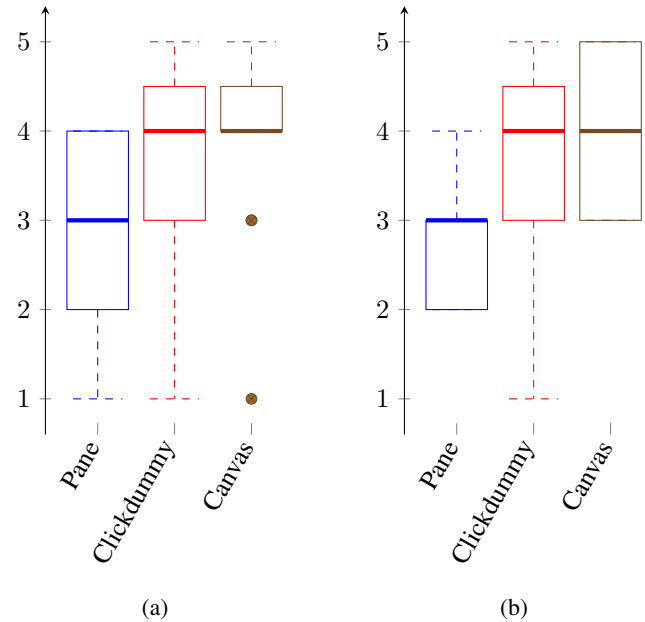


Fig. 5: User satisfaction regarding different aspects with a rating from 1 (least) to 5 (best) points. (a) Workflow through the system, (b) interaction with the process view.

For our GUI Redesign, we used this paradigm to display the still generic version of the user-created process (Fig. 4g) and its working-cell mapped counter-part (Fig. 4h), arranging the mandatory inputs into step-by-step wise decisions.

V. USER STUDIES AND EVALUATION

As presented in Section IV, a redesign was initiated to implement the mentioned universally applicable user experience paradigms in pursuit of increased user accessibility and usability of the system. The task to be performed by the participants in all three user studies is the identical robot-aided build of a four-part gearbox, which can be viewed online². All three studies consisted of a first part in which the participants interacted with the real workcell to solve the task, while being recorded on video, and a second part in which a questionnaire with 26 questions was filled out to gather feedback in the fields of user satisfaction, user awareness of progress-making, as well as user-interface and user-experience related topics.

The first user study was conducted to examine the initial GUI for user-experience related thresholds with 12 participants. Some of the results have been discussed in Section III, with 50% of participants being complete novices to robot workcells. 42 percent of the participants were younger than 26 years, compared to 58% that were 26 or older.

Redesigning the task-based GUI reached an intermediate result with the creation of a GUI clickdummy prototype. Some example screens can be seen in Fig. 3. This clickdummy was used as an opportunity to create a comparison to the status quo, verifying the application of mentioned paradigms actually represented a step in the right direction

²<https://www.youtube.com/watch?v=ZxMZrs1nf7Q>

and furthermore provided valuable user feedback heading into the redesign implementing phase. This second user study was conducted with 20 participants and 50% of participants being completely new to robot workcell environments. The study involved 10% participants younger than 26 years.

Finally, the third user study tested the implemented and working GUI redesign on the identical hardware as the previous two studies. The study was again conducted with 20 participants and 60% novices to the robot workcell, with 20% of the participants younger than 26 years.

An example for a newly tested feature reviewed in the clickdummy was the introduction of several confirmation screens throughout the service (Fig. 3b). These screens should provide short overviews over previously entered user inputs and were meant to raise user awareness, yet were met with mixed feedback by the clickdummy study participants due to the extension of total system usage time as well as the addition of further inputs the user had to perform.

A. Increased User Rating for Overall Workflow

One of the key issues of the pane-based system was its complexity content-wise, wording-wise, and interaction-wise. The participants in the initial study struggled with all three points and awarded the pane-based workflow only 2.82 out of 5 maximum points. The participants testing the clickdummy certified the redesigned concept improvements regarding content and interaction and awarded 3.75 out of 5 points to the system. The participants testing the build and running redesigned service generally agreed with the clickdummy testers and awarded an average of 4.0 out of 5 points (Fig. 5a). The user studies indicate improvements ($F(2, 48) = 4.79226, p = .012661$) in terms of user-system interaction, although existing interaction problems connected to the system wording are still present.

B. Iconization Paradigms Work

One of the key arguments for the implementation of known and previously used paradigms is its supportive nature for the user. A user is able to associate positive emotions as well as interest when engaging a service while feeling supported. By redesigning the existing text-based information to 3D icons, the user acceptance of the system has increased from 2.72 out of 5 for the old system to 3.75 out of 5 in the clickdummy concept and 3.95 out of 5 points for the redesigned system (Fig. 5b). The studies indicate the successful support of user interaction via app-like paradigms and TUI-like (Tangible User Interface) devices, such as the used touchscreen ($F(2, 48) = 6.95958, p = .002218$).

C. Faster Feeling of Progress Achievement

The user experience of a service includes the user's anticipation, actual task fulfillment, and subsequent impression. Therefore, a beneficial user experience should support users throughout the task specification and in ultimately completing the task successfully, detached from any potential complexity and difficulty of task-related inputs. For this user study task, a key factor to be measured indicating the

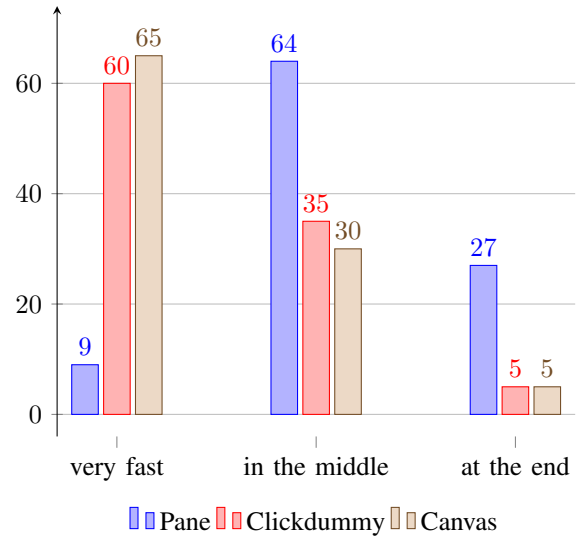


Fig. 6: User feeling of progress achievement using the system. Rating from 0% to 100%.

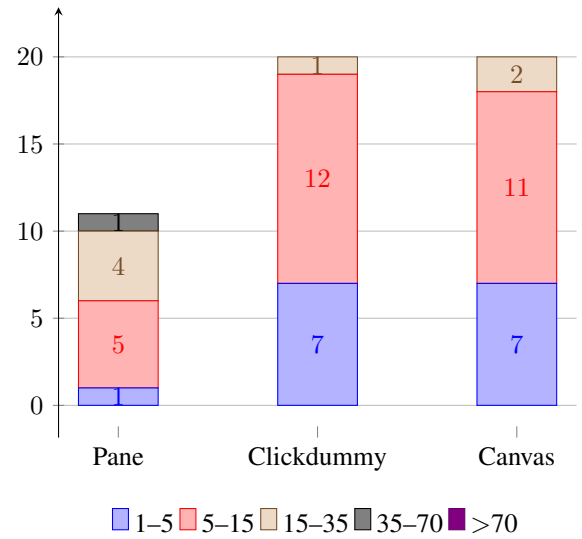


Fig. 7: Number of misclicks scored by participants using the different systems.

successful motivation for or resistance against system usage is the feeling of achieving progress solving the task. The overall user motivation directly corresponds to the user's feeling of achieving progress. 64% of participants using the old system said that they received the feeling of achieving progress at some point in the middle of the task. However, 60% of participants of the clickdummy concept and 65% of the redesigned system's users felt an achievement of progress very fast from the start (Fig. 6).

D. Overall Reduction in Accidental Misclicks

A major indicator for a system's usability is how confident and purposeful the user is able to navigate through the offered functionality. A high number of (unintended) misclicks leads to severe problems in the short term (stopping and

double-checking inputs) as well as the long term (reduced system trust and motivation).

While performing the task with the pane-based GUI, most users scored between 5 to 15 (45%) and 16 to 35 misclicks (36%). The clickdummy improved the overall misclick rate, with participants mostly scoring 1 to 5 (35%) and 5 to 15 misclicks (60%). The canvas-based design produced similar results with participants scoring 1 to 5 (35%) and 5 to 15 misclicks (55%) (Fig. 7).

VI. CONCLUSION

We presented some of the most important aspects of modern web and app design, such as the implementation of easy-to-understand and universally applicable paradigms, the focus on complete user experiences and user-centered design, as well as the importance of simplification and general abstraction when designing graphical user interfaces for human-robot interaction. By applying and implementing the introduced paradigms into an existing system, we were able to investigate the positive effects of such paradigms in a robotic workcell application. Progress-wise, the user study results indicate the necessity of paradigm usage and enhance the interaction with the system in terms of fewer accidental and intentional misclicks, faster and more fluid navigational decisions, as well as a quicker feeling of progress-achieving by the participants. All these factors indicate a better user experience throughout the service, also leading to higher overall system scores given by the participants. Indicating the usefulness of such implemented pre-known paradigms, more attention and efforts should be laid into designing a beneficial user experience for inexperienced users in a robotic workcell environment.

REFERENCES

- [1] V. Nitsch and T. Glassen, "Investigating the effects of robot behavior and attitude towards technology on social human-robot interactions," in *Proc. of the IEEE Intl. Symposium on Robot and Human Interactive Communication*, 2015, pp. 535–540.
- [2] C. P. Quintero, R. Tatsambon, M. Gridseth, and M. Jagersand, "Visual pointing gestures for bi-directional human robot interaction in a pick-and-place task," in *Proc. of the IEEE Intl. Symposium on Robot and Human Interactive Communication*, 2015, pp. 349–354.
- [3] R. D. Schraft and C. Meyer, "The need for an intuitive teaching method for small and medium enterprises," in *Proc. of the Joint Conf. of the Intl. Symposium on Robotics and the German Conf. on Robotics*, 2006.
- [4] R. Drath and A. Horch, "Industrie 4.0: Hit or hype?" *IEEE Industrial Electronics Magazine*, vol. 8, no. 2, pp. 56–58, 2014.
- [5] A. Airaksinen, H. Luomaranta, P. Alajääskö, and A. Roodhuijzen, "Statistics on small and medium-sized enterprises," Eurostat Statistics Explained, 2015.
- [6] P. Gorle and A. Clive, "Positive impact of industrial robots on employment," Metra Martech Limited, Tech. Rep., 2013.
- [7] S. Profanter, A. Perzylo, N. Somani, M. Rickert, and A. Knoll, "Analysis and semantic modeling of modality preferences in industrial human-robot interaction," in *Proc. of the IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems*, 2015, pp. 1812–1818.
- [8] A. Perzylo, N. Somani, S. Profanter, I. Kessler, M. Rickert, and A. Knoll, "Intuitive instruction of industrial robots: Semantic process descriptions for small lot production," in *Proc. of the IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems*, 2016, pp. 2293–2300.
- [9] K. Baizid, S. Čuković, J. Iqbal, A. Yousnadj, R. Chellali, A. Meddahi, G. Devedžić, and I. Ghionea, "IRoSim: Industrial robotics simulation design planning and optimization platform based on CAD and knowledgeware technologies," *Robotics and Computer-Integrated Manufacturing*, vol. 42, pp. 121–134, 2016.
- [10] C. Bartneck, M. Soucy, K. Fleuret, and E. B. Sandoval, "The robot engine - making the Unity 3D game engine work for HRI," in *Proc. of the IEEE Intl. Symposium on Robot and Human Interactive Communication*, 2015, pp. 431–437.
- [11] N. Vahrenkamp, M. Wächter, M. Kröhnert, K. Welke, and T. Asfour, "The robot software framework ArmarX," *Information Technology*, vol. 57, no. 1, pp. 99–111, 2015.
- [12] S. Cremer, F. Mirza, Y. Tuladhar, R. Alonzo, A. Hingeley, and D. O. Popa, "Investigation of human-robot interface performance in household environments," in *Proceedings of SPIE, Sensors for Next-Generation Robotics III*, 2016.
- [13] S. Profanter, "Implementation and evaluation of multimodal input/output channels for task-based industrial robot programming," Master's thesis, Technische Universität München, Munich, Germany, 2014.
- [14] A. Perzylo, N. Somani, S. Profanter, M. Rickert, and A. Knoll, "Toward efficient robot teach-in and semantic process descriptions for small lot sizes," in *Proc. of the Workshop on Combining AI Reasoning and Cognitive Science with Robotics, Robotics: Science and Systems*, 2015.
- [15] M. Kraft, "Design and evaluation of an intuitive graphical user interface for industrial robotic system programming," Bachelor's thesis, Technische Universität München, Munich, Germany, 2016.
- [16] M. Fauquex, S. Goyal, F. Evequoz, and Y. Bocchi, "Creating people-aware IoT applications by combining design thinking and user-centered design methods," in *Proc. of the IEEE World Forum on Internet of Things*, 2015.
- [17] B. Buxton, *Sketching User Experiences: Getting the design right and the right design*. Morgan Kaufmann, 2007.
- [18] T. Römer and R. Bruder, "User centered design of a cyber-physical support solution for assembly processes," *Procedia Manufacturing*, vol. 3, pp. 456–463, 2015.
- [19] J. Sauer and A. Sonderegger, "The influence of product aesthetics and user state in usability testing," *Behaviour & Information Technology*, vol. 30, no. 6, pp. 787–796, 2011.
- [20] V. Winter, J. Perry, H. Siy, S. Srinivasan, B. Farkas, and J. McCoy, "The tyranny of the vital few: The Pareto principle in language design," *Journal of Software Engineering and Applications*, vol. 4, no. 3, pp. 146–155, 2011.
- [21] K. Ferris and S. Zhang, "A framework for selecting and optimizing color scheme in web design," in *Proc. of the Hawaii Intl. Conf. on System Sciences*, 2016, pp. 532–541.
- [22] E. Raita and A. Oulasvirta, "Too good to be bad: The effect of favorable expectations on usability perceptions," *Proc. of the Human Factors and Ergonomics Society Annual Meeting*, vol. 54, no. 26, pp. 2206–2210, 2010.
- [23] A. L. Marin Mejia, "Robotic interfaces design: Avatar and GUI competing for older users attention," in *Proc. of the Intl. Conf. on Human Aspects of IT for the Aged Population. Design for Everyday Life*, 2015, pp. 300–310.
- [24] A. T. Al-Halhouli, H. Qitouqa, N. Malkosh, A. Shubbak, S. I. Al-Gharabli, and E. Hamad, "LEGO mindstorms NXT for elderly and visually impaired people in-need: A platform," *Technology and Health Care*, vol. 24, no. 4, pp. 579–585, 2016.
- [25] G. M. Poor, S. D. Jaffee, L. M. Leventhal, J. Ringenberg, D. S. Klopfer, G. Zimmerman, and B. A. Klein, "Applying the Norman 1986 user-centered model to post-WIMP UIs: Theoretical predictions and empirical outcomes," *ACM Transactions on Computer-Human Interaction*, vol. 23, no. 5, pp. 30:1–30:33, 2016.
- [26] Y. Zhang, L. Wang, and W. Luo, "Research on web design style based on digital technology," in *Proc. of the Intl. Conf. on Measuring Technology and Mechatronics Automation*, 2017, pp. 443–446.
- [27] D. Norman, "Design, business models, and human-technology teamwork," *Research-Technology Management*, vol. 60, no. 1, pp. 26–30, 2017.
- [28] D. Hama, M. Kurioka, M. Kato, K. Imamura, and M. Nakanishi, "Novel method of evaluating GUI design from the viewpoint of worker experience," in *Proc. of the Intl. Conf. of Design, User Experience, and Usability. Design Philosophy, Methods, and Tools*, 2013, pp. 283–293.
- [29] D. A. Norman, *The Design of Everyday Things*. Basic Books, 2013.