Towards Tool-Support for Robot-Assisted Product Creation in Fab Labs

Jan Van den Bergh[1](✉), Bram van Deurzen[1], Tom Veuskens[2], Raf Ramakers[1], and Kris Luyten[1]

[1] Hasselt University - tUL - Flanders Make, Expertise Centre for Digital Media,
Wetenschapspark 2, 3590 Diepenbeek, Belgium
{jan.vandenbergh,bram.vandeurzen,raf.ramakers,kris.luyten}@uhasselt.be
[2] Hasselt University, Hasselt, Belgium
tom.veuskens@student.uhasselt.be

Abstract. Collaborative robot-assisted production has great potential for high variety low volume production lines. These type of production lines are common in both personal fabrication settings as well as in several types of flexible production lines. Moreover, many assembly tasks are in fact hard to complete by a single user or a single robot, and benefit greatly from a fluent collaboration between both. However, programming such systems is cumbersome, given the wide variation of tasks and the complexity of instructing a robot how it should move and operate in collaboration with a human user.

In this paper we explore the case of collaborative assembly for personal fabrication. Based on a CAD model of the envisioned product, our software analyzes how this can be composed from a set of standardized pieces and suggests a series of collaborative assembly steps to complete the product. The proposed tool removes the need for the end-user to perform additional programming of the robot. We use a low-cost robot setup that is accessible and usable for typical personal fabrication activities in Fab Labs and Makerspaces. Participants in a first experimental study testified that our approach leads to a fluent collaborative assembly process. Based on this preliminary evaluation, we present next steps and potential implications.

Keywords: Human-robot collaboration · Toolkit
End-user development

1 Introduction

Robots were introduced in manufacturing during a time where replicating a single design for mass-market was a dominant strategy that enabled more people to enjoy the advantages of technology. In recent years, however, focus returns to customized or even personalized products. Likewise, robotic support is evolving to support these more flexible production requirements. Collaborative robots or

Published by Springer Nature Switzerland AG 2019
C. Bogdan et al. (Eds.): HCSE 2018, LNCS 11262, pp. 219–230, 2019.
https://doi.org/10.1007/978-3-030-05909-5_13

cobots are introduced as cheaper, more flexible robots than traditional industrial robots. Cobots can be reprogrammed by their operators to support customized, shorter-lived tasks. While these tools are effective, these still require separate, offline programming of the robot. This can be cost effective for customized products where there is still some repetition of the same task. It however becomes undesirable for personalized or one-off production. Such production is already happening in Fab Labs and Makerspaces.

In this paper, we explore how programming robotic support for personalized production in e.g. Fab Lab or Makerspace can be realized. To do so, we started from the tasks that creators need to perform when they want to create a new three-dimensional object. These tasks include the conceptualization of the product, trying things out digitally, specification of the different parts of the artifact, creation of all components using machines, such as laser cutters and 3D printers, and finally assembly of the artifact.

While technological support exists for most of these steps, the assembly step is still a manual process that may sometimes be cumbersome. In this paper, we start from the assumption that a robot could support this step and could assist the creator by offering a third hand. We present our envisioned integrated toolkit, as well as the current prototype that includes a design tool as well as robotic support that assists during the assembly process. We present the results of a first preliminary study with the toolkit's robotic support and offer directions for future work.

2 Related Work

Our approach partly relies on a CAD (computer-aided design) model of the product that a *maker* wants to create. There are already some approaches that use CAD data to ease programming of robots. We provide a short selection of related work on how CAD models are already used to program robots and on different approaches to end-user programming of robots.

2.1 Tools for CAD-Based Robot Programming

CAD-based programming is not a new idea; it was already investigated over 30 years ago [3]. It however is still an active research area that also benefits from advances in commercial tools, as is exemplified by some recent approaches.

Neto et al. [7,8] propose such a CAD-based system. It uses Autodesk Inventor as the CAD system and the communication is through the API. Starting from a CAD model of the robotic cell, the user generates a robot program by drawing the desired robot path. The end effector position and rotation has to be defined as well, by placing simplified tool models along the robot paths. Interpolation is implemented on the end effector path to smooth out the movement. The user defines areas of risk where the interpolation is applied. The needed information is automatically extracted, analyzed and converted into robot programs. Sensor data can be used to track the movement in real-time and make adjustments on the fly. The proposed tool is tested with two different experiments in which offline programs are created. The results of the tests show that the system is easy to use and within minutes an user can generate a robot program.

Baizid et al. [1] presented an Industrial Robotic Simulation Design Planning and Optimization platform named IRoSim. The platform is based on Solidworks and uses its API to extend the functionality. The goal of the platform is to offer an intuitive and convertible environment for designing and simulating robotized tasks. The platform includes various 3D models that are essential for developing any robotized tasks including different types of robotic arms. Besides aiding in the creation of the robotic program, the platform can be used to check the reachability of the end effector, to simulate the motion, and to validate the trajectory to avoid possible collisions. These can be used for time optimization and collision avoidance of the robotic task designed with the platform.

These approaches are aimed at professional users and want to support them in creating more complex robot programs. In contrast, our approach is aimed a end-users and targets a specific, albeit customizable toolkit.

2.2 End-User Robot Programming Approaches

End-user programming of robots is receiving much attention from the research community as it is essential to deliver on the promises of robotics in many sectors including both social and manufacturing and maintenance applications. There is significant diversity in approaches to realize this.

CoStar [10] lets users program robots using physical demonstration or graphical behavior trees that can be quickly reconfigured to deal with other, but similar situations. The system is cross platform, building on the capabilities of ROS [12].

Hammer [6] uses a block-based language, inspired by Scratch [13], as part of an integrated development environment for robot programming that runs on a tablet. It also includes the option of a teaching pendant, which allows direct control over a virtual model over the robot.

Pedersen and Krüger [11] propose an approach that links body poses to specific pre-programmed skills. These skills are higher level robot tasks that combine different robot actions and observations such as object detection. End-users can than use the gestures (in combination with a graphical user interface) to program the robot.

Orendt et al. [9] confirmed that the use of One-Shot Programming by Demonstration (programming a robot with a single demonstration) can be effective and intuitive, especially for end-user participants that successfully accomplished the tasks in the experiment. The results held regardless of the instruction modality. This type of programming by demonstration, in combination with a graphical programming environment is also used in industry to deploy collaborative robots.

Sefidgar et al. [14] propose the use of situated tangible programming; programming a robot through tangible instruction blocks within its operating environment. They found that people can comprehend and create robot programs with little or no instructions.

In contrast to these approaches that aim to make it easier to program robots, our approach is a step towards making programming the robot nearly transparent to the end user. In addition to these approaches aimed at programming robot behavior, there also several efforts to ease programming of interaction with robots. As the interaction with the robot is predefined in our proposed approach, we don't give an overview of these approaches but refer to recent work on this topic, such as that of Van den Bergh and Luyten [2].

3 Envisioned Approach

People go to a Fab Lab or Makerspace to create physical things with all kinds of machines, such as laser-cutters or 3D printers. These machines require digital models of the physical forms to be created. When considering tool support for further support of this creation process, we tapped into this aspect and used a digital model of the artifact to drive robotic support. We want to minimize the required knowledge to use the approach to 3D manipulation in a desktop tool and assembly skills.

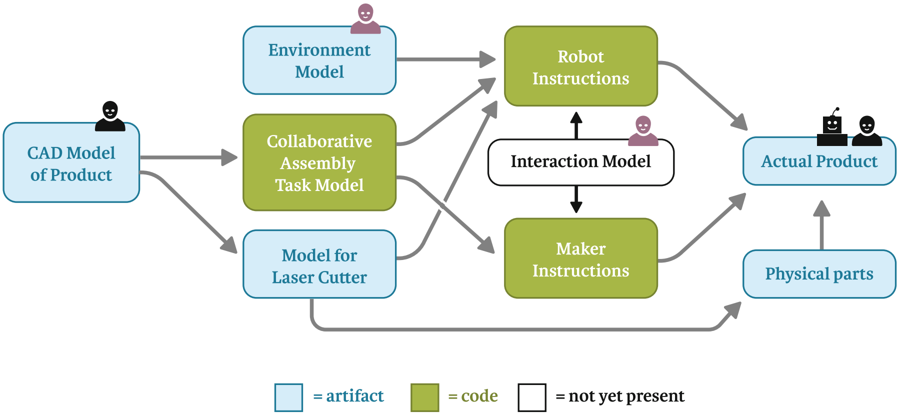


Fig. 1. Envisioned process for collaborative human-robot fabrication in Fab Lab. Input from human or robot is indicated with a symbol. Black indicates input per product, otherwise only configuration of the setup is required. Support in the current toolkit (Sect. 4) is indicated using differences in background color. (Color figure online)

Figure 1 gives an overview of our envisioned process and all artifacts that it uses or creates. All rounded rectangles represent artifacts. The background color indicates how the model is represented in our current prototype (Fig. 2). Most steps are thus completely automated, only two steps require input from the maker: CAD model creation and creation of the actual product. Two other steps (environment model and interaction model) are used to configure the environment.

The *CAD model* specifies the components that will used in the physical model and their configuration in 3D space. The order in which these components appear in the model is used to determine the order in which the components will be used to create the physical model. The creation of the CAD model, currently can thus be regarded as a virtual programming by demonstration exercise. The CAD model is thus used to create a *collaborative task model* of the assembly by allocating tasks to the robot (fetching plates and holding them in place) or the maker (fixing the plates using connection pieces), although this model is not externalized in a separate file. The plates used in a CAD model are also extracted and combined in a *model for a laser cutter*. This information is also used to determine where plates are to be picked up by the robot. To do this, the information is combined with information in the *environment model*. This latter model is also used to decide to which position the robot should bring a specific plate so that it is close enough to the product under construction to be fixed to it. The maker is guided through the whole assembly process with instructions provided by the tool. Currently a push on a small red button when the robot (or platform) should perform the next action but we plan to make this more flexible through the inclusion of an *interaction model*, which would enable implicit or explicit, perhaps even multimodal human-robot interaction. Several approaches can be considered to accomplish this [2].

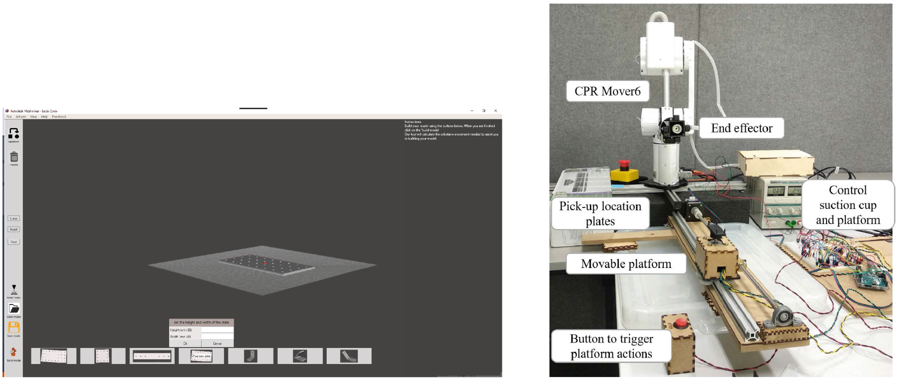
4 A Flexible Toolkit for Robot-Assisted Assembly Tasks

Our system includes a physical setup, consisting of a robotic arm and movable assembly platform, and a CAD tool that computes the collaborative assembly steps using the robotic arm. A set of predefined shapes of various sizes can be used to model a 3D object in the CAD tool; these shapes represent the different plates our setup can use. During assembly of the physical object, the robotic arm will pick the appropriate plate and put in position so the user can further assemble the modeled object. Assembly steps that required "a third hand", e.g. screwing together two plates under a specific corner, can now be completed much more easily. Furthermore, the robotic hand is instructed on the order and placement of the plates by the CAD tool, and requires no further programming or intervention by the user.

Physical Setup. Our current setup is based on the Commonplace Robotics Mover6 robotic arm[1] (Fig. 2b). This robotic arm has a relatively large reach (up to 80 cm with the default end effector) and has 6 degrees of freedom. The custom end effector has a bellows suction cup to pick up and hold the plates. There are four small cushions to keep plates perpendicular to the end effector.

The position of the movable platform and the robotic arm need not be perfect. It is sufficient if these are close enough together. The movable platform and the robotic arm allow small manual adjustments. For example, slightly moving the robotic arm by hand, rotating the model or tilting the model.

[1] https://www.cpr-robots.com/products/mover6.html.



Instructions:
Build your model using the buttons below. When you are finished click on the 'build model'
Our tool will calculate the robotarm movement needed to assist you in building your model.

CPR Mover6
End effector
Pick-up location plates
Control suction cup and platform
Movable platform
Button to trigger platform actions

(a) Tool on PC

(b) Physical robot setup

Fig. 2. Two parts of the current toolkit: makers use the tool on PC (a) to create CAD models. The tool guides assembly with support from the robot setup (b).

CAD-like Toolkit. A CAD-like tool was built on top of Autodesk meshmixer (Fig. 2a). In this tool the user can build the desired model by selecting the supported primitives. It supports positioning plates at a 45, 90 or 135° angle, the ability to create custom sized plates, and automatic extraction of a plates file that can be used to laser cut the required plates of the model.

To support plates at different angles the code that automatically calculates the instructions for the robotic arm should not only take the position of the plate into consideration, but when a plate is angled also any plates beneath it. To prevent hitting that plate, the robotic arm may be required to hold the plate from the other side.

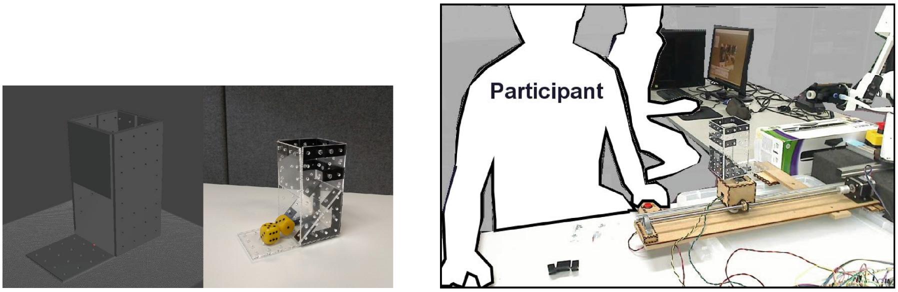
In the toolkit the user can specify the dimensions of the plate and place it into the model. These plates should be custom created using a laser cutter when the user wants to build the model. The suction cup end effector that is used with the Mover6 robotic arm, enables it to pick up plates without first having to attach metal pieces to these plates. The robotic arm can pick up plates directly from the original laser cut plate, if placed on a fixed position next to it. The 6 degrees of freedom allow the robotic arm to move freely over the plate, which isn't possible with the Arduino Braccio robotic arm.

When a user wants to build a model that uses custom size plates, he is required to laser cut the plates first. To make the toolkit usable for as many people as possible, automatic creation of a plates file is supported. This file can be used directly with a laser cutter to laser cut the plates. The plates are placed next to each other, the first plate is the base plate, the second plate the second plate in the model and so on. Around all the plates a buffer of 20 mm is created, which is used to hold all the plates in the correct position when placed next to the robotic arm.

5 Evaluation

5.1 Method

A formative study was conducted to evaluate and further refine our approach. The goal of the user study was to explore the overall usability of building a model in collaboration with a robot arm. The overall process of assembling a physical object of a model with the robot was tested. Our study started with a brief overview of the design tool, after which participants assembled a dice tower model in collaboration with the robot (see Fig. 3a). When the participant finished the object, we asked the participant to fill in a questionnaire followed by a structured interview. We selected the dice tower as the physical object to be made, because it requires assembling plates at an angle that is not perpendicular, uses various custom plate sizes, and is basically more complex than a basic tray but still feasible to assembly in a limited amount of time.



(a) Dice tower model

(b) Physical setup

Fig. 3. The dice tower model (a) that participants built with the setup (b).

Participants and Apparatus. The study was conducted with four participants (all male and between 23 and 35 years old) that have some familiarity with personal fabrication. Three participants had constructed a physical object in a Fab Lab setting at least once. Three of the participants were computer science students.

During the user study, we recorded the participants using a webcam facing the user. This webcam was connected to a second computer setup behind the participant (Fig. 3b). On this computer, the observer made notes in real time using Techsmith Morae Recorder[2].

Procedure. The user study consists of the following three parts: a study introduction, the assembly process, and a questionnaire and interview to close.

[2] https://www.techsmith.com/morae.html.

Study Introduction. The study starts with getting the informed consent of the participant. Next, the researcher familiarizes the participant with the design tool. They demonstrate how the tool can be used to design a 3D model of a physical object. They open the model of the dice tower (Fig. 3a) in the design tool on behalf of the participant, and inform them about the instructions that are shown during the assembly process, and how progress during the assembly process is visualized by the tool.

Assembly Process. The researcher starts the assembly process by attaching the base plate of the model on the platform. The last instruction the participant receives is to start with attaching the corner pieces to the base plate, as shown in the tool. All other instructions are given by the design tool, which the participant assembles the model without further guidance. During the assembly process, the researcher annotated the video recordings in real time to ease analysis after the experiment. The participants are allowed to ask questions during the process if they are not sure how to proceed.

Questionnaire and Interview. After the participant finishes assembling the model, a one-page questionnaire is given. The questionnaire asks about their connection to a Makerspace and about their experience during the assembly:

Q1 The robot arm was useful during assembly.
Q2 The instruction in the tool are clear.
Q3 Building a model was easier because the robot holds plates at the right spot.
Q4 Your role during assembly is clear.
Q5 The robot's role during assembly is clear.

Answers to the latter questions are on a Likert scale from 1 to 5 (fully disagree - fully agree). Each experiment ends with a short, semi-structured interview including open ended questions on their experience.

5.2 Results

Observations. Most completion times were similar (20, 21, 23 min). One participant was faster (15 min), but he had a less polished result. Time was spent between looking at instructions and doing the assembly. In case of doubt, additional time was spend reading instructions.

The robotic arm holds the plate in the required position, after which the user attaches the plate to the model. The idea is that the user attaches the plate with just enough bolts so that the plate stays in place, after which the robotic arm can move away. Moving the robotic arm out of the way gives the user more space to work. This was not clear to most of the participants, especially in the first step. Some going as far as to attach all the corner pieces on the plate before moving the robot arm away. The participant stated the instructions in the tool were clear, but they experienced two main problems during the assembly.

All participants experienced a problem when the robotic arm picks up a plate. It waits with holding the plate in the desired position until the user presses the step button again. This step allows the user to make a last check before continuing with the next plate. This was explained in the instructions:

```
The robot is picking up the next plate, it's finished when
   ↪ holding the plate.
The platform moves to the required position, it's finished
   ↪ after moving.
-------------------------------------------------------------
Press the STEP-button when the previous step is completed, to
   ↪  start the next step in the build process.
```

This explanation was however not clear to any of the participants. They expected the robotic arm to pick up a plate and hold it in one action. Thus when the robotic arm stopped they were not sure what to do. Some of the participants thought they had to attach corner pieces to the plate already. Two participants asked the researcher how to proceed in this step. One of them later indicated that he did not notice the instructions immediately because he was not sure if the step was finished. The other participants read the instructions again and saw that they should press the step button again.

A second problem occurred when the participants were finishing the active step, by tightening the last-added plate and placing the required corner pieces. During this step, the user is allowed to rotate the model to a position in which he can better reach it. The user however has to return the holder to the original position, before starting a new step. The stepper motor that is used in the movable platform does not know whether the platform rotated. This means that the tool assumes to execute rotations from the original position. The instructions in the tool do not explain this to the user, which is why three of the participants expected the platform to always rotate to the correct position. One of the participants did not rotate the platform at all because he expected that the platform should stay in its original position.

Questionnaire and Interview. Participants rated the overall experience of building a model with the robot arm positively (3x 4, 1x 5 on Q1). They all experienced the robot as helpful during the assembly (same ratings as Q1). Three out of four would use the setup again to build another model. The fourth participant would prefer not using the robot, but he saw potential for people with less technical knowledge or for children. This correlates with the answers to Q3: 1x 3, 2x 4, 1x 5. Participants generally agreed that the roles were clear in Q4 and Q5; only one rating of 3 for clarity of the human role, all other ratings were 4 or 5. No one experienced the system as too slow, when asked about it. One participant read the instructions and prepared for the next step while the robot was picking up the next plate. All the participants would use the design tool to design their own 3D model. One participant indicated that it would be easier than creating a 3D design from scratch using other CAD software.

All participants indicated that they sometimes had problems inserting a bolt into a hole, because the end effector was in the way. Especially on the smaller plates. A second problem that the participants indicated was with the very small bolts and nuts, which made it hard to tighten them. The problem was the most common in combination with the corner piece at a 45° angle. These corner pieces have little room left to insert the bolt or hold the nut in the right place. Some of the positions in which a corner piece had to be placed were hard to reach with your fingers, which also made it harder to attach the plate.

Summary of Results. The study that was executed was only a small one, but it still gave some valuable information. It showed that there is potential in using such a setup to aid users in building a model, even for more technical users. The way the robot arm and the movable platform work together to show the user how the model should be built, is helpful. One of the participants even indicated in the interview that he experienced it as building with a third hand.

The results of the user study indicate that the position of the robotic arm and the platform do not have to be perfect, as long as manual adjustments are possible. During the study, an early indication of problems associated with the assembly process occurred. Most of these problems can be fixed by improving the instructions and streamlining the build process. The participants provided insights in improvements that can be made to the tool and the process.

6 Discussion

The current prototype toolkit and the preliminary evaluation indicate the proposed approach is feasible. It is possible to generate all instructions for both maker and robot in a collaborative human-robot assembly process based on a 3D CAD model and an environment description. The resulting collaborative human-robot application is appreciated by potential users in the target group.

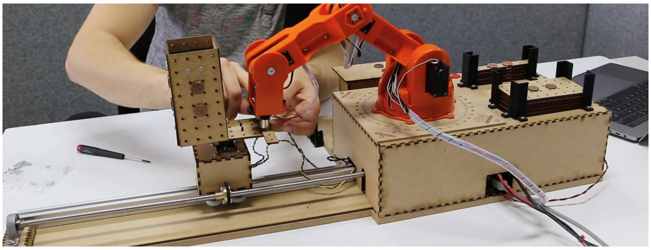


Fig. 4. The setup of an early prototype with the Arduino Braccio robot.

The results of the experiment indicate however that further work is needed to come to an intuitive walk-up-and-use version. This may require custom interaction possibilities, which can be provided through the envisioned inclusion of an interaction model. An example of a small change could be the addition of a soft button on the end effector of the robot that allows the maker to indicate that the robot no longer needs to hold the plate near the product under construction. This kind of interaction is similar to push interactions that are explored in industrial collaborative robot (cobot) applications. Cobots, however, typically have such sensors already built-in as part of their safety system.

The empirical results also indicate that requirements for speed and precision were met by the current prototype and cobots, which support higher speeds, more accurate movement, but at a much higher cost, may not be required to deploy robots in this specific type of application. It may even be possible to reconsider even cheaper robots, such as the Arduino Braccio, which was used for the creation of the initial, more basic, versions of the toolkit (Fig. 4).

The literature available on work instructions in the professional environment to address problems with the instructions for the makers may also be useful in this non-professional environment. Haug [5] proposed a framework for work instruction quality, while several studies are available that provide more information on which modalities can be best used to provide instructions. Funk et al. [4] evaluated some alternative modalities from which projection seemed to be most promising, but the main drawback in the study of e.g. a tablet solution may be less relevant in the current setting.

7 Conclusion

We presented a toolkit that contributes a new approach to program robots. It complements existing approaches that allow programming by demonstration with direct robot manipulation or without, tangible programming, mobile programming, or use of end-user programming languages such as Scratch. The approach embeds the programming of a human-robot collaboration activity within activities that the robot users naturally perform as part of their activities.

The tool may benefit from additional refinement: improved presentation of instructions as well as integration of an interaction model. The results obtained with this minimal effort approach are promising. Definitive conclusions on the viability of such approaches are however subject to further research. While the presented approach naturally fits the use of robots in assembly applications, the overall idea may even be applied in other domains, such as social robotics.

Acknowledgements. This research was partially supported by Flanders Make, the strategic research centre for the manufacturing industry. We thank the reviewers for their constructive comments.

References

1. Baizid, K., et al.: IRoSim: Industrial Robotics Simulation Design Planning and Optimization platform based on CAD and knowledgeware technologies. Robot. Comput. Integr. Manuf. **42** (2016). https://doi.org/10.1016/j.rcim.2016.06.003
2. Van den Bergh, J., Luyten, K.: Dice-R: defining human-robot interaction with composite events. In: Proceedings of the ACM SIGCHI Symposium on Engineering Interactive Computing Systems, pp. 117–122, EICS 2017. ACM, New York (2017). https://doi.org/10.1145/3102113.3102147

3. Duelen, G., Bernhardt, R., Schreck, G.: Use of CAD-data for the off-line programming of industrial robots. Robotics **3**(3), 389–397 (1987). https://doi.org/10.1016/0167-8493(87)90055-6
4. Funk, M., Kosch, T., Schmidt, A.: Interactive worker assistance: comparing the effects of in-situ projection, head-mounted displays, tablet, and paper instructions. In: Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing, pp. 934–939, UbiComp 2016. ACM, New York (2016). https://doi.org/10.1145/2971648.2971706
5. Haug, A.: Work instruction quality in industrial management. Int. J. Ind. Ergon. **50**, 170–177 (2015). https://doi.org/10.1016/j.ergon.2015.09.015
6. Mateo, C., Brunete, A., Gambao, E., Hernando, M.: Hammer: an android based application for end-user industrial robot programming. In: 2014 IEEE/ASME 10th International Conference on Mechatronic and Embedded Systems and Applications (MESA), pp. 1–6. IEEE (2014). https://doi.org/10.1109/MESA.2014.6935597
7. Neto, P., Mendes, N.: Direct off-line robot programming via a common CAD package. Robot. Auton. Syst. **61**(8), 896–910 (2013). https://doi.org/10.1016/j.robot.2013.02.005
8. Neto, P., Mendes, N., Araújo, R., Norberto Pires, J., Paulo Moreira, A.: High-level robot programming based on CAD: dealing with unpredictable environments. Ind. Robot Int. J. **39**(3), 294–303 (2012). https://doi.org/10.1108/01439911211217125
9. Orendt, E.M., Fichtner, M., Henrich, D.: Robot programming by non-experts: intuitiveness and robustness of one-shot robot programming. In: 2016 25th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN), pp. 192–199. IEEE (2016). https://doi.org/10.1109/ROMAN.2016.7745110
10. Paxton, C., Hundt, A., Jonathan, F., Guerin, K., Hager, G.D.: CoSTAR: instructing collaborative robots with behavior trees and vision. In: 2017 IEEE International Conference on Robotics and Automation (ICRA), pp. 564–571. IEEE (2017). https://doi.org/10.1109/ICRA.2017.7989070
11. Pedersen, M.R., Krüger, V.: Gesture-based extraction of robot skill parameters for intuitive robot programming. J. Intell. Robotic Syst. **80**(1), 149–163 (2015). https://doi.org/10.1007/s10846-015-0219-x
12. Quigley, M., et al.: Ros: an open-source robot operating system. In: ICRA Workshop on Open Source Software, vol. 3, p. 5 (2009)
13. Resnick, M., et al.: Scratch: programming for all. Commun. ACM **52**(11), 60–67 (2009). https://doi.org/10.1145/1592761.1592779
14. Sefidgar, Y.S., Agarwal, P., Cakmak, M.: Situated tangible robot programming. In: Proceedings of the 2017 ACM/IEEE International Conference on Human-Robot Interaction, HRI 2017, pp. 473–482. ACM, New York (2017). https://doi.org/10.1145/2909824.3020240