Universal Industrial Interface - Mobile

Ciontos, Andreea-Emilia; Sarivan, Ioan-Matei; Schou, Casper

Published in:
Procedia Manufacturing

DOI (link to publication from Publisher):
10.1016/j.promfg.2020.01.050

Publication date:
2019

Document Version
Publisher's PDF, also known as Version of record

Available online at www.sciencedirect.com

ScienceDirect

Procedia Manufacturing 38 (2019) 391–399

Procedia MANUFACTURING

www.elsevier.com/locate/procedia

29th International Conference on Flexible Automation and Intelligent Manufacturing (FAIM2019), June 24-28, 2019, Limerick, Ireland.

Universal Industrial Interface - Mobile

Andreea Ciontos, Ioan-Matei Sarivan, Casper Schou

Aalborg University, 9000, Denmark

Abstract

Robot programming can be a tedious task, especially in industrial environments, where different types of robots are required and in flexible automation lines, where robots need to be frequently reconfigured. There are several shortcomings to be solved in this regard. The main issue being time efficiency. This paper presents a novel solution proposed for robot programming, a mobile application, called Uii Mobile, meant to introduce the smartphone on the shop floor, while facilitating robot programming for operators with basic technical knowledge. The use of Uii Mobile's intuitive graphical user interface (GUI) reduces the time a worker allocates for learning different programming languages. Uii Mobile also enables faster reconfiguration of robots for small, flexible tasks, such as pick and place. Nonetheless a user study using the System Usability Scale (SUS) rates Uii Mobile with a score of 75.38/100 which makes it an above average system, regarding its usability.

Keywords: industry 4.0; IoT; smartphones; industrial robots; flexible automation; human-machine interaction; usability

1. Introduction

Smartphones are used by people of almost all ages across the globe, and for many it has become the primary technical interface for accessing information, entertainment and communication. Smartphones have become an important part of our daily, social life, and they provide a familiar interface easily extendable by millions of available apps. Lately, by connecting to Internet of Things (IoT), smartphones have become interfaces for controlling or at least monitoring various devices (e.g. wireless speakers, vacuum cleaners, etc.). [1] However, whenever stepping into a factory with automated manufacturing lines, smartphones become obsolete. Instead, non-standard interfaces are used to operate machinery; including robots. The problem is, that firstly, it requires training for the operators to use this equipment, as the interface of most robots are not intuitive. Secondly, often all these controllers have their own programming language, and whenever switching from one robot to another, one needs to change language as well as

10.1016/j.promfg.2020.01.050

interface [2]. In response to this issue, we propose a simple, intuitive teach-pendant-replacement running of an Android smartphone. The smartphone application is part of a framework called *Universal Industrial Interface* (Uii) and it provides an add-on interface for controlling and programming industrial and collaborative robotic manipulators. In the spirit of Industrie 4.0, the application is cross-vendor compatible and is applicable to any robot setup. The interface of Uii is simple and intuitive and significantly reduces the need for extensive training prior to programming robots; hereby, enabling robotics novices to program robots.

The structure of this paper is the following: Section 2 presents related research in the field, followed by Section 3 which presents the Uii Framework, including work on Uii Desktop, a computer software, as well as Uii Mobile, the Android application mentioned above. Further on, Section 4 presents a user study conducted to test the usability and user experience related to the system, together with results and findings. Finally, Section 5 concludes the paper.

2. Related work

As seen through the emergence of Industrie 4.0, Smart Manufacturing and similar initiatives, the need for more agile manufacturing solutions is imminent [3]. This is not least true within the field of industrial robotics, where robots will have to take on smaller and smaller batch sizes with ever increasing variance [4]. As a result, more intuitive and faster means of reconfiguring robots is needed as opposed to traditional robot programming using non-standard teach pendants that are different across devices from different producers [5]. Furthermore, since most industrial robots today feature proprietary interfaces and programming languages, it becomes challenging for operators to learn and operate robots of various brands.

One solution explored by several researchers [6-10] is to create an intuitive robot programming tool using the familiarity of a smart phone. Not only is the modality of interacting through the touch screen is familiar to most, but the smart phone also provides a suite of additional sensors. Both Lambrect et al. [6] and Yepes et al. [7] utilize the accelerometer of a smartphone to create an intuitive way of jogging the robot end-effector by tilting and turning the phone itself. Yepes et al. [7] conducted a small usability study in which 72% of the participants found a smart phone an adequate tool for controlling a robot.

In extend of controlling the end-effector, a smart phone application must be able to store and execute tasks to successfully replace existing teach pendants. Jan et al. [8] presents the Android application called “Smart teaching pendant” which offers both a teaching mode and an operation mode. In operation mode the “Smart teaching pendant” allows the operator to play, pause and skip the current task. The tool is only tested against a simulated robot, which is perhaps why no information on how the programming is conducted is included.

Su and Young [9] presents a tablet-based interface offering both robot control and task programming and execution. The interface is tailored for a 6-DOF robot from Industrial Technology Research Institute in Taiwan, but it works both online with the real robot and offline with a simulated robot. As in [8], very little focus is put on describing how the programming and subsequent execution of tasks is done.

Mateo et al. [10] presents “Hammer”; a comprehensive robot interface implemented in Android and running on a tablet. The interface features both robot control, task programming and task execution. Furthermore, it includes a graphical drag-and-drop programming IDE and augmented reality. In [10], “Hammer” is applied together with a COMAU robot but intended for use with any robot brand. However, the paper does not disclose details on the implementation or the interface towards the hardware.

Lambrecht et al. [6] introduce a standardized interface between their smart phone application and the specific hardware. It consists of a generic robot language with a set of generic functions used by the application. When invoked, these generic functions are then translated into the language of the specific robot controller by an interpreter.

Similar to the work presented in [6-10] we propose a smart phone application serving a as a replacement for the teach pendant when programming and operating industrial robots. Contrary to the work presented above, we emphasize the system level architecture and the use cases of both programming and executing tasks on an industrial robot. Our framework is independent of robot vendor, which we demonstrate with two robots; a KUKA and a Universal Robots. Lastly, we conduct a study with participants having varying robotics knowledge to explicitly assess the usability of the smart phone application.

3. Universal Industrial Interface framework

The smartphone app itself is only one part of the solution that enables industrial robot control through a smartphone. The overall framework, called *Uii framework*, consists of two primary pieces, as illustrated on Fig. 1. Uii Desktop is the core of the system and provides the communication towards the equipment (robot, gripper, etc.) along with the programming and execution engines. Uii Desktop is intended to run on a PC or Raspberry Pi with a wired connection to the equipment over e.g. LAN. Uii Mobile runs on an Android smartphone, and serves as the front-end towards the user. Uii Mobile and Uii Desktop communicate via a TCP/IP connection.

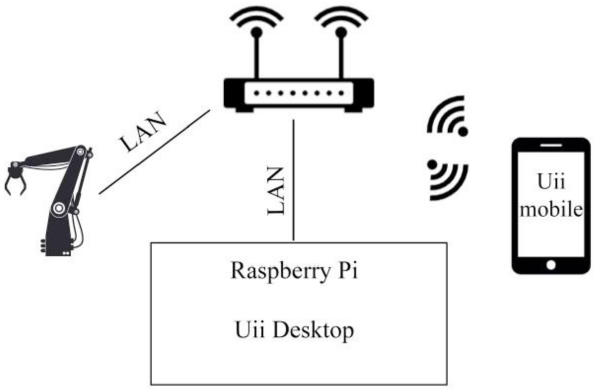

Figure 1. Uii Framework setup

3.1. Uii Desktop

Uii Desktop is a vendor independent, cross-platform software meant to facilitate robot programming by introducing a GUI based programming approach rather than code. Uii Desktop was developed previously in [11], and it in itself also provides a front-end user interface, through which robot programming and task execution can be done. However, in this paper we propose an extension of Uii Desktop, allowing a remote front end to be used from a smartphone (Uii Mobile). Uii Desktop as well as the desired manipulators and grippers must be connected to the same local area network (LAN) in order to establish a TCP/IP connection which is necessary for communication and data transfer. In this case the robot's controller hosts the server sockets which receives requests from the client sockets located on the computer running Uii Desktop. The low-level communication between the hardware is done through specific drivers. Fig. 2 is a representation of how specific hardware is integrated with Uii Desktop.

The user interface of Uii Desktop is split into tabs for different functionalities. The fist tab is the welcome tab, which is meant for establishing connection to a robot. A robot configuration has a name and it is composed of one manipulator and one gripper. The user can create a new robot configuration in the welcome tab by providing a name for the robot configuration, by selecting what type of manipulator the configuration has (Kuka, UR5 etc.) and what type of gripper the robot has (Schunk, Robotiq, pneumatic etc.), and by inserting the IP address for each device. After connecting to a specific robot, a new tab is automatically displayed, called the Robot tab, which contains arrow shaped buttons for manipulating orientations and positions of the manipulator, as well as gripper actions. Uii Desktop offers the option of connecting to multiple robots at the same time via a multi-threading method. In this case, the program opens a jogging tab for each connected robot. This jogging interface makes it possible to control a robot from a computer, such as a Raspberry Pi equipped with a touch screen, or a regular personal computer.

In extension of Uii Desktop following the development of Uii Mobile, the welcome screen contains another option, meant for remote control. At the click of a button, a server is opened for a robot, which displays a new tab with a QR code containing parameters needed to establish connection to a robot, as well as status lights which serve as connectivity information for the manipulator, gripper and the smartphone. The status lights light up green, yellow and red in case of ready to execute a command, the component is executing a command and fault respectively. This function is meant to give a visual feedback to the user, as the status lights are visible from a distance and an operator can attend in case of need. Fig. 4 shows the user interface of Uii Desktop when being used in remote control mode.

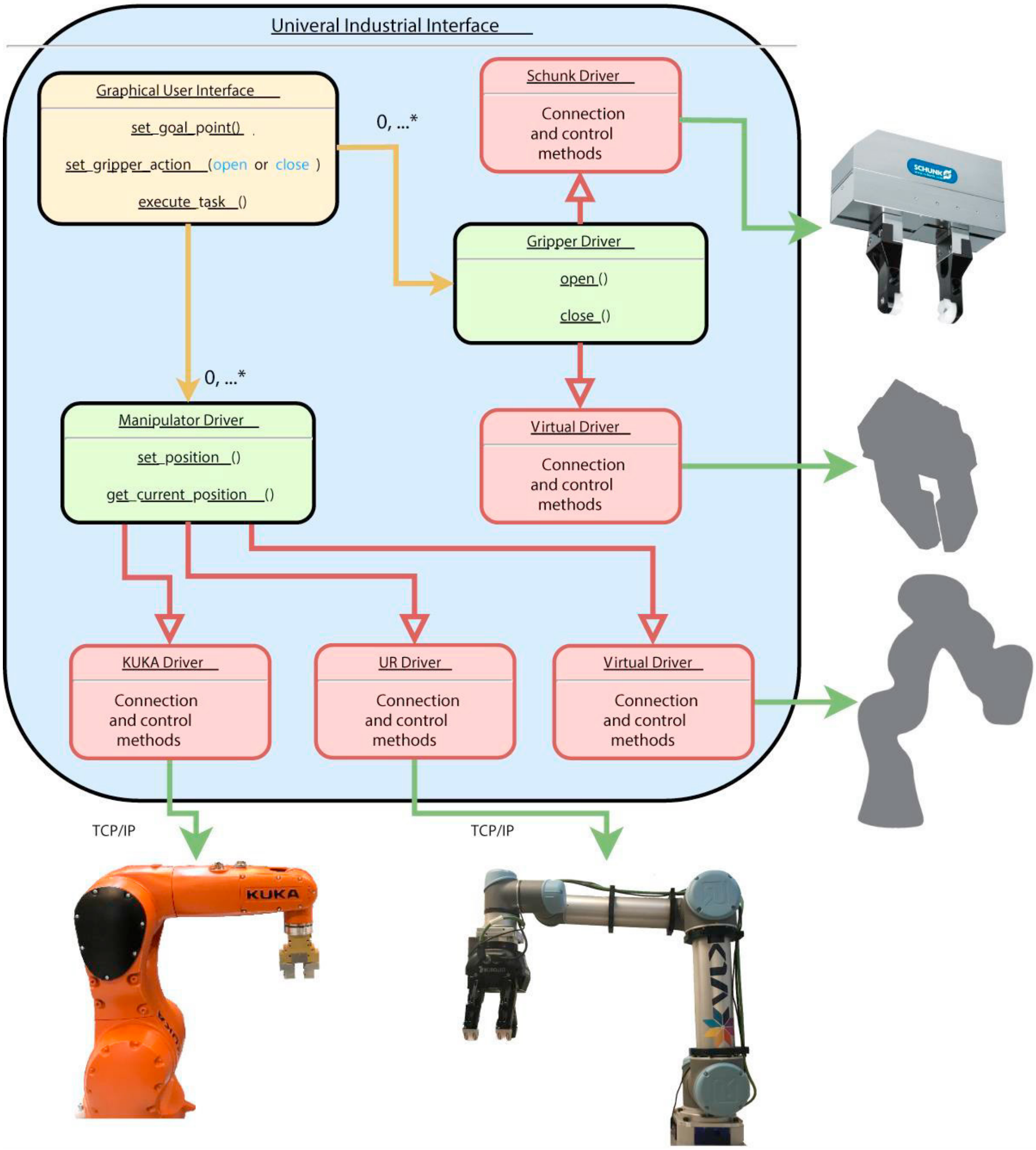

Figure 2. Uii Desktop architecture.

3.2. Uii Mobile

Uii Mobile serves as a portable front-end for the Uii framework. It allows the operator to connect to any robot running Uii Desktop by scanning the QR code displayed in in the user interface of Uii Desktop when put into remote mode. The scanning of the QR code ensures the operator are present at the desired robot cell before he/she assumes control. The QR code contains the IP address of the computer on which the Uii Desktop program is running, implicitly, this provides connection to the current online manipulator and gripper, the second item in the QR code is the server port number and the connected robot name. Thus, the user can control the robot from their smartphone.

The Uii Mobile app is developed in Android Studio, a platform designed for developing android applications, including front end and back and functionalities. An important aspect when developing a tool to facilitate robot programing is the intuitiveness of the system. In this case, the graphical user interface plays a major role in achieving the desired effect. The Uii Mobile GUI contains seven screens with layouts inspired by the Universal Robots teach pendant. These screens can be observed in Fig. 3 (a-g).

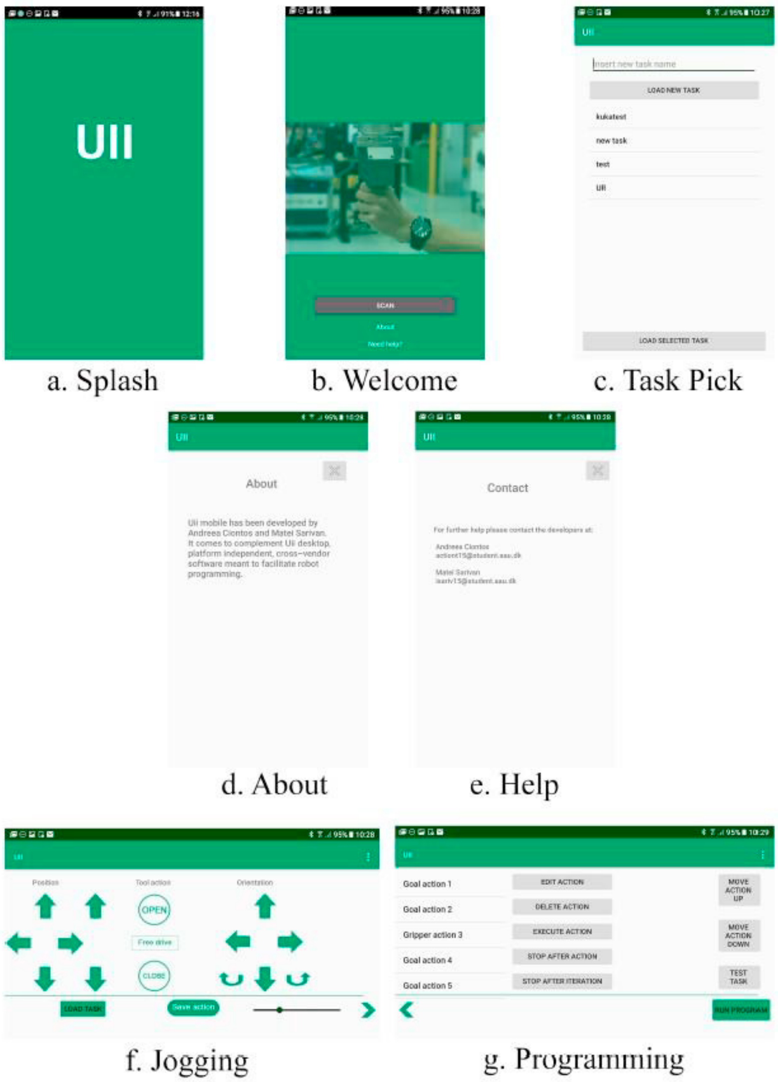

Figure 3. Uii Mobile layout

Splash Screen

The splash screen (Fig. 3.a) is meant to notify the user that the program is launching. In this case, the splash screen is mainly used for enhancing the visual identity of Uii Mobile. This screen is active for four seconds, after which the Welcome screen automatically pops up.

Welcome Screen

The welcome screen, (Fig. 3.b) has three main elements in forms of buttons. The first button, 'Scan', when pressed it opens a camera screen ready to scan the QR code containing pairing information with Uii Desktop and it redirects automatically to the Task Pick screen. The second button is called 'About', and when it is pressed it opens the About screen, (Fig. 3.d), which contains a short information about the application. And finally, the 'Need help' button which opens the Help screen, (Fig. 3.e), which contains the contact information of the developers for further guidance.

Task Pick Screen

The Task pick screen, (Fig. 3.c) becomes available after the pairing with Uii Desktop is successful. The Task Pick screen offers the possibility for a new task file to be created, or an existing task file to be loaded. Once the user has loaded a task file, Uii Mobile loads the Jogging screen.

Jogging Screen

The jogging screen (Fig. 3.f) is split up in four main areas. The first area is dedicated to the control of the position of the robot, with six arrows. Similarly, the second area contains also six arrows to control the orientation of the robot. The third area contains three buttons for the gripper actions and the free drive option (which toggle kinesthetic teaching on or off for collaborative robots). The fourth area connects the Jogging screen and the Programming screen, in the form of a menu bar. The user can switch to the Programming Screen by pressing a small arrow in the bottom right and bottom left corners. It also contains a speed slider, where the user can modify the jogging speed of the robot. There is also a button available for returning to the Task Pick screen if the user wishes to create or load another task file. The last button is a 'Save action' button which must be pressed in order to save desired via points and gripper actions. The last action which was performed by the user is saved. For example: if the user jogged the robot right before pressing "Save action", the current coordinates of the gripper are saved. If the user closed the gripper before pressing the button, the current state of the gripper is saved as an action.

Programming Screen

The programming screen (Fig. 3.g) presents the user with an overview on the programmed actions in the current task file. Here, two of the available buttons are dedicated to moving up or down a selected action in the list. The other buttons serve for deleting, editing, executing a selected action. Furthermore, the user has the possibility to test out their program by pressing the button called 'test task' and they can stop it by pressing 'Stop after action', which will stop the program after the current action is performed. The 'Run program' button will run the program indefinitely and the user can press the 'Stop after iteration' button which, as the name suggests will stop the program after the current iteration.

3.3. Deployment

For each robot cell that needs to be controlled using the Uii framework, one instance of Uii Desktop needs to be running. Uii Desktop will run on multiple OS platforms, including both Linux and Windows. It is light-weight, and will also run on a simple Raspberry Pi, or similar single-board computers. This makes the addition to an already existing robot cell fairly small and cheap. UII Desktop comes preinstalled on an SD card that can be inserted in a Raspberry Pi computer. At this moment, UII Desktop is compatible with KUKA robots and UR robots together with Schunk, Robotiq and any I/O controlled grippers (pneumatic or electric). When booting up the Raspberry Pi, Uii Desktop is automatically launched.

The first time Uii Desktop is launched, it needs to be configured for the specific hardware of the robot cell it is connected to. This requires the operator to ensure that the robot, gripper(s) and the Raspberry Pi are all connected on the same local area network. Through the graphical user interface of Uii Desktop, the operator adds the connected hardware by selecting the brand of the manipulator, inserting the IP address of the manipulator, the brand of the gripper and inserting the IP address of the gripper. Once all the available robotic manipulators are added in Uii, the operator can select the one they desire to program. If another robot is connected to Uii and has a task running, the programming process of the selected robot will not interfere with the running one.

When put into remote-control mode, Uii Desktop will display a status screen with three large boxes, one indicating the status of the robot, one indicating the status of the gripper, and one indicating the status of the taskcontrol. In addition, a QR-code is displayed, see Fig. 4. The three large status indicators make is possible for operators in the area to quickly monitor the status of the robot from a distance. The QR-code is used to “connect” with Uii Mobile.

When the QR-code is scanned using Uii Mobile the operator is able to monitor the current task running on the robot. This is useful to track detailed progress of the robot or in case of error-correction. Another option is to execute a different task, modify an existing task, or start programming a new task. Since the task programming and execution engines are running on Uii Desktop, the operator can disconnect (intentionally or un-intentionally) without it affecting any executing tasks or ongoing programming; thus, once any instance of Uii Mobile connects again, ongoing programming can be resumed. It is a key characteristic of the framework, that an instance of Uii Mobile is not bound to one specific instance of Uii Desktop. This supports the fundamental concept of a smartphone; it is personal. Each operator carries a personal smartphone through which he/she can interact with all the robot cells in the production.

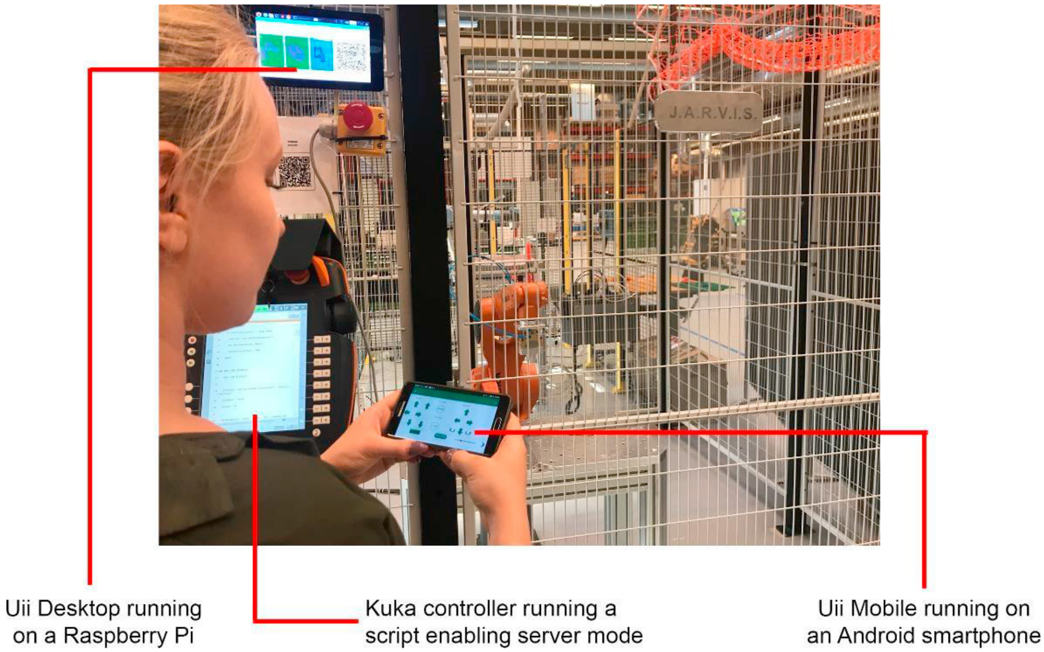

Figure 4. Operator interacting with the Uii Framework

4. Usability Study

In order to assess the intuitiveness and the user experience when using the Uii framework with the Uii Mobile app, a user study was conducted with 26 test participants. The participants ranged in from people with no technical background to people with expertise in robotics. It was intentional to enroll people with varying experience in robotics, as the Uii framework is intended not only for novices, but also for experienced users. To assess the user experience, the System Usability Scale (SUS) was used [12]. SUS is a UX measurement tool widely used in systems engineering.

One of the key benefits when using the SUS is that it is a reliable tool even when applied to small samples of test subjects. The questions and the results of the SUS for Uii Mobile can be found on at: https://tinyurl.com/y2mdqt85 .

4.1. Protocol

The participants carried out the test individually, with no interaction between them. The procedure for each of the participants was as follows:

Instruction/training

A sheet of instructions has been presented to each participant. This assured that everyone gets the exact same information prior to the test, thus avoiding the contamination of the data.

Task Programming

They were asked to program a task using Uii Mobile. In order to do this, the subject had to:

- Start Uii Desktop and connect to a robot and gripper via Uii Desktop.
- Start the Uii Mobile and connect to Uii Desktop.
- Program a simple pick and place task.
- Modify several points in the existing program.
- Execute the programmed task.

Survey

After conducting the test, the subjects had to fill in the SUS survey. The survey contained 10 questions on the SUS scale, with options from 1 (Strongly agree) to 5 (Strongly disagree). These questions follow the standard SUS format which can be found at [13]. To acquire qualitative data as well, the subjects with robotics experience were asked to provide their personal qualitative assessment of Uii Mobile and to propose improvements.

4.2. Results

When analyzing and interpreting the results of the SUS, the overall score can range between 0 and 100. If the system's score is below 68 it is considered a system with under average usability. And everything above the value of 68 is considered a system with over average usability. Uii Mobile scored an average of 75.38. More about SUS and result interpretation can be found at [13]. The participants found it easy to get acquainted to the system, and the majority considered they would not need the help of a technical person to use this app. Regarding the qualitative data, a major finding, which also complies with the requirements of the system, was that people with experience within the field of robotics found Uii Mobile to be easier to use than traditional teach pendants for fast reconfiguration of robots.

5. Conclusion

As mentioned in the introduction of this paper, the programming process of industrial and collaborative manipulators can raise problems in terms of time and finances. The main cause for this, as presented in this paper, is the lack of a standardized interface across robots regardless of their producer. In a flexible manufacturing system, the operators interact with robots (industrial or collaborative) in frequent manner. By shortening the time needed to reprogram them and by having a familiar interface for the operator, a new product can reach the market faster. Smartphones are devices familiar to almost everybody. Through the development of Uii Mobile on top of Uii Desktop, a successful attempt was made to integrate smartphones in the industrial production environment and obtain a standardized interface across robotic manipulators regardless of their producer. This paper only presented the use of the system on Kuka and Universal Robots manipulators, however, a library that contains as many devices as possible is a subject of further development of the Uii system as whole.

Nevertheless, the main objective of having an accessible interface for the operator is successfully met. This is confirmed by the SUS test made with 26 subjects that played the operator role (with and without prior training) using Uii Mobile for operating and programming a pick and place task on a robotic manipulator. The obtained SUS score of 75.38% is a strong evidence that Uii Mobile makes robot operation and programming accessible to anyone regardless of background and training.

When considering industrial robot programming there are of course the safety concerns regarding this solution. Uii is not meant to replace any component of the robotic manipulator as it is, but rather be an adjacent system that helps integrate multiple devices from multiple vendors under one single interface. Therefore, the operators, even if they use Uii or not, must be aware of the safety protocols regarding industrial robots. One functionality that can be introduced on Uii Mobile to improve the safety of the operators and of the manipulator itself is to ensure that the operator is in the visual proximity of the robot. This is yet another subject of further development of Uii.

Being a smartphone application, Uii can also be further developed in a viable IoT element. Data can be transmitted from the robots to the smartphone in real time over internet, therefore the robot cell can be monitored remotely at any time.

References

[1] H. Beigi, “Automation in manufacturing,” 1997.

[2] K. Fischer, B. Glavina, E. Hagg, G. Schrott, J. Schweiger, and H.-J.Siegert,Robot Programming, 08 2007, pp. 303–342 [3] Y. Lu, “Industry 4.0: A survey on technologies, applications and open research issues,” J. Ind. Inf. Integr., vol. 6, pp. 1–10, 2017.

[4] SPARC, “Robotics 2020 Multi-Annual Roadmap: Horizon 2020 Final release to call,” vol. 2016, p. 325, 2015.

[5] Z. Pan, J. Polden, N. Larkin, S. Van Duin, and J. Norrish, “Recent progress on programming methods for industrial robots,” Robot. Comput. Integr. Manuf., vol. 28, no. 2, pp. 87–94, 2012.

[6] J. Lambrecht, M. Chemnitz, and J. Krüger, “Control layer for multi-vendor industrial robot interaction providing integration of supervisory process control and multifunctional control units,” 2011.

[7] J. C. Yepes, J. J. Yepes, R. Mart, and Z. P. Vera, “Implementation of an Android based teleoperation application for Controlling a KUKAKR6 robot by using sensor fusion,” 2013.

[8] Y. Jan, S. Hassan, S. Pyo, and J. Yoon, “Smartphone Based Control Architecture of Teaching Pendant for Industrial Manipulators,” pp. 370– 375, 2013.

[9] Y.-H. Su and K.-Y. Young, “Effective manipulation for industrial robot manipulators based on tablet PC,” J. Chinese Inst. Eng., vol. 41, no. 4, pp. 286–296, 2018.

[10] C. Mateo, A. Brunete, E. Gambao, and M. Hernando, “Hammer: An Android based application for end-user industrial robot programming,” MESA 2014 - 10th IEEE/ASME Int. Conf. Mechatron. Embed. Syst. Appl. Conf. Proc., pp. 1–6, 2014.

[11] I. M. Sarivan, A. Batuev, A. Ciontos, O. Holtskog, E. Sivertsen, and C. Schou, Using Collaborative Robots As A Tool For Easier Programming Of Industrial Robots. Elsevier, 2018, pp. 201– 211.

[12] M. R. Drew, B. Falcone, and W. L. Baccus, “What Does the System Usability Scale (SUS) Measure?,” in *Design, User Experience, and Usability: Theory and Practice*, 2018, pp. 356–366.

[13] J. R. Lewis, B. S. Utesch, and D. E. Maher, “Measuring Perceived Usability: The SUS, UMUX-LITE, and AltUsability,” *Int. J. Human–Computer Interact.*, vol. 31, no. 8, pp. 496–505, 2015.