Available online at www.sciencedirect.com

ScienceDirect

Procedia CIRP 107 (2022) 1077–1082

www.elsevier.com/locate/procedia

55th CIRP Conference on Manufacturing Systems

Towards the integration of a pointing-based human-machine interface in an industrial control system compliant with the IEC 61499 standard

Antonio Paolillo*[a], Gabriele Abbate[a], Alessandro Giusti[a],
Šejla Trakić[b], Hilmo Dzafic[b], Artur Fritz[b], Jérôme Guzzi[a]

[a]Dalle Molle Institute for Artificial Intelligence (IDSIA), USI-SUPSI, Lugano, Switzerland
[b]nxtControl GmbH, Leobersdorf, Austria

* Corresponding author. Tel.: +41-58-666-6666. *E-mail address:* antonio.paolillo@supsi.ch

---

Abstract

In the context of Industry 4.0, we tackle the problem of how human operators can interact with a cyber-physical system at run-time. We focus on sporadic interactions where operators, normally occupied with other activities, need to communicate to the system a piece of information relative to a specif part of the plant, such as an anomaly. As a concrete instance of this problem, we consider an automated system composed of a robot loading packages on a conveyor belt that transports and sorts them. We argue that gesture-based interaction modalities offer important advantages for the operators in the considered scenario and can be deployed respecting the IEC 61499 standard.

Keywords: human-machine interface; IEC 61499; pointing; conveyor belt; cyber-physical systems.

---

1. Introduction

Machines are the base of manufacturing processes for their unquestionable efficiency in executing repetitive and demanding work. However, they are not limited to the mere execution of tasks, but are more and more asked to interact with human coworkers [17, 15]. In the context of Industry 4.0, it is expected to face situations where a human operator, who is in charge of other specific tasks in the manufacturing line, is far from the machine and needs to sporadically communicate information, such as anomalies. In particular, we focus on the following concrete example: a Cyber-Physical System (CPS) consisting of a gantry robot that picks and places packages on a conveyor belt that, in turn, sorts them to appropriate locations. Defective packages can be detected by a human operator and indicated to the CPS, so that they can be properly discarded from the process line using pneumatic diverters, see Fig. 1.

In such situations, gestures represent a sensible modality to interact with machines in industrial environments. Pointing gestures, in particular, are a convenient approach to implement a Human-Machine Interface (HMI) for our interaction purposes. In fact, the act of pointing is very natural for humans and is very effective to indicate locations in space [3]. Furthermore, it has been already used for robot control [6, 10, 11, 28] and in industrial scenarios [8, 18, 22, 25]. However, to be fully integrated and actually usable in real manufacturing scenarios, such an HMI has to comply with industrial automation standards.

In particular, we focus on how to integrate the HMI within an automation system compliant with IEC 61499. This standard is an event-driven, object-oriented, distributed approach used to implement industrial architectures [27]. It is based on networked function blocks synchronized through events. Its event-driven nature is particularly suited to the scope of our work since it matches the pointing-based HMI design and working principle, which is triggered by a specific event.

Our contribution is the analysis of a pointing-based HMI to be used in industrial environments, and a generic approach to integrate such HMI in IEC 61499 automation architectures. The remainder of the paper is organized as follows. Section 2 describes the pointing-based HMI, its advantages, drawbacks, and implementation to select packages on conveyor belts. Section 3 provides an overview of the considered CPS architecture based on a run-time environment compliant with IEC 61499. Finally, Section 4 discusses the integration of the pointing-based HMI

10.1016/j.procir.2022.05.111

This is a resupply of March 2023 as the template used in the publication of the original article contained errors. The content of the article has remained unaffected.

(a) (b)

Fig. 1. The manufacturing line considered in this work features a gantry robot in charge of loading packages on a system of conveyor belts. The packages are sorted into different unloading bays through pneumatic diverters (a). Using the pointing-based human-machine interface, an operator can communicate to the system possible defective packages (b). LED strips along the conveyor belts provide the user with visual feedback about the pointed location as a yellow cursor.

in the automation system, and Section 5 concludes the paper with final remarks and considerations about future work.

2. Pointing-based human-machine interface

2.1. A general approach for interaction

The pointing-based HMI considered in this work is built on a generic approach that uses a simple geometric model and very little sensory information. The model assumes to know basic biometric data of the operator, i.e., the length of the arm, the shoulder height, and the eye height. Furthermore, it is assumed that the operator points with their arm straight. Thus, from the measured orientation of the wrist, it is possible to compute the pointing ray that originates in the eye of the operator and passes through the tip of the pointing finger [4, 16, 19, 21]. The orientation is measured by a small and inexpensive Inertial Measurement Unit (IMU), which the operator wears at the wrist. Assuming that the location of the operator in the environment is known, the position of the pointed object is computed by intersecting the pointing ray with the environment. Given the highly structured geometry of manufacturing lines, it is reasonable to assume that the map of the environment is known.

Different technologies, such as computer vision [13] or radio frequency [24], can provide robust, device-free operators localization. In our context, pointing provides an interesting alternative method where operators are asked to point objects of known position: the operator is then localized at the pose that minimizes the error in the reconstructed pointed positions [9].

An interaction using such pointing-based HMI consists in a simple procedure [2, 12, 9], whereby an operator: (i) interrupts their activity and starts the interaction; (ii) if necessary, localize themself with the procedure described above; (iii) points to a location of the environment. After these steps, the pointed location in the environment is reconstructed by the system in real-time and can be used for further operations, e.g., to indicate a destination target to a robot [9].

2.2. Application to the selection of packages

The general approach described in the previous section can be instantiated to our specific use case [2, 12]. Here, an operator needs to select packages traveling on conveyor belts and pass this information to the CPS handling the packages. In this context, the reconstructed pointed location provided by the HMI described in Section 2.1 might not be accurate enough to disambiguate packages that are close to each other. In fact, non-modeled effects due to, e.g., a loose wearing of the IMU or the drift of the measurement, reduce the pointing performance. To overcome these issues, the conveyor belts are provided with LED strips that show a *cursor*, in real-time, at the reconstructed pointed location along the conveyor belts, see Fig. 1(b). This mechanism serves as visual feedback to the operator for correcting possible errors or inaccuracies, increasing the accuracy of the pointing location. The selection is thus performed by comparing the position of the pointing cursor with the package position provided by the CPS. If they are close enough during a short period (e.g., 1 second), the package is considered selected.

2.3. Analysis of the pointing-based HMI in the considered CPS

In this section we discuss advantages and drawbacks of the pointing-based HMI in the specific considered use case.

The HMI is very intuitive to use, requires little equipment, almost no infrastructure and is computationally light. These aspects facilitate the portability of the system that can be potentially applied in any industrial scenarios, with no need of complex customization. The use of the pointing-based HMI does not require the operator to hold specific tools; indeed, they can

This is a resupply of March 2023 as the template used in the publication of the original article contained errors. The content of the article has remained unaffected.

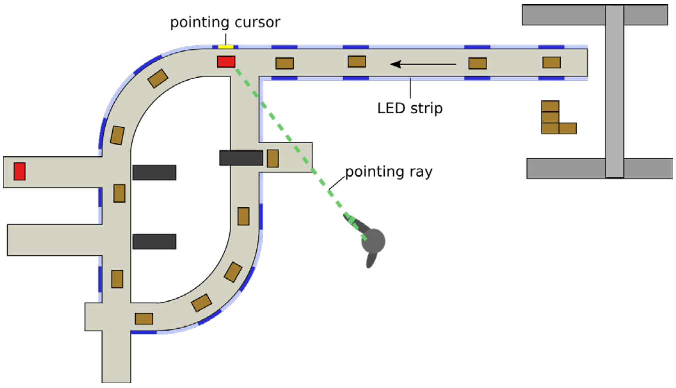

Fig. 2. Schematic of the top view of the considered CPS. The gantry robot, the conveyor belts, and the diverters are controlled by the IEC 61499 automation runtime. The pointing-based HMI, implemented in ROS2, handles the IMU and the LED strips and reconstructs the pointing ray.

do other (completely different) tasks when the interaction is not active. Compared to other classical HMIs using screens or displays, the pointing-based HMI allows a direct view of the current state of the CPS, with a beneficial effect on the ease of use. Furthermore, the pointing-based HMI can be easily deployed in large plants; also, it can be simultaneously used by multiple operators, thus covering larger areas of interaction.

Other aspects might limit the performance and the usability of the pointing-based HMI. The localization of the operator, required at each interaction phase, and the triggering of the interaction make the interface not very reactive. As a result, manufacturing lines where packages travel too fast might be too challenging. Tracking systems or other localization technologies could alleviate this issue, but they also reduce the system portability. Furthermore, in complex CPS architectures, e.g., with multiple conveyor belts intersecting each other, the pointing might produce ambiguous results. Additionally, the human operator view might be limited; other systems, e.g., based on augmented reality [5] or using screens, could be more suited in complicated scenarios. For long or frequent interaction sessions, pointing the arm straight might be too demanding and tiring for the user, and the performance could degrade over time. Finally, the pointing-based HRI allows to communicate only one information at a time (in our case a package selection), which might be reductive for more sophisticated interactions.

2.4. Implementation details

The pointing-based HMI has been implemented [2] using the Robot Operating System (ROS2) framework [26] and tested with both simulated scenarios and simple real-world installations, with experimentation in the virtual reality domain [12]. Internally, it is composed of a set of processes (*ROS nodes*) that communicate using a publisher-subscriber pattern (*ROS topics*). In an interactive application with the considered CPS, the pointing-based HMI provides the macroscopic module depicted on the left of Fig. 3: as input, it expects notifications about *packages* (e.g., when a package is added or diverted from a conveyor belt) and *belts* (e.g., when the speed of a portion of the conveyor belt changes) which are then continuously integrated to update the positions of packages currently traveling on the conveyor belts; as output, it provides a *selection*, i.e., the list of selected packages. Note that internally the pointing-based HMI module takes also the IMU signal as input.

With respect to previous work [2, 12], here we describe the steps towards the integration of the proposed HMI in industrial control systems compliant with the IEC 61499 standard.

3. Real-time automation infrastructure

3.1. The IEC 61499 standard

IEC 61499 is a standard for modeling and programming distributed control systems for industrial automation [27]. It supports the integration of different automation tasks such as controller logic programming, system testing, and connecting field devices. The standard simplifies the programming and operation of automation systems by leveraging:

- *portability*: software tools can correctly interpret and process elements provided by other software tools;
- *configurability*: automation devices and their functions can be configured by different software tools;
- *interoperability*: devices from different vendors can operate and interact with each other.

The design of the standard is application-centric. When programming an automation system, one or more applications, defined by networks of interconnected *function blocks*, are created for the whole system and subsequently distributed to the available devices. Function blocks (see right part of Fig. 3) use two different pathways (contrary to a single pathway in ROS2) to pass data (black) or events (magenta) to other function blocks.

More specifically, the design follows a hierarchical structure: all the system devices are described within their device model; the system topology is reflected by the system model; the distribution of an application is described by the mapping model. Thus, the system applications are distributed (but maintained together) so that they can run on one or more devices. Also, applications can be developed independently of the hardware.

3.2. Implementation on an industrial demonstrator

In the context of the 1-SWARM research project [1], industrial and academic partners collaborate on orchestrating intelligent CPSs across their whole life cycle. The demonstrator [7], introduced in Section 1 and depicted in Fig. 1 and Fig. 2, showcases innovation, developed in 1-SWARM, linked to Industry 4.0 and, in particular, to IEC 61499. It has several software and hardware subsystems based on IEC 61499, among which:

- a motion controller for the gantry robot;
- controllers for the gantry robot gripper, the conveyor belt motors, and the pneumatic pistons actuating the diverters;
- the IEC 61499 runtime and the task scheduler.

This is a resupply of March 2023 as the template used in the publication of the original article contained errors. The content of the article has remained unaffected.

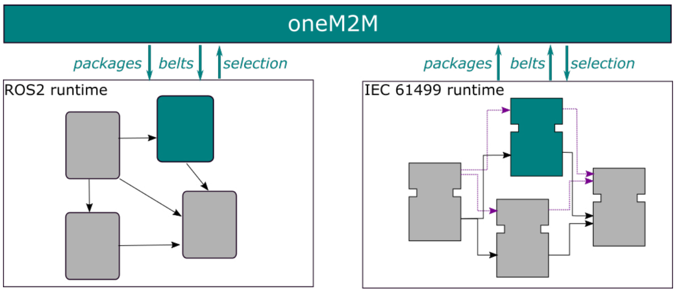

Fig. 3. Integration of the pointing-based HMI implemented in ROS2 (left) with the IEC 61499 automation runtime (right) using a bridge based on oneM2M (top). One of the ROS nodes (turquoise) acts as a oneM2M application entity, while it communicates with others ROS nodes (gray) through ROS topics (black arrows). The IEC 61499 runtime is composed by function blocks: one of them (turquoise) acts as a oneM2M application entity. The two oneM2M application entities exchange data (*packages*, *belts*, *selection*) through a oneM2M common service entity.

The distributed nature of the hardware topology is reflected in the software architecture: each major subsystem is represented by an IEC 61499 function block, either directly implementing the internal logic governing its behavior or wrapping an implementation done in other languages (see Fig. 3, right block). All these subsystems are real-time logic and are developed within the context of the IEC 61499 hierarchical application.

In this work, we use the simulated version of this demonstrator as test-bed for our pointing-based HMI.

4. Integration

4.1. The oneM2M framework

OneM2M [20] is a framework for the Internet of Things (IoT) that standardizes the interfaces of network participants like devices and services, and promotes their automatic discovery. It provides functionalities needed by IoT applications across different industry domains (e.g., data transport, security/encryption, scheduling, event notifications, remote software update) based on the REST (REpresentation State Transfer) pattern, commonly used for HTTP internet services. It supports different transport protocols, like HTTP, CoAP, MQTT, and Web Sockets, and different serialization protocols for payloads, like XML, JSON, and CBOR.

We use oneM2M to interface the IEC 61499 platform with external systems, such as the HMI described in Section 2.

4.2. Connecting the HMI with the automation runtime

As illustrated in Fig. 3, the pointing-based HMI and the IEC 61499 runtime are connected by a common oneM2M service entity. The IEC 61499 application is responsible for initializing oneM2M resources, which are then discovered by the HMI. Within each network, one node (i.e., one ROS node and one IEC 61499 function block) acts as oneM2M *application entities*, translating any I/O message between their respective networks and the oneM2M service. In particular, the application entities update the following resources on the server:

- on the IEC 61499 side, the packages transported by the conveyor belts as a list of tuples `<uid, position>`, where `uid` is the package unique identifier;
- on the IEC 61499 side, the speed of the conveyor belts as a list of tuples `<part, speed>`, where `part` identifies portions of the conveyor;
- on the HMI side, the current selection of packages as a list of `uid` identifiers.

Furthermore, the two application entities are notified when the resources get updated on the server and republish on their respective network the current selection (on the IEC 61499 side), and the current packages and speeds (on the HMI side). The two application entities use JSON for data serialization but different transport protocols to communicate with the oneM2M server: MQTT for the IEC 61499 runtime, and HTTP for the HMI ROS2 runtime.

4.3. Testing

OneM2M bridge. We tested the oneM2M bridge between the ROS2 and IEC 61499 runtimes by exchanging messages of increasing size. Table 1 reports the results using Mobius, current version 2.5.6 [14], as the oneM2M server, running on two cores of a modern consumer-grade CPU: we can exchange up to 100 messages smaller than 100 kB per second, keeping the latency below 30 ms for 95% of them. These numbers are suitable to manage hundreds of packages.

System simulation. We performed a preliminary feasibility test in simulation using CoppeliaSim [23]. The simulated scene contains the digital twin of the automation system that diverts selected packages to a specific bay, see Fig 4. It also simulates a user who points defective packages, while publishing readings from the simulated IMU on the wrist. To test the system under different levels of stress, we vary the number of packages being transported at the same time and the timing at which the automation runtime publishes their positions. In the most extreme case, we tested the system with as many packages as possible on the belts at same time (i.e., 30, with a 10% probability of being defected), and with positions updated continuously

Table 1. OneM2M bridge performance

Payload (Bytes)	Maximal message rate (Hz)	Latency (ms) [5th, 95th] percentiles
1	173	[10, 24]
10	173	[10, 24]
100	173	[10, 24]
1000	173	[10, 24]
10000	165	[10, 24]
100000	106	[14, 29]
1000000	20	[52, 79]

This is a resupply of March 2023 as the template used in the publication of the original article contained errors. The content of the article has remained unaffected.

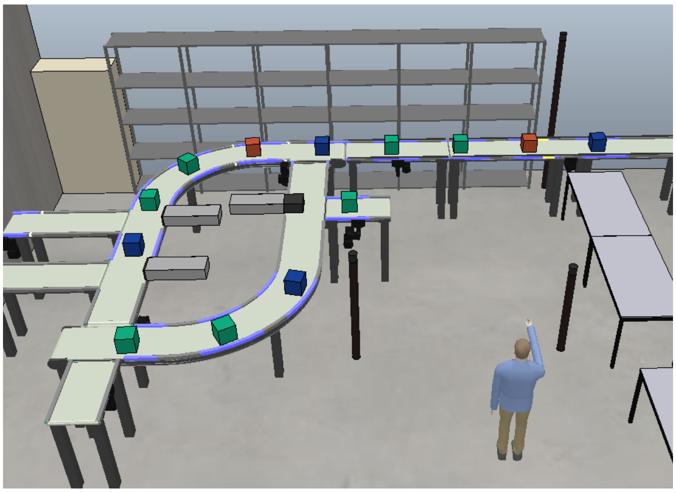

Fig. 4. The simulation environment used to test the feasibility of the system.

at 60 Hz. All the data published by the simulation is fed into the pipelines presented in Section 4.2. In particular, packages positions (up to 60 messages per second, 670 bytes per message), belt speeds (only one message at start, 64 bytes for 8 belts), and packages selection (up to 0.2 messages per second, 50 bytes per message) are exchanged over the oneM2M service in real-time. As the requirements are well within the capabilities of the bridge, we recorded no loss of messages and latency was negligible. The simulated user managed to select all the defective packages and the system was able to divert them.

5. Conclusion and future work

We integrated a pointing-based human-machine interface in the framework of an industrial cyber-physical system compliant with the IEC 61499 standard. We analyzed both advantages and drawbacks of using such an interface to establish an interaction with a system of conveyor belts. In this particular context, a human operator can notice and indicate to the system a defective package that can be properly discarded from the process line. We discussed the novel integration required to use the proposed interface with an IEC 61499 automation runtime, which allows the actual deployment within an industrial automation framework. The implementation is built on the oneM2M framework serving as a bridge between the pointing-based interface (implemented in ROS2) and the IEC 61499 runtime. The tests on the oneM2M bridge showed that it is possible to execute the considered interaction, i.e., to exchange the required messages without observing any significant loss of information or latency. Feasibility tests in simulation showed that the interaction requirements are respected by our integration, paving the road towards the deployment on real systems.

Future work aims at testing the pointing-based human-machine interface within the real Industry 4.0 demonstrator. To this end, we plan to implement our framework in standalone computational units, so that the deployment in the real demonstrator will be favored by an higher level of portability. So far, we have tested the HMI and the bridge to connect it to an IEC 61499 compliant framework in simulation. As common for real industrial deployments, and in particular for control systems programmed using IEC 61499, the integration of external parts (like the pointing-based HMI) requires careful engineering to avoid computation and communication bottlenecks. For example, the number of events triggered in the IEC 61499 application will need to be managed in the real scenario. To overcome such implementation gaps, a redesign of the oneM2M interface could be necessary in the final product solution.

Acknowledgements

This work is supported by the European Commission through the Horizon 2020 project 1-SWARM, grant ID 871743.

References

[1] 1-SWARM, Accessed: 2021. https://www.1-swarm.eu/.
[2] Abbate, G., Giusti, A., Paolillo, A., Gromov, B., Gambardella, L., Rizzoli, A.E., Guzzi, J., 2022. PointIt: A ROS toolkit for interacting with co-located robots using pointing gestures, in: ACM/IEEE International Conference on Human-Robot Interaction, pp. 608–612.
[3] Butterworth, G., 2003. Pointing is the royal road to language for babies. Pointing: Where Language, Culture, and Cognition Meet .
[4] Cosgun, A., Trevor, A.J.B., Christensen, H.I., 2015. Did you Mean this Object?: Detecting Ambiguity in Pointing Gesture Targets, in: HRI'15 Towards a Framework for Joint Action Workshop.
[5] Dimitropoulos, N., Togias, T., Zacharaki, N., Michalos, G., Makris, S., 2021. Seamless human–robot collaborative assembly using artificial intelligence and wearable devices. Applied Sciences 11.
[6] Droeschel, D., Stückler, J., Behnke, S., 2011. Learning to interpret pointing gestures with a time-of-flight camera, in: International Conference on Human-robot Interaction, pp. 481–488.
[7] Fabbrica diffusa, Accessed: 2021. https://www.comonext.it/laboratori/.
[8] Gleeson, B., MacLean, K., Haddadi, A., Croft, E., Alcazar, J., 2013. Gestures for industry intuitive human-robot communication from human observation, in: ACM/IEEE International Conference on Human-Robot Interaction, pp. 349–356.
[9] Gromov, B., Abbate, G., Gambardella, L., Giusti, A., 2019a. Proximity human-robot interaction using pointing gestures and a wrist-mounted IMU, in: IEEE International Conference on Robotics and Automation, pp. 8084–8091.
[10] Gromov, B., Gambardella, L., Giusti, A., 2019b. Guiding quadrotor landing with pointing gestures, in: International Workshop on Human Friendly Robotics, Springer International Publishing.
[11] Gromov, B., Gambardella, L.M., Di Caro, G.A., 2016. Wearable multi-modal interface for human multi-robot interaction, in: IEEE International Symposium on Safety, Security, and Rescue Robotics, pp. 240–245.
[12] Guzzi, J., Abbate, G., Paolillo, A., Giusti, A., 2022. Interacting with a conveyor belt in virtual reality using pointing gestures, in: ACM/IEEE International Conference on Human-Robot Interaction, pp. 1194—1195.
[13] Imtiaz, J., Koch, N., Flatt, H., Jasperneite, J., Voit, M., van de Camp, F., 2014. A flexible context-aware assistance system for industrial applications using camera based localization, in: IEEE Emerging Technology and Factory Automation, pp. 1–4.
[14] IoTKETI, Accessed: 2021. https://github.com/IoTKETI/Mobius.
[15] Janssen, C.P., Donker, S.F., Brumby, D.P., Kun, A.L., 2019. History and future of human-automation interaction. International Journal of Human-Computer Studies 131, 99–107.
[16] Kondo, K., Mizuno, G., Nakamura, Y., 2016. Analysis of human pointing behavior in vision-based pointing interface system - difference of two typical pointing styles -. IFAC-PapersOnLine 49, 367–372.

[17] Makris, S., Tsarouchi, P., Matthaiakis, A.S., Athanasatos, A., Chatzigeorgiou, X., Stefos, M., Giavridis, K., Aivaliotis, S., 2017. Dual arm robot in cooperation with humans for flexible assembly. CIRP Annals 66, 13–16.

[18] Maurtua, I., Ibarguren, A., Kildal, J., Susperregi, L., Sierra, B., 2017. Human–robot collaboration in industrial applications: Safety, interaction and trust. International Journal of Advanced Robotic Systems 14, 1729881417716010.

[19] Nickel, K., Stiefelhagen, R., 2003. Pointing Gesture Recognition based on 3D-Tracking of Face , Hands and Head Orientation Categories and Subject Descriptors, in: International Conference on Multimodal Interfaces, pp. 140–146.

[20] oneM2m, Accessed: 2021. https://www.onem2m.org.

[21] Plaumann, K., Weing, M., Winkler, C., Müller, M., Rukzio, E., 2017. Towards accurate cursorless pointing: the effects of ocular dominance and handedness. Personal and Ubiquitous Computing 22, 633–646.

[22] Profanter, S., Perzylo, A., Somani, N., Rickert, M., Knoll, A., 2015. Analysis and semantic modeling of modality preferences in industrial human-robot interaction, in: IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 1812–1818.

[23] Rohmer, E., Singh, S.P.N., Freese, M., 2013. CoppeliaSim (formerly V-REP): a versatile and scalable robot simulation framework, in: IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 1321–1326.

[24] Savazzi, S., Rampa, V., Vicentini, F., Giussani, M., 2016. Device-free human sensing and localization in collaborative human–robot workspaces: A case study. IEEE Sensors Journal 16, 1253–1264.

[25] Sheikholeslami, S., Moon, A., Croft, E.A., 2017. Cooperative gestures for industry: Exploring the efficacy of robot hand configurations in expression of instructional gestures for human–robot interaction. The International Journal of Robotics Research 36, 699–720.

[26] Thomas, D., Woodall, W., Fernandez, E., 2014. Next-generation ROS: Building on DDS, in: ROSCon Chicago 2014, Open Robotics.

[27] Vyatkin, V., 2009. The IEC 61499 standard and its semantics. IEEE Industrial Electronics Magazine 3, 40–48.

[28] Wolf, M.T., Assad, C., Vernacchia, M.T., Fromm, J., Jethani, H.L., 2013. Gesture-based robot control with variable autonomy from the JPL BioSleeve, in: IEEE International Conference on Robotics and Automation, pp. 1160–1165.

This is a resupply of March 2023 as the template used in the publication of the original article contained errors. The content of the article has remained unaffected.