Cloud-Based Digital Twin for Industrial Robotics

Timon Hoebert, Wilfried Lepuschitz, Erhard List, and Munir Merdan(✉)

Practical Robotics Institute Austria, Wexstrasse 19-23, 1200 Vienna, Austria
{hoebert,lepuschitz,list,merdan}@pria.at

Abstract. Production systems are becoming more flexible and agile to realize the need for more individualized products. Robotics technology can accomplish these demands, but programming and re-configuration of robots are associated with high costs, especially for small- and medium-sized enterprises. The use of digital twins can significantly reduce these costs by providing monitoring and simulation capabilities for the robot and its environment using real-time data. The integration with an ontology as a knowledge base to describe the robot and its 3d-environment enables an automatic configuration of the digital twin and the particular robot. In this paper, this concept is coupled with cloud-computing to enable an effortless integration as service in existing cloud architectures and easy access using the common web-technology-stack for the end-users. A novel architecture is presented and implemented to incorporate the real system with its digital twin, the ontology and a planner to infer the actual operations from the knowledge base. Finally, the implementation is applied to the industrial manufacturing domain to assemble different THT-Devices on a PCB to evaluate the concept.

Keywords: Digital twin · Industrial robotic · Autonomous system · Ontology

1 Introduction

In order to handle the ever-increasing number of individualized products with short delivery times, the production system has to become much more flexible and agile. Particularly in the field of assembly systems the growing variety of products adds new complexities to the planning process and increases the costs, because (re-)planning efforts tend to grow exponentially to the number of variants [16]. Robotics technology, which is able to prove high efficiency, precision and repeatability, seems like a viable solution to cope with these challenges. However, robot systems still often do not meet the demands of small- and medium-sized enterprises (SMEs); with a high number of small lot sizes and product variants, frequent re-configuration and re-programming, but also little or no in-house robot expertise. Also, the setup and operation of a robot in an SME environment are not that easy, since it is typically less structured and involves more uncertainties than large-scale or mass production industries [5].

In order to be able to dynamically adapt to new products and unpredicted production changes, robotic systems need to work autonomously. Autonomous systems, in

V. Mařík et al. (Eds.): HoloMAS 2019, LNAI 11710, pp. 105–116, 2019.
https://doi.org/10.1007/978-3-030-27878-6_9

this context, means that robots systems are able to perform high-level task specifications without explicitly being programmed [3]. In order to reach specific goals, such systems employ automated reasoning and choose their actions knowing their own capabilities and their current state. To perform an aimed action successfully, the autonomous systems have to have a realistic representation of their own capabilities and environment as well as of the current state of the production process. Nonetheless, the complexity of the issues involved in systems, a large number of subsystems and factors that directly or indirectly affect the productivity, efficiency, and synchronization of a production or assembly process, requires searching and verification of the possible solutions in the processes of designing and modeling these systems [17]. Besides, to guarantee safety, new control methods and architectures have to be fully tested and proven before practical usage. According to the International Federation of Robotics (IFR), the challenges of adopting new technologies as well as the lack in advanced safety systems for the supervision of human-robot workspace sharing and cooperation represent some of the main reasons that companies do not use robots yet [13].

However, performing tests in a running production process can generate additional costs and put the personal and production equipment in danger. Simulation can be a very effective way of testing different control architectures and improving the quality of the designed solutions and can enable to some extent the reliability and effectiveness of the entire system. The application of simulation solutions for designing, testing, and verification of control programs and algorithms in the industry but also in service robotics is well established and there are a lot of solutions currently available (e.g. RobotStudio, KUKA SimPro, Robotics Suite, Gazebo, etc.). An overview of current robot offline programming tools can be found in [4, 6, 8]. Nevertheless, most of these tools do not support semi-automatic program generation on the abstraction level of process steps [10]. Furthermore, most of them are focused on processes where they are used to specify basic robot movements and are mostly not suitable for all use cases within specific domains. Besides, there is usually a strict separation between the virtual runtime environment provided by robotic simulation frameworks, and the motion and grasp planning components, imposing severe limitations on the interaction possibilities between planning and runtime components when it comes to robotic simulations [1]. Data models used by engineering and simulation tools are mostly not compatible or insufficient to fulfill all parameters of others, restricting the interoperability in an increasingly multi-vendor tool environment [20]. In this context, software that aims to mimic robotic work cells needs to represent exactly all components and display them accurately while facilitating poses, paths and trajectories testing [18]. Besides, to evaluate the impact of external and internal changes and to react in a timely manner to critical influences on production management, changes within a physical environment have also to be mirrored within the virtual model [20].

From a simulation point of view, the Digital Twin approach is seen as the next big step in modeling, simulation and optimization technology [15]. The idea of the digital twin is to create a digital counterpart to mimic the characteristics and behaviors of its corresponding twin, realizing in an effective way the communication and interaction between the physical and digital world [23]. It is essential to represent the characteristics, behavior, and relations of the system components like products, resources, sensors as well as the kinematics of a robot properly within the digital twin [9]. Digital twin technology establishes feedback loops between real-world objects and their virtual substitutes. In this article, our goal is, on the one hand, to reduce the required manual steps for the modeling of digital twin environments with sensors, actuators, and devices. The digital twin generation should also support the communication and interaction between the physical and digital environment. On the other hand, the established digital twin should enable easier tests and development of autonomous robotic systems including their automated configuration and programming in realistic environments under specific conditions and constraints. The entire framework provides also monitoring means of involved devices represented through their digital twins. A further aim was to present data through the digital twin framework and to visualize the information in a user-friendly graphical interface as well as to perform proof of concept tests on laboratory equipment.

This paper is structured as follows. Section 2 discusses the framework architecture. Section 3 shows the digital twin integration and configuration. In Sect. 4, we present in detail the implemented framework. Section 5 describes the digital twin, when applied in specific use case. Finally, the paper is concluded in the sixth section.

2 Framework Architecture

The main goal of this research is to create a digital twin of an autonomous robotic system. The twin should be automatically created from the semantical description of the system, and furthermore, it should able to supervise the real system's current actions based on the data that the digital twin receives from it. The digital system analyzes the current situation in the real system, tests some alternative scenarios and gives relevant feedback to the real system when these scenarios result in improved action. The digital twin collects all relevant knowledge about the product, production equipment, and the production process and uses this knowledge to suggest suitable actions to the real system. It enables the correct and efficient modeling of the environment but takes a vital role in the prediction of the possible autonomous system behaviors. In this setting, the digital twin is a vital enabler of autonomy and an important factor for achieving a new level of flexibility in automation systems. There are basically four major components contained in the digital twin framework, including the semantic description of the robot system environment, a decision-making mechanism (planner), the digital environment model and the real system (Fig. 1).

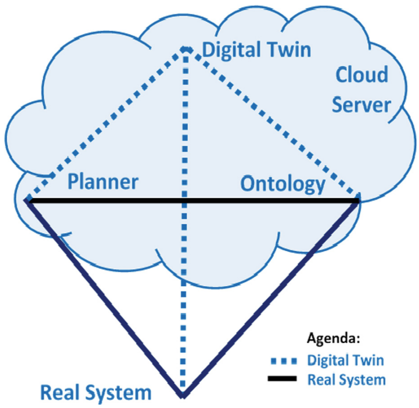

Fig. 1. Digital twin concept

2.1 Ontology

In order to realize the digital twin, a vital part is the development of a digital model of the environment, which must be as precise and detailed as its real twin in order to execute accurate simulations and evaluations [9]. Lately, the attention has been also oriented on the application of ontologies to meet the goals of modeling, meta-modeling, and interoperability between the digital tools in the Virtual Factory context in order to guarantee an accurate digital link. The key benefits of using an ontology-based approach include (i) the exploitation of semantic web technologies in terms of interoperability, data distribution, extensibility of the data model, querying, and reasoning; and (ii) the re-use of general purpose software implementations for data storage, consistency checking and knowledge inference [19]. In this context, it is crucial to accurately model and represent the relevant elements of the system such as products, robots, actuators, users, tools, sensors, etc., considering their properties, behavior, and relations to each other as well as corresponding locations. For instance, the kinematics of a robot must be properly represented as well as sensor and actuator specifications (e.g., their accuracy and frequency). Based on the exact virtual representation of the physical environment, the model can provide information about the concrete position of the specific element. The ontology also specifies technical details, which are required for automated device registration and enabling access to the devices.

2.2 Planner

The reasoning functions of the robotic system are realized in the planner, the second important part is the digital framework. The responsibility of the planner is to link the semantic description of the framework and the skills of the acting resources in the system. In this context, the planner uses an ontology-based product model to reason about relations between workpieces as well as to connect abstract processes and needed equipment to generate concrete operations. Based on the initial state of the system (its relevant elements considering their positions and orientations) and goal that the system aims to reach, the planner generates the list of high-level actions, which are needed to reach that goal. The preconditions dictate items that must be initially valid for the action to be started. During ongoing execution, the continuous evaluation of the performance of the actions ensures they are executed as expected. When the execution has finished, a final postcondition check determines if the effects of the undertaken actions are as predicted.

2.3 Digital Twin

Autonomous systems are aware of their own capabilities (skills) as well as about their current state. Nevertheless, in order to be able to undertake appropriate actions, they need to have a meaningful representation of the surrounding environment. Moreover, considering the fact that they act autonomously and that their actions are sometimes not predictable, it is of crucial importance to test or supervise their behaviors. Their control programs and related behaviors can be developed and tested using real control systems in combination with a digital representation of the later real application environment, including all relevant models of resources, tools, sensors, etc. The test planning and testing environment can be used to validate robotic tasks and actions. Creating a digital production environment including all relevant elements gives a possibility to compare the actual path of a workpiece with an optimal one. It can be also used to validate robot actions for specific initial states and adjust particular parameters of the control strategy until one or more optimal cases are resolved. Also, the integration of collision detection in the digital twin enables a virtual sensing domain which is not possible with current industrial robots, where a collision is fatal during application.

2.4 Cloud

Cloud technologies provide a shared environment of manufacturing capability, computing power, knowledge, and resource, offering an environment to connect the computing and service resources in the digital world to the machines and robots in the physical world [22]. The integration of digital twins into cloud manufacturing systems imposes strong demands such as reducing overhead and saving resources [7]. We have created a cloud-based digital twin framework that exploits modern Web technologies and offers modeling, simulation, and supervision environment to the end-users' needs. The digital twin layout is generated automatically online and can be accessed via a web browser, even via a mobile device. In our approach, we have every physical component in the system represented by its digital twin hosted in the cloud. Every time the real world state changes, an update of the current component status is sent to its digital representation in the cloud. Additionally, the use of cloud-based technology enables easy use of the digital twin as service without a user interface, for example, directly connected to the path planning component for collision detection of a proposed path.

3 Framework Integration

The components, as presented above, are integrated into the digital twin framework as visualized in Fig. 2. In our previous work, we described the function and implementation of the ontology-driven industrial robotics system as well as its automated configuration [12]. As a basis for this system and presented digital twin framework, the Rosetta ontology [24], which is concentrating on robotic devices and particular skills. We also integrated the BREP ontology [21], which represents geometric entities in a semantic way. Within the digital twin framework, we particularly focused on representations of industrial robots and their features: manufacturer, type, axes, segments, weight, tools, etc. The ontology describes also concepts which are required for an automated configuration and access to the devices.

The Planner automatically determines a set of suitable actions that lead to the final goal state. Nevertheless, the selected plan is strongly related to the current state of the entire robotics system. The action schemas of a planning domain include parameters, preconditions, and effects. The preconditions dictate items that must be initially valid for the action to be legal. The effect equation dictates the changes in the world that will occur due to the execution of the action.

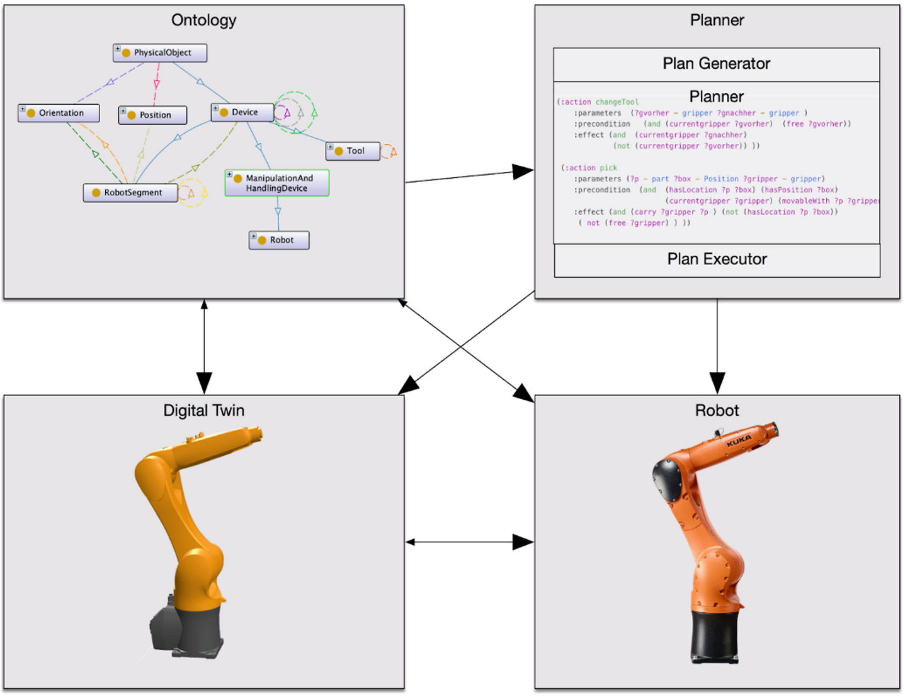

Fig. 2. The implemented framework of the digital twin

The digital twin is automatically configured based on the information of the ontology and the real system. The ontology provides spatial information, which is interconnected with other semantic data-structures, for example, skills. Especially the spatial information about the geometry and positioning of the workpieces, the robot itself and its environment are relevant to automatically adapt the digital twin. Therefore the abstract representation of the joints of the robots and its actual triangulated geometry are transferred to the digital twin for the initial configuration. The segments of the robot are represented using their length and the link to the next connected segments. The orientation and position of the next connection point is defined in a local coordinate space, based on the current segment. This process enables the fast configuration of new robots based on similar or slightly changed geometry. In a similar way, the tools are represented. Each tool has an operation reference point, for example, the center of the vacuum gripper. Similarly, the environment of the robot is also mapped into the digital-twin, to determine the working area of the robot. Using the collision detection of the digital twin, changes in this mapped environment can lead to adapted robot paths to avoid collisions with the surroundings. In the ontology, the workpiece and its sub-parts are defined by the geometry and the handling information. The handling-information contains information about the local picking position of the part and the possible tool which can carry out the operation. This information adapts the used tool of the robot and path-planning.

As stated before, the digital twin also configures based on spatial information of the robot and its sensor data. In this context, the current angles of the axes and the sensor values of the tool are transmitted to the digital twin. This information helps to monitor the robot and to gain insight into the robot's autonomous movements.

4 Implementation

The digital twin framework consists of 5 components, as shown in Fig. 3. The main component is the cloud-based Service Hoster, which connects the different components transparently.

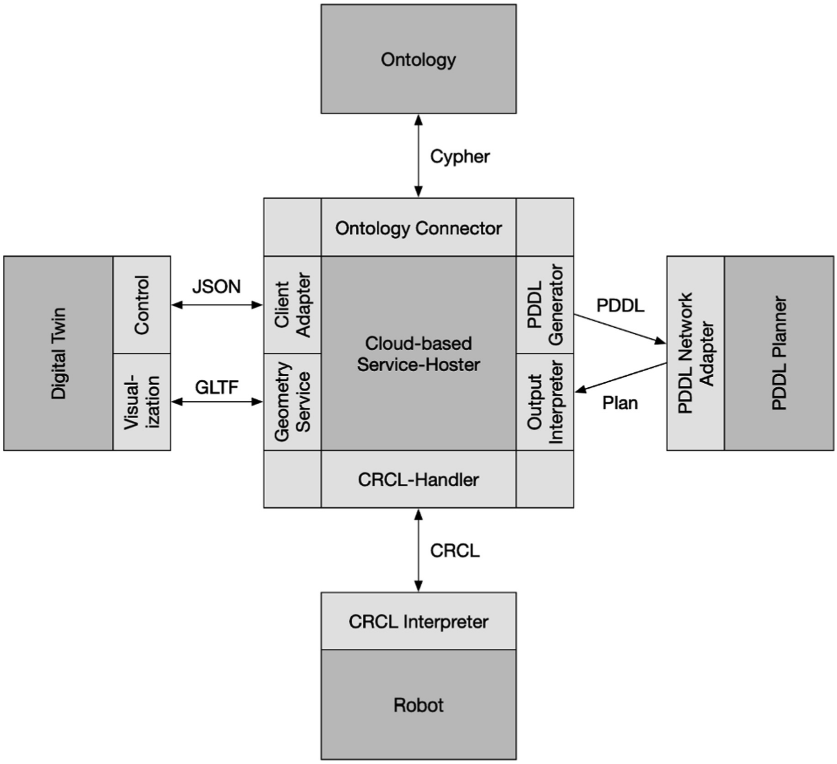

Fig. 3. The implemented framework of the digital twin

It is implemented using Node.JS for maximal interoperability with the other components because of existing included frameworks. It has a dedicated Ontology-Connector service to interact with the ontology. The ontology is implemented using the open-source named property graph database Neo4j to handle the performance demands of geometry data processing and storing. The Sci-Graph project is used to migrate

OWL-Ontologies into the database[1]. This implementation choice led to the use of the Cypher query language, which is used to query the ontology data.

As mentioned before, the ontology stores all information about the workpieces, their geometry, and the production process. The geometry is associated with production information, for example, the possible handling tools. Also, the capabilities and geometry of the robot are stored and linked here via relations. Additionally, the saved information of the composition of the final workpiece is relevant for planning. Hereby, a planner is used to generate a possible production plan based on a minimal cost function, e.g. minimal production time. The PDDL-standard [11] is used for this purpose. A generic network adapter is implemented to connect the planner with the service. The PDDL problem and domain files are generated based on the information of the ontology. The generated files are sent to the planner, which solves the problem instance with a concrete plan. This plan is sent back to the server to interpret this output plan with the spatial information of the ontology. Using this data, concrete commands for the robot can be formulated. For this purpose, the CRCL-standard [14] is used to communicate with the robot and the digital twin. To support this web-based application area, the XML-based syntax is replaced by JSON. This JSON-format is also used since it has a lower message-size. A CRCL-interpreter on the robot receives these messages for its execution queue. The execution status information ("received", "queued", "started", "finished") is sent back to the server for synchronization with the digital twin. The current axis positions are also transmitted using CRCL-status messages.

The digital twin is implemented as HTML5-application to support different devices and operating systems. Figure 4 shows the user interface of this web-application. It visualizes the production plan and the digital twin. The client is connected to the server via HTTP and JSON for asynchronous communication. The digital twin visualizes the workpieces and the robot as 3D-realtime rendering. Here, BabylonJS is used as rendering engine which uses WebGL to support a browser visualization without any installation. The geometries of the workpieces and the robot are fetched from the server in the GLTF standard which is a JSON based format optimized for WebGL. This triangulated geometry is also used for collision detection. In this digital twin, the user can upload new workpieces which change the planned production steps. Additionally, the workpieces in the digital twin can be moved in space to change the spatial information in the ontology, to configure the robot's path planning.

5 Application

The assembly process is one of the most important production phases since the quality, life, performance, reliability, and maintainability of a product mostly depend on assembly results [25]. Assembly operations are considered being among the most intensive and costly processes in production lines, due to the variability of the assembled parts and the complexity of the executed tasks [2]. The assembly process can become especially extensive when coordination of several different actions need to

[1] https://github.com/SciGraph/SciGraph.

be integrated and performed by an autonomous robotics system. Especially considering that this system also requires an exact representation of the current state of the production process and of its actions.

These difficulties make the assembly domain relevant to the use of the digital twin concept. As mentioned before, the digital twin can collect all this vitally important information on the production process and make it available to the autonomous system. This information can be simulated with the digital twin model to anticipate the consequences of actions by the autonomous system in a given situation and possibly modify the course of action [15]. Due to these reasons, we apply the digital twin for the test case that focuses on the assembling of different THT-Devices (relay, capacitor, screw terminal, and potentiometer) on a PCB. In this scenario, THT-Devices for an assembly line are placed in boxes at the workstation. Each box contains different parts, which have an exact position in the box. In the automated pilot case, the KUKA robot KR 6 R900 sixx performs a series of pick-and-place operations in assembling the PCB board, as presented in Fig. 4.

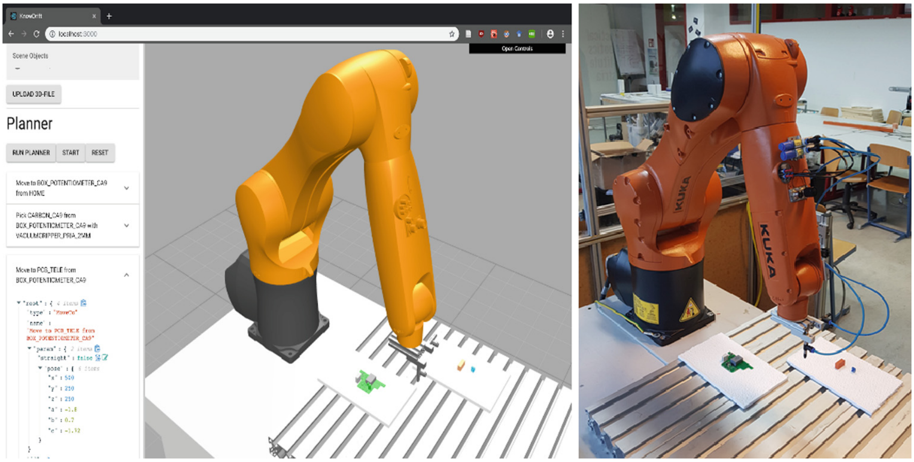

Fig. 4. Digital twin applied in the assembly domain

In the digital twin, the end product is specified by the user through the GUI as the final goal, being defined as the assembled PCB board with the geometric information containing the characteristics (weight, position, and orientation) of the individual parts). These data are forwarded in combination with the semantic description to the planner. Once the user starts the process, the digital twin provides the information about the current state of the system and the planner starts with the specification of the single assembly activities autonomously. Each activity consists of a series of operation that can be executed with different tools or by one or more resources. The PDDL planner extracts the corresponding production/assembly operations and links the particular action with the required tool, which has to be used to perform the specific operation (e.g. handling, assembly, fixing). These tools and relevant skills, as well as resources, are represented in the ontology. Based on these specific operations the robot can reason about the required tool (e.g. a gripper for handling). This activate sequence is then mapped accordingly into a robot program, which is then executed. The activities are then reviewed by the user who could then approve them or change specific parameters in order to improve the efficiency of the operations.

6 Conclusion

The application of the digital twin technology to support the development of an autonomous robotics system is proven as a promising approach. In this paper, we use the digital twin framework for modeling, simulation, supervision, and optimization of the autonomous robotics system. The developed framework is composed of four major components: semantic description of the robot system (ontology), decision-making mechanism (planner), digital environment model and the real system. The use of an ontology provides a unified structure for the representation of the system environment. The reasoning functions of the robotic system are realized in the planner and connected with the ontology-based world model. The function of the entire digital twin framework is based on the continuous interaction and iterative improvement among those four elements.

The digital twin framework is automatically configured based on the information from the ontology and enables reduction of manual steps during the modeling process. The developed digital twin facilitates easier tests and development of autonomous robotic systems including their programming and monitoring. The proposed visualization technique and use of cloud-based technology also enable easier supervision and better understanding of the autonomous approach. The feasibility and effectiveness of the proposed framework are validated on the real use case that focuses on the assembling of different THT-Devices.

Future work will be continued with more complex assembly use cases as well as usage of a depth-camera to detect and track dynamic objects in the system. Moreover, we aim also to extend the digital twin approach to other types of manufacturing machines (cutting and grinding machines, etc.).

Acknowledgment. The authors acknowledge the financial support from the “Production of the Future” program of the Austrian Ministry for Transport, Innovation and Technology under contract FFG 858707.

References

1. Aichele, F., Schenke, B., Eckstein, B., Groz, A.: A framework for robot control software development and debugging using a real-time capable physics simulation. In: Proceedings of ISR 2016: 47st International Symposium on Robotics, Munich, Germany (2016)
2. Argyrou, A., Giannoulis, C., Papakostas, N., Chryssolouris, G.: A uniform data model for representing symbiotic assembly stations. Proc. CIRP **44**, 85–90 (2016). 6th CIRP Conference on Assembly Technologies and Systems CATS, 2016
3. Beckey, G.A.: Autonomous Robots. MIT Press, Cambridge (2005)

4. Gan, Y., Dai, X., Li, D.: Off-line programming techniques for multirobot cooperation system. Int. J. Adv. Robot. Syst. **10**, 282 (2013)
5. Haage, M., Profanter, S., Kessler, I., Perzylo, A., Somani, N., Sörnmo, O.: On cognitive robot woodworking in SMErobotics. In: Proceedings of ISR 2016: 47st International Symposium on Robotics, pp. 1–7 (2016)
6. Harris, A., Conrad, J.: Survey of popular robotics simulators, frameworks, and toolkits. In: Proceedings of IEEE Southeastcon (2011)
7. Hu, L., et al.: Modeling of cloud-based digital twins for smart manufacturing with MT connect. Proc. Manuf. **26**, 1193–1203 (2018)
8. Kumar, K., Reel, P.: Analysis of contemporary robotics simulators. In:2011 International Conference on Emerging Trends in Electrical and Computer Technology (ICETECT) (2011)
9. Kuts, V., Modoni, G.E., Terkaj, W., Tähemaa, T., Sacco, M., Otto, T.: Exploiting factory telemetry to support virtual reality simulation in robotics cell. In: De Paolis, L., Bourdot, P., Mongelli, A. (eds.) Augmented Reality, Virtual Reality, and Computer Graphics, AVR 2017. LNCS, vol. 10324, pp. 212–222. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-60922-5_16
10. Macho, M., Naegele, L., Hoffmann, A., Angerer, A., Reif, W.: A flexible architecture for automatically generating robot applications based on expert knowledge. In: Proceedings of ISR 2016: 47st International Symposium on Robotics, pp. 1–8 (2016)
11. Mcdermott, D., et al.: PDDL - the planning domain denition language (Technical report). CVC TR-98- 003/DCS TR-1165, Yale Center for Computational Vision and Control (1998)
12. Merdan, M., Hoebert, T., List, E., Lepuschitz, W.: Knowledge-based cyber-physical systems for assembly automation. Prod. Manuf. Res. **7**, 223–254 (2019)
13. Michalos, G., Makris, S., Spiliotopoulos, J., Misios, I., Tsarouchi, P., Chryssolouris, G.: ROBO-PARTNER: seamless human-robot cooperation for intelligent, flexible and safe operations in the assembly factories of the future. In: 5th CIPR Conference on Assembly Technologies and Systems, (CATS 2014), Dresden, Germany, November 2014
14. Proctor, F., Balakirsky, S., Kootbally, Z., Kramer, T., Schleno, C., Shackleford, W.P.: The canonical robot command language (CRCL). Ind. Robot: Int. J. **43**, 495–502 (2016)
15. Rosen, R., von Wichert, G., Lo, G., Bettenhausen, K.D.: About the importance of autonomy and digital twins for the future of manufacturing. In: 15th IFAC Symposium on Information Control Problems in Manufacturing (INCOM), vol. 48, no. 3, pp. 567–572 (2015)
16. Schlick, C.M., Faber, M., Kuz, S., Bützler, J.: A symbolic approach to self-optimisation in production system analysis and control. In: Brecher, C. (ed.) Advances in Production Technology. LNCS, pp. 147–160. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-12304-2_11
17. Sekala, A., Gwiazda, A., Kost, G., Banaś, W.: Modelling and simulation of a robotic work cell. In: IOP Conference Series: Materials Science and Engineering, vol. 227, p. 012116 (2017)
18. Tavares, P., Silva, A., Costa, P., Veiga, G., Moreira, P.: Flexible work cell simulator using digital twin methodology for highly complex systems in industry 4.0. In: Iberian Robotics conference, pp. 541–552 (2017)
19. Terkaj, W., Tolio, T., Urgo, M.: A virtual factory approach for in situ simulation to support production and maintenance planning. CIRP Ann.- Manuf. Technol. **64**(1), 451–454 (2015)
20. Um, J., Weyer, S., Quint, F.: Plug-and-simulate within modular assembly line enabled by digital twins and the use of AutomationML. IFAC-PapersOnLine **50**(1), 15904–15909 (2017)
21. Perzylo, A., Somani, N., Rickert, M., Knoll, A.: An ontology for CAD data and geometric constraints as a link between product models and semantic robot task descriptions. In: International Conference on Intelligent Robots and Systems (IROS), pp. 4197–4203 (2015)

22. Wang, X.V., Wang, L., Mohammed, A., Givehchi, M.: Ubiquitous manufacturing system based on cloud: a robotics application. Robot. Comput. Integr. Manuf. **45**, 116–125 (2017)
23. Yan, K., Xu, W., Yao, B., Zhou, Z., Pham, D.: Digital twin-based energy modeling of industrial robots. In: the 18th Asia Simulation Conference (AsiaSim 2018), Kyoto, Japan, 27–29 October 2018
24. Patel, R., Hedelind, M., Lozan-Villegas, P.: Enabling robots in small-part assembly lines: the "rosetta approach"—an industrial perspective. In: The 7th German Conference on Robotics, Proceedings of ROBOTIK, pp. 1–5 (2012)
25. Zhuang, C., Liu, J., Xiong, H.: Digital twin-based smart production management and control framework for the complex product assembly shop-floor. Int. J. Adv. Manuf. Technol. **96**, 1149–1163 (2018)