When Softbots Meet Digital Twins: Towards Supporting the Cognitive Operator 4.0

Ricardo J. Rabelo[1(✉)], David Romero[2], Saulo P. Zambiasi[3], and Luciano C. Magalhães[1]

[1] UFSC - Federal University of Santa Catarina, Florianópolis, Brazil
ricardo.rabelo@ufsc.br
[2] Tecnológico de Monterrey, Monterrey, Mexico
[3] UNISUL - University of Southern Santa Catarina, Florianopolis, Brazil

Abstract. Digital Twins (DT) represent one of the most powerful technologies related to Industry 4.0 implementation. Combined with the advances in Computer Science and Production Management, DTs have become an emerging trend in many applications. Despite the DT potentials, Industry 4.0 is pushing companies to adopt digital solutions that provide higher human satisfaction to increase operational excellence. One approach to handle this involves how workers should interact with systems and smart machines. Softbots is a prominent direction to underpin this goal. This paper presents ongoing results of a research that exploits how softbots can be combined with DTs to create a more "symbiotic" human-machine/computer interaction within a smarter decision-making environment.

Keywords: Industry 4.0 · Softbots · Digital Twin · Cognitive Operator 4.0

1 Introduction

Industry 4.0 paradigm involves the use and combination of several so-called "enabling technologies", being the *Digital Twin (DT)* one of them [1]. A *DT* can be defined as an intelligent and evolving digital model, following the lifecycle of its physical twin to monitor, control, simulate, and optimize its processes, at the same time that continuously predicts its future statuses for further decision-making upon that [2].

One of the most adopted strategies pursued in the *Industry 4.0 digital transformation path* is the development of "data-driven environments", where decision-makers are provided with access to integrated and more accurate information related to the diverse organisational areas, including the shop floor. Their ultimate goal is to support more agile, comprehensive, and confident decision-making, nowadays usually supported by management and real-time production dashboards [3].

Despite the potentials brought up by such data-driven environments (as smart shop floors), many decision-makers are not prepared at all to correctly understand the plethora of technical terminologies related to production management, to make wider analyses and correlations between different performance indicators, to go through data to glean insights for decision-making, among other limitations [4, 5]. Besides that, data-driven environments can bring more complexity to decision-makers if not

properly designed to avoid "information overload" [6]. The practice has been showing that decision-makers have been increasingly exposed to massive amounts of information about their companies' processes, systems (including different DTs), and equipment, where lots of checking (sometimes repetitive), analyses, supervision, reporting actions as well as critical decision-making need to be more and more often and rapidly performed. All this leaves managers' capabilities to a cognition and stress limit, jeopardizing the quality of their decision-making [5, 6].

In a *Smart Shop floor context,* although a DT can, for instance, show production data via *digital dashboards* and monitor in real-time the execution of the *Cyber-Physical Systems (CPSs)* it represents, there are some relevant issues to consider once the DT is implemented [7–11]:

- The data to be shown in a digital dashboard is typically predefined (and hence limited), including the way data will interact with its users.
- The set of data to be shown in a digital dashboard is designed for a generic user.
- The use of a predefined and generic set of data prevents different users from having access to information outside the DT data model's border in the case of ad-hoc questions or of the need for very particular situations. This means making users going through other information systems' GUIs (e.g., ERP or MEs) to pick the required information and to 'mentally' reason about it outside of the DT's environment. This can also be as difficult due to home office, as many corporate information systems are still only available as local and non-web-based systems.
- The human limitation to follow (with the eyes) many 'things' simultaneously in a DT visualization or CPS execution.
- The many 'hidden information' in a DT visualization because of poor GUI design.
- The current limitations of data and business analytics solutions, which are not yet able to provide much intelligence and effective decision-support, leaving to decision-makers the more sensible and business-context analyses out of the plenty of information and available dashboards.
- Most of the development tools in the market for DTs (especially those provided by big IT/OT players) are not so easy to interoperate with (besides being costly, and therefore, not affordable for most SMEs).

Regarding the strong impact Industry 4.0 has on the workforce, several authors have highlighted the importance of systems to evolve in a way to support the newer types of interactions between humans, machines, and systems towards boosting human satisfaction and operational excellence [e.g., 11, 12]. In this context, the emerging concept of *Operator 4.0* ascends to significance. Coined by [13], its vision refers to supporting a socially and cognitive sustainable manufacturing workforce environment in the factories of the future, where smart operators should work collaboratively and aided by machines via advanced human-machine/computer interaction (HMI/HCI) technologies.

The *Operator 4.0 concept* can be implemented using different technologies. When looking at the type 'Smarter Operator 4.0', software robots (or just *softbots*) are one of the most promising approaches [13, 14]. A *softbot* can be generally defined as a computer program that interacts with and acts on behalf of a user or another program [15]. This is in line with the Cognitive DT concept, described as a "digital reflection of

the user, intended to make decisions and carry out tasks on the user's behalf" [16]. [17] have also pointed out the importance of improving the level of DT's autonomy and interactivity in smart manufacturing systems to quickly respond to unexpected events without central re-planning. [18] have called this next-generation of software robots as *Softbots 4.0* – due to their capabilities of assisting operators in the management and automation of business processes, computer systems, and production assets in an industrial networked environment.

The goal of this paper is to present the results of a research that tackles those DT issues. It addresses them by introducing softbots to intermediate the interactions between the DT and production managers in Industry 4.0 production scenarios. This does not mean replacing the native DT's user interfaces, but rather adding a complementary and more "symbiotic" way of HMI/HCI towards supporting and leveraging the "Cognitive Operator 4.0" [13, 20]. This concept refers to empowering workers with more capabilities, autonomy, resources, and interactivity to better perform their activities within smart manufacturing systems [13].

This paper is organised as follows: Sect. 1 highlights the problem to be tackled and the desired improvements; Sect. 2 summarizes the research methodology adopted in this work; Sect. 3 provides basic foundations and related works to softbots and DTs; Sect. 4 details the experimental setup and preliminary results; and Sect. 5 presents the preliminary conclusions of this research and its next short-term steps.

2 Research Methodology

Adopting the *Action-Research methodology,* the prototype was developed as a proof-of-concept with the primary goal of evaluating to which extent the addition of a softbot could cope with the DT issues mentioned in the previous section.

Three previous authors' works have been used as a basis for the prototype. In [14], a group of collaborative *softbots* was developed to interact with a shop floor composed of six workstations of a FESTO MPS system existing in a university lab. In [5], a set of new dialogues, production scenarios and analytics have been implemented, related to the integration of a *softbot* to a Manufacturing Execution System (MES) system from the market. In [10], a DT prototype was built up on top of the same FESTO MPS system, allowing users to remotely follow the production in real-time as well as to act on the system in some situations.

The prototype developed for this work took advantage of some codes and GUIs from those two previous *softbots* and has extended and adapted them to the new interaction scenario involving the developed DT.

3 Basic Foundations and Related Works

3.1 Basic Foundations

Most of the R&D works found out in the literature are not DTs, but "digital shadows" instead. There are three types of virtualizations of a system [21]:

- *Digital Model* – it is a digital representation of a physical object; there is no interaction between the physical and the virtual models.
- *Digital Shadow* – it is characterized by the unidirectional connection between physical and digital models; only the physical model updates the digital model.
- *Digital Twin* – it is also characterized by the bidirectional connection between physical and digital models; the physical and the digital models continuously interact and update each other.

A DT can *mirror* one or more entities of a physical environment, such as operators, machines, materials, and shop floor layouts. In its execution cycle, a DT needs to collect information from its CPSs and send information and commands back to them; CPSs need to send data to its DT, and users need to follow and visualize the CPSs operation via its DT as well as to interact with the DT itself. This involves functionalities for data management, analysis, and computing [22].

A comprehensive DT in a manufacturing domain should support the following general functionalities [21]: (i) real-time monitoring and updating of the DT with information about parts, products, operations, machines, and all the related shop floor entities; (ii) analysis and energy consumption forecast; (iii) intelligent optimization based on the analysis of users' operation and product and/or production process data; (iv) analysis upon users' operation to evaluate their behaviours; (v) virtual product maintenance, including testing, evaluations, and new maintenance plans set-ups; and (vi) failure analysis and forecasting for equipment and/or product maintenance.

A *softbot* can be defined as a *"virtual system deployed in a given computing environment that automates and helps humans in the execution of tasks by combining capabilities of conversation-like interaction, system intelligence, autonomy, proactivity, and process automation"* [23]. It interfaces with smart machines and robots, computers, databases, and other information systems, aiding the operator in the execution of different tasks in "human-like" interactions within Industry 4.0 production and management scenarios [5]. Depending on the softbot's purpose, intelligence, autonomy, architecture, and modelling approaches, a "softbot" can functionally be seen as equivalent to other concepts, such as an *agent, personal assistant, chatbot, bot, holon, or avatar* [18].

Softbots does not aim at replacing the native human-machine/computer interfaces between workers and machine tools, or the ones with enterprise information systems, such as ERP or MES. Instead, they represent a higher-level and new type of frictionless human interaction, also skipping fixed and predefined menus commonly accessed via keyboard and mouse. That interaction can be provided by different means, like web browsers, mobile devices, augmented reality, natural language, haptics, etc. [24].

3.2   Related Works

After a search in scientific databases, it could be observed that the use of softbots (or equivalent concepts) at the shop floor level in the context of Industry 4.0 got some attention since about five years ago [25]. Their use integrated into DTs is more recent. Actually, not many works were found out combining softbots (or equivalent terms) and

DTs. Among them, the large majority only mention it as a trend, both in scientific papers [e.g., 16, 26-30] and in whitepapers, like in[1,2].

To the best of our knowledge, only two works have considered that combination in a more concrete way. In [31], a reference architecture for DTs, called *JARVIS,* is proposed. Although a module to support the interaction with a "chatbot" has been designed, it seems that no implementation has been made yet. In [32], the *Alexa* chatbot (from *Amazon*) is used to interact with a product's digital counterpart (which represents a DT) via voice using a reference vocabulary as a basis. The authors [32] mention an implementation, but no examples are shown in their paper.

4 Experimental Setup and Preliminary Results

4.1 The FESTO MPS System

The manufacturing infrastructure used as the basis for the development of both DT and softbot is a FESTO MPS system. It is composed of six machines (or stations) that work in sequence. Each station is equipped with a *PLC Siemens S7-1200 full DC* and a set of sensors and actuators. They use the *Profinet network and OPC protocol.* Stations' tags statuses and production data are accessed via the *TIA Portal,* from Siemens.

4.2 General System's Architecture

Figure 1 illustrates the general architecture of the developed prototype.

At the local level environment, a *softbot* has been deployed to help local operators to interact with the MPS system more comfortably and informally. Operators can request many different types of data from the planned and ongoing production. Besides that, the softbot was programmed to proactively carry on some actions, for example, checking some predefined situations and going into alarm in some cases [14].

Still, at this level, the operator can be provided with past information (e.g., what has been produced, via accessing the ERP database), current information (e.g., what is being produced), and future information (basically in terms of what is planned to be produced). In these two last cases, the TIA Portal acts as the main source of information.

In terms of the DT, one of its parts is placed at the local level. A *wrapper* was implemented to grab information from the MPS's stations in real-time and store them in the DT database *(MongoDB).* There is a total of 135 variables (tags) that indicate several aspects of the stations during their operation. The communication between the wrapper and the TIA Portal is done via OPC-DA protocol. All this layer runs with Windows 7 OS.

---

[1] Virtual Assistants and Digital Twins Advance Personalised Medicine. https://healthcare-in-europe.com/en/news/virtual-assistants-digital-twins-advance-personalised-medicine.html.

[2] Affective Computing. https://cmte.ieee.org/futuredirections/2020/11/27/megatrends-for-this-decade-x/.
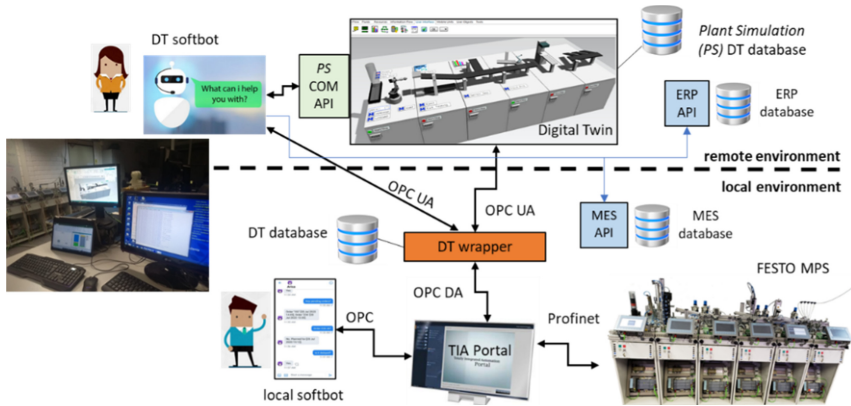
Fig. 1. General systems' architecture

At the remote level environment, a DT has been developed (in *Python*) on top of the MPS [10]. The MPS environment has been drawn up using the *Plant Simulation (PS) software,* from Siemens. The real-time visualization of the MPS operation is done via a permanent access to the DT's database using the OPC-UA protocol. It can be accessed via its Python/COM API, in the same way as its internal database. On the left side of Fig. 1, it can be shown the DT environment with a partial view of the MPS in the background. The monitor/PC on the right side has the DT wrapper/OPC server. The DT appears on the left side, with two screens (the one at the bottom is just a screen extension of the DT visualization environment).

Another *softbot* has been developed, acting as a kind of client application of the DT: the "DT softbot". It can access both other information that is not handled within the DT environment (e.g., the ERP) and the ones dealt with by the DT itself via its API. All this layer runs with Windows 10 OS. The DT and the softbot run on different servers, although in the same local network.

At this level, the operator: can be provided with past information (e.g., what has been produced, via accessing the Plant Simulation and/or the DT databases); can *visualize* current information (e.g., what is being produced, the machines in operation, bottlenecks formation, etc.); can have access to related information from other systems; can monitor and take care of the MPS execution; and can make some (current and future) analytics (e.g., prediction) based on some data and performance indicators.

Considering the objective of this paper and of the associated *proof-of-concept prototype,* the whole architecture and issues related to e.g., the integration approach and security, were simplified. Yet, only the most important MPS stations' entities were graphically represented in the DT software. However, the environment works for a real case, including facing all issues related to integrating different systems, interoperating with different technologies and protocols, and supporting real-time communication.

4.3 The ARISA NEST Platform for Softbots

Both "softbots" have been created using the *ARISA NEST*[3]. It is a PaaS-based-academic tool that allows the derivation and specialization of "instances of" both single and groups of *service-oriented softbots* [14]. They can be accessed via Web or mobile phone. ARISA is free and it is deployed in another cloud/server. It supports the communication between it and the derived softbots in various ways using open protocols/standards.

ARISA NEST supports three types of behaviour modes in its communication with end-users: (i) *Reactive* – when the softbot acts in direct response to users requests via chatting (e.g., to ask about more detailed information from a given machine); (ii) *Planned* – when the softbot acts in response to predefined scheduled tasks (of different types and complexities), bringing their results to users after their execution (e.g., to generate daily performance reports); and (iii) *Pro-active* – when the softbot performs predefined tasks autonomously on behalf of users or of some system it represents, bringing their results to users if needed (e.g., to continuously checking communication problems between data collectors and the DT wrapper - and to promptly take measures to solve them - or sending warnings and alarms).

4.4 The Use Cases

Several scenarios have been devised to preliminary assess this research. For this paper, due to space restrictions, scenarios were mixed, figures were condensed, and only two stations have been considered to leave the figures clearer. These scenarios show the combination of the reactive, planned, and pro-active behaviours, with the capabilities of doing some analytics, actuation on the real system (as a truly DT), and the gathering of information from different systems other than from the DT itself.

Use Case 1: Reactive + Request Decomposition + Access to other Systems + Analytics
In this case, the user checks the DT's analytics module (developed in this prototype) and observes that the *Distribution station's OEE (Overall Equipment Effectiveness)* is strangely too unstable. He then realizes that some production orders will be delayed if this continues as the DT is predicting, but he does not know either which orders might be affected by that (this information is actually handled by the MES system) nor the respective customers' names (this information is handled by the ERP system) so that he could warn both customers and the company's CRM area in advance.

As the DT does not have all the required information, this request is got by the softbot, which "reacts" to attend it. The message is interpreted by the softbot (based on predefined keywords set up when the softbot is derived), it is decomposed into "subtasks" (one for each correct subsystem - the MES and ERP in this case - that should be accessed via calling their APIs to get the information), and each subtask is executed (within/by the softbot). This Use Case is shown in Fig. 2.
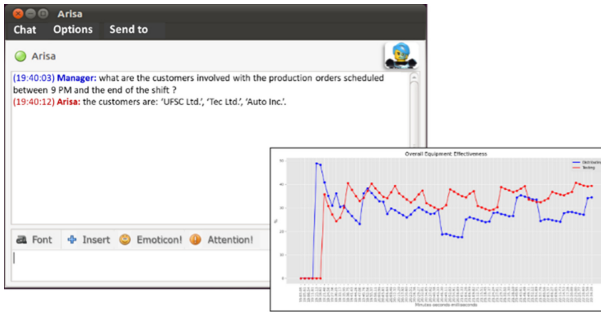

Fig. 2. Use Case 1: Reactive + Request Decomposition + Access to other Systems + Analytics

Use Case 2: Planned + Automatic Actions

In this case, there is no user request. The softbot simply executes the tasks that have been previously configured by the user, on behalf of him. In this scenario, the softbot should automatically generate the utilization level of the *Distribution and Testing stations* at the end of each working shift for his general analysis. As the *ARISA NEST* does not support internally graphical elements in its GUI, it does it calling the Plant Simulation's API to see the utilization level. This Use Case is shown in Fig. 3.
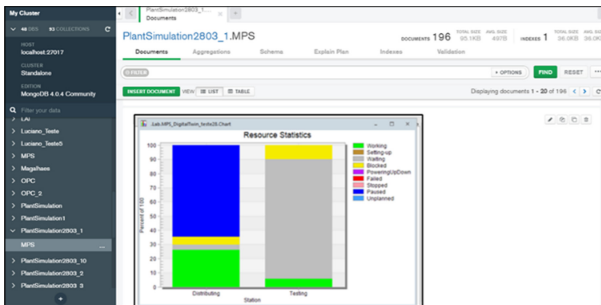
Fig. 3. Use Case 2: Planned + Automatic Actions

Use Case 3: Pro-Active + Actuation on the Real System

In this case, there is not a user request either. However, the pro-active mode relates to the possible autonomy level a softbot can have. For example, a softbot could always warn the user (e.g., sending him a message or getting into alarm) and order a given machine (the Distribution station, for example) to be turned off when its OEE gets lower than 20%. This value of 20% would be a result of a more sophisticated algorithm that calculated that after processing a long OEE historical series also considering the stored company's good practices knowledge base.

In the developed prototype this was just assumed, and a simple algorithm has been implemented to really act on the *physical Distribution station,* to be turned off in that OEE situation. Figure 4 zooms the moment when this situation occurs as well as it shows the change in the respective station's tag, from 'false' (i.e., it was turned on) to 'true' (i.e., it is now turned off).

The possibility to turn a given station off by the user is also allowed via pressing the red button of each station directly on the DT's graphical interface (see Fig. 1). Another example of this Case is to support the Use Case 1. Instead of doing the user to follow the OEE performance of the Distribution station "manually" over time, the softbot proactively and permanently checks that and warns the user when that situation happens. This Use Case is shown in Fig. 4.
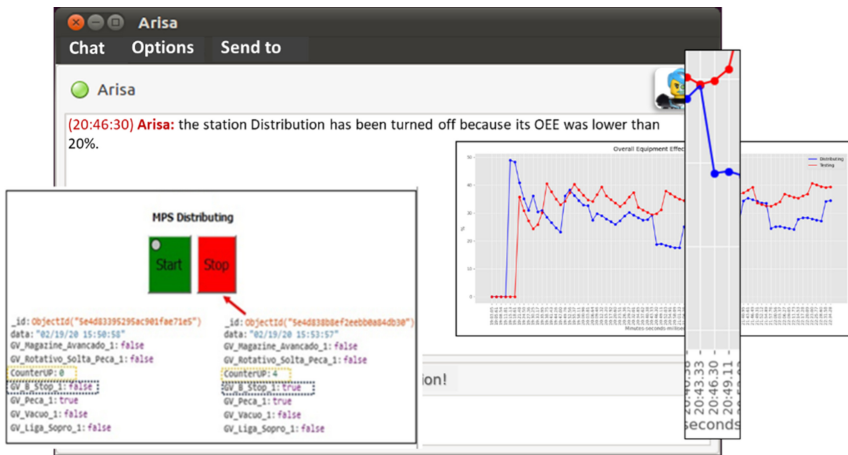

Fig. 4. Use Case 3: Pro-Active + Actuation on the real system

5 Conclusions

This paper has presented an approach for the use of *softbots* to improve the way users interact with DTs and on how they manage the production management in general with manufacturing DTs. An initial prototype has been developed, integrated into a near-real shop floor, which is composed of six machines that act as cyber-physical systems.

A set of scenarios was designed to evaluate the devised approach against the issues highlighted in the introduction. The scenarios were implemented and tested in a controlled environment. The evaluation was qualitative, given the nature of the issues.

Current results could confirm the potential of the approach, where the integration of a *softbot* with a DT environment helped to overcome the targeted issues.

Some open points have arisen from this research. One of the most critical issues refers to the autonomy level a softbot can have, especially when an actuation on the real system is needed. As this is a very sensitive situation, a kind of systems governance

model should exist to determine the frontiers of operation and actuation among the most directed involved systems, namely the MES, the DT, and the softbot.

There are many possible further directions to be exploited from now on. Besides enriching the DT's graphical visualization with more MPS' elements, the main next short-term steps refer to enhancing the interaction between the *softbot* and the DT in terms of better aiding users in business analyses and decision-support based on the data the DT and other systems (including wearables) can jointly provide.

References

1. Mittal, S., Khan, M.A., Romero, D., Wuest, T.: Smart manufacturing: characteristics, technologies and enabling factors. J. Eng. Manuf. **233**(5), 1342–1361 (2017)
2. Barricelli, B.R., Casiraghi, E., Fogli, D.: A survey on digital twin: definitions, characteristics, applications, and design implications. IEEE Access **7**, 167653–167671 (2019)
3. Romero, D., Vernadat, F.: Enterprise information systems state of the art: past, present and future trends. Comput. Ind. **79**, 3–13 (2016)
4. ASG: The ASG Technologies 2020 Survey Report – Barriers to Success with Information-Powered Business: Bad Data Burdens the Modern Enterprise. https://content.asg.com/MarketingCollateral/DataIntelligence/whitepaper-asg-data-intelligence-barriers-to-success.pdf
5. Abner, B., Rabelo, R.J., Zambiasi, S.P., Romero, D.: Production management as-a-service: a softbot approach. In: Lalic, B., Majstorovic, V., Marjanovic, U., von Cieminski, G., Romero, D. (eds.) Advances in Production Management Systems. Towards Smart and Digital Manufacturing, APMS 2020. IFIP Advances in Information and Communication Technology, vol. 592. Springer, Cham. https://doi.org/10.1007/978-3-030-57997-5_3
6. McDermott, A.: Information Overload Is Crushing You. Here are 11 Secrets That Will Help. https://www.workzone.com/blog/information-overload/
7. Jones, D., Snider, C., Nassehi, A., Yon, J., Hicks, B.: Characterising the digital twin: a systematic literature review. CIRP J. Manuf. Sci. Technol. **29**(1), 36–52 (2020)
8. Rabelo, R.J., Zambiasi, S.P., Romero, D.: Collaborative softbots: enhancing operational excellence in systems of cyber-physical systems. In: Camarinha-Matos, L.M., Afsarmanesh, H., Antonelli, D. (eds.) PRO-VE 2019. IAICT, vol. 568, pp. 55–68. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-28464-0_6
9. Lua, Y., Liub, C., Xu, X.: Digital twin-driven smart manufacturing: connotation, reference model, applications and research issues. Robot. Comput. Integr. Manuf. **61**, 101837–101852 (2020)
10. Rabelo, R.J., Magalhaes, L.C., Cabral, F.G.: A proposal of a reference architecture for digital twins in industry 4.0 (in Portuguese). In: Proceedings of the XXIII Brazilian Congress of Automation (2020)
11. Rooein, D., Bianchini, D., Leotta, F., Mecella, M., Paolini, P., Pernici, B.: Chatting about processes in digital factories: a model-based approach. In: Nurcan, S., Reinhartz-Berger, I., Soffer, P., Zdravkovic, J. (eds.) BPMDS/EMMSAD-2020. LNBIP, vol. 387, pp. 70–84. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-49418-6_5

12. Frazzon, E.M., Agostino, I.S., Broda, E., Freitag, M.: Manufacturing networks in the era of digital production and operations: a socio-cyber-physical perspective. Annu. Rev. Control. **49**, 288–294 (2020)

13. Romero, D., Stahre, J., Wuest, T., et al.: Towards an operator 4.0 typology: a human-centric perspective on the fourth industrial revolution technologies. In: International Conference on Computers & Industrial Engineering (CIE46), Tianjin, China, pp. 1–11 (2016)

14. Rabelo, R.J., Romero, D., Zambiasi, S.P.: Softbots supporting the operator 4.0 at smart factory environments. In: Moon, I., Lee, G.M., Park, J., Kiritsis, D., von Cieminski, G. (eds.) APMS 2018. IAICT, vol. 536, pp. 456–464. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-99707-0_57

15. Schwartz, T., et al.: Hybrid teams: flexible collaboration between humans, robots and virtual agents. In: Klusch, M., Unland, R., Shehory, O., Pokahr, A., Ahrndt, S. (eds.) MATES 2016. LNCS (LNAI), vol. 9872, pp. 131–146. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-45889-2_10

16. Somers, S., Oltramari, A., Lebiere, C.: Cognitive twin: a cognitive approach to personalized assistants. In: AAAI Spring Symposium: Combining Machine Learning with Knowledge Engineering (2020)

17. Rosen, R., Wichert, G., Lo, G., Bettenhausen, K.: About the importance of autonomy and digital twins for the future of manufacturing. IFAC-PapersOnLine **48**(3), 567–572 (2015)

18. Rabelo, R., Zambiasi, S.P., Romero, D., Abner, B.: Softbots 4.0: excelling smart production management. Int. J. Comput. Integr. Manuf. (to appear)

19. Brain. Chatbot Report 2019: Global Trends and Analysis. https://chatbotsmagazine.com/chatbot-report-2019-global-trends-and-analysis-a487afec05b/

20. Thorvald, P., Fast-Berglund, Å., Romero, D.: The cognitive operator 4.0. In: Proceedings of the 18th International Conference on Manufacturing Research. IOS Press (2021)

21. Kritzinger, W., Karner, M., Traar, G.: Digital twin in manufacturing: a categorical literature review and classification. IFAC PapersOnLine **15**(11), 1016–1022 (2018)

22. Zhong, R.Y., Xu, X., Klotz, E.: Intelligent Manufacturing in the Context of Industry 4.0: A Review. Engineering, 3(5):616–630 (2017)

23. Kim, J.H.: Ubiquitous robot. In: Reusch B. (eds.) Computational Intelligence, Theory and Applications. Advances in Soft Computing, vol. 33, pp. 451–459 (2005)

24. Papcun, P., Kajáti, E., Koziorek, J.: Human machine interface in concept of Industry 4.0. In: World Symposium on Digital Intelligence for Systems and Machines, pp. 289–296 (2018)

25. Preuveneers, D., Ilie-Zudor, E.: The intelligent industry of the future: a survey on emerging trends, research challenges and opportunities in Industry 4.0. J. Ambient Intell. Smart Environ. **9**(3), 287–298 (2017). https://doi.org/10.3233/AIS-170432

26. Rasheed, A., San, O., Kvamsdal, T.: Digital twin – values, challenges and enablers from a modeling perspective. IEEE Access **8**, 21980–22012 (2020)

27. Yarnold, J., Henman P., McEwan, C.: Artificial intelligence. https://policy-futures.centre.uq.edu.au/files/6128/PolicyFutures_Unpacking_Artificial_Intelligence.pdf

28. Guo, J.: Digital twins are shaping future virtual worlds. SOCA **15**(2), 93–95 (2021)

29. Boschert, S., et al.: Symbiotic Autonomous Systems. IEEE Digital Reality (2019)

30. Bolton, R., McColl-Kennedy, J., et al.: Customer experience challenges: bringing together digital, physical and social realms. J. Serv. Manag. **29**(5), 776–808 (2018)

31. Parri, J., Patara, F., Sampietro, S.: A framework for model-driven engineering of resilient software-controlled systems. Computing **103**, 589–612 (2021)

32. Wellsandta, S., Foosheriana, M., Thoben, K-D.: Interacting with a digital twin using Amazon Alexa. In: 5th International Conference on System-Integrated Intelligence, pp. 4–8 (2020)