Available online at www.sciencedirect.com

ScienceDirect

Procedia Manufacturing 39 (2019) 66–73

Procedia MANUFACTURING

www.elsevier.com/locate/procedia

25th International Conference on Production Research Manufacturing Innovation: Cyber Physical Manufacturing
August 9-14, 2019 | Chicago, Illinois (USA)

Virtual Commissioning of Industrial Control Systems - a 3D Digital Model Approach

Matthias Schamp[a,b,*], Lauren Van De Ginste[a,b], Steven Hoedt[a,b], Arno Claeys[a,b], El-Houssaine Aghezzaf[a,b], Johannes Cottyn[a,b]

[a] *Department of Industrial Systems Engineering and Product Design, Ghent University, Technologiepark 46, 9052 Gent-Zwijnaarde, Belgium*
[b] *Industrial Systems Engineering (ISyE), Flanders Make, www.FlandersMake.be*

Abstract

With the growing presence of industry 4.0, flexible workstations and distributed control logic, software development has become an even more important part of the automation engineering process than before. In a traditional workflow, the main commissioning part of industrial control systems is performed on the real set-up and consequently during a time critical phase of the project. Virtual commissioning can be used to reduce the real commissioning time and can allow an earlier commissioning start, reducing the overall project lead time, risk of damaging parts, amount of rework and cost of error correction. Previous research showed already a reduction potential of the real commissioning time by 73%, when using a virtual commissioning strategy based on a 3D digital model. However, the robustness of that approach still highly depends on the human expertise to fully evaluate the correct behavior in all possible use scenarios. This paper describes an approach to further automate these virtual commissioning steps by embedding functional specifications and use scenarios through a formal notation inside the 3D digital model. Configuration steps inside the virtual environment describe the conditions, independent from the control logic but related to component states and transitions in the digital model (actuator and sensor values, time restrictions, counters, positions of objects, etc.). These conditions are continuously monitored during an extensive commissioning run of the digital model covering all possible component states and transitions. A small scale experiment will show the reduction of the virtual commissioning time and earlier detection of quality issues.

* Corresponding author. Tel.: +32-(0)56-241-221; fax: +32-(0)56-241-224.
E-mail address: matthias.schamp@ugent.be

2351-9789 © 2019 The Authors. Published by Elsevier Ltd.
This is an open access article under the CC BY-NC-ND license (https://creativecommons.org/licenses/by-nc-nd/4.0/)
Peer-review under responsibility of the scientific committee of the ICPR25 International Scientific & Advisory and Organizing committee members
10.1016/j.promfg.2020.01.229

Keywords: Virtual commissioning; 3D digital model; Industrial control system; Functional specification; Use scenarios

1. Introduction

Modern production plants face a continuously increasing complexity. The automation systems of production plants are critical to ensure safe, productive and qualitative operations. The correct behavior of these automation systems are essential, making the early testing of the engineered automation solution and software a critical task to reduce the risks during the real commissioning and plant operation [1]. The importance of software will even increase due to the evolution towards industry 4.0, industrial internet of things (IIOT), flexible production, smaller batch sized, one-piece flow and more customized products [2]. That additional complexity puts pressure on the real commissioning phase of a project, which consists of a lot of (manual) software debugging.

This section will introduce 2 key elements of this research: a digital model and virtual commissioning of industrial control systems. Section 2 gives an overview of the used approaches for the commissioning of control systems and their differences**Error! Reference source not found.**. A conclusion and outlook for further research is presented in section 3.

1.1. Digital model

Within the concept of Industry 4.0, the Digital Twin (DT) concept has gained more momentum in the industry sectors as a method for managing increasing system's complexities and as a link to the digital world, thus facilitating the application of intelligent functions and services to the physical systems [3,4]. Recent research summarizes the various definitions of DT and proposes a classification into three subcategories, according to their level of integration.

- Digital model: A digital representation of an existing or planned physical object, capable of fully mirroring its characteristics and functionalities. There is no automated data exchange between the physical and the digital object.
- Digital shadow: Based on the definition of a digital model, extended with an automated one-way interaction between the state of an existing physical object and the corresponding digital model.
- Digital Twin: A model, where the dataflow between the digital and the physical object is fully integrated in both directions, is called a Digital Twin. A change in state of one of the objects leads to a change in state of the other objects.

The digital model in this paper is a 3 dimensional graphical representation of the physical object [5]. Beside the graphical representation, the model also contains the dynamic behavior and information about the sensors and actuators used in the physical object.

1.2. Virtual commissioning

Kormann [6] summarizes that the most frequent fault in machine or production system control software is the missing reaction on undefined machine or system states. Also the importance of testing is shown, but this is mostly a manual test in a non-operative state and late in the project. Virtual commissioning of industrial control systems enables control software engineering and testing earlier in the design process of production systems. Virtual commissioning involves replicating the behavior of hardware within a software environment. The proposed design procedure will provide a virtual plant that can interact with real controller hardware. The behavior of the digital model should be the same as that of the real production system. The virtual components have to include both the mechanical aspects (a geometric model and a kinematic model) and the electrical aspects (the behavior model to interact with a programmable logic controller (PLC)). The goal is to validate the control logic prior to real implementation, enabling a seamless transition from the virtual to the real environment. Virtual commissioning reduces the effort and costs for error correction and shortens the project lead time (Fig. 1)) [7–9].

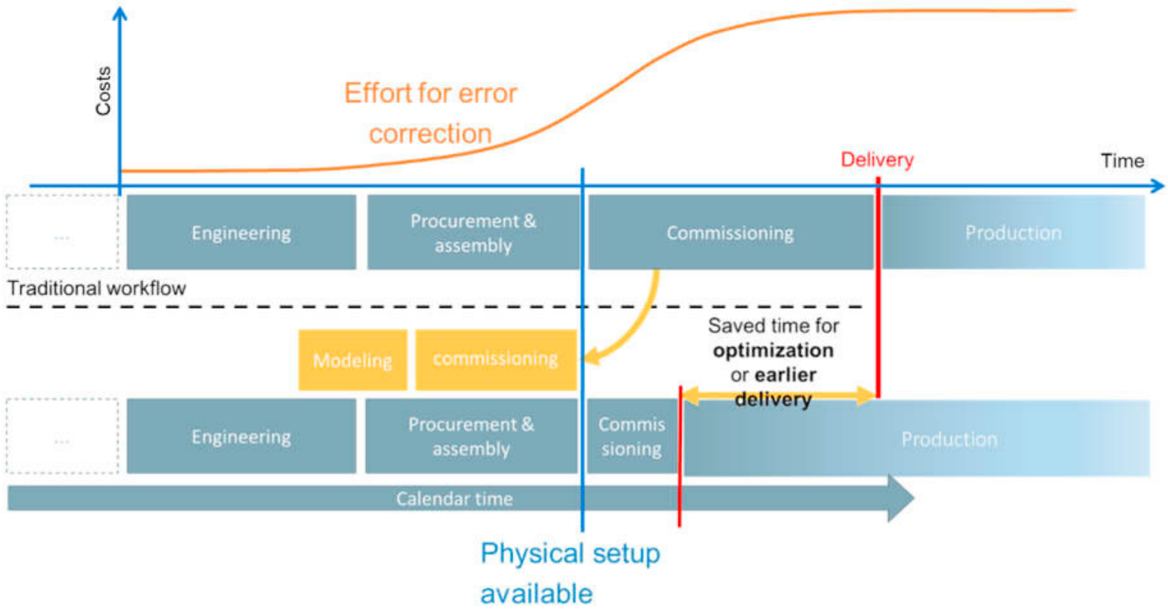

Fig. 1: Automation project timeline and related error correction costs

Earlier small scale tests showed the potential of using a digital model based verification of the control logic. The experiment showed an improved quality of 31% in terms of fulfilled requirements before real commissioning and a reduction of real commissioning time by 73% [10]. In this experiment the commissioning phase with the digital model was still a manual task, resulting in the uncertainty that all the requirements were actually checked. A person also had to spend time to execute all possible (good and bad) execution scenarios and watch the emulation results. Typical simulation tools that enable engineers to virtually commission the control systems are WinMOD, Emulate3D, TarakosVR, Experior, Visual components and Mechatronics concept designer [11–14].

2. Virtual commissioning approaches

A difference is noticeable between the use of virtual commissioning today by Flemish automation companies and recent literature. This section gives an overview of the current approaches of the Flemish companies and describes the benefits and disadvantages of this approach. Next, an overview of the state-of-the-art in literature is given, focused on the automated verification of the control logic. Again the benefits and disadvantages are listed. This proposed approach tries to reduce the drawbacks, while keeping the benefits of the other approaches. This section ends with a table that summarizes some properties of the different approaches.

2.1. State-of-the-use

Currently, most of the commissioning of industrial control systems is done on the real hardware, also known as real commissioning. This results in a long commissioning phase, a late detection of errors and a high cost and effort to correct the error. To partially solve this, manual virtual commissioning is typically introduced in the workflow. The control logic can be verified early in the process by running it on (the simulator of) the real controller and interacting with the program by overwriting or forcing the inputs to emulate the behavior of the system. The operator needs to know a lot of the behavior of the system. A common way to partially automate this manual virtual commissioning approach is to add additional code to the control logic that emulates the behavior of the system. The verification itself is still a manual task to interpret the results.

2.2. State-of-the art

The automated verification of control logic is of main interest and is particularly researched in control logic engineering. Verification methodologies typically specify the control logic using a formalism such as state machines through Unified Modeling Language (UML) or Petri nets. The specifications are converted into models, which then are automatically checked [15].

For automated checks to be applicable, six requirements are identified [16]:

- Support of industrial software properties: The programming standard IEC61131-3 for PLC has to be supported.
- Real time capability and memory size: The test should not influence the real time properties of the tested system. The needed real time capabilities are seen as unaffected if a possible increase in execution time of modified code does not lead to the PLC cycle time to be exceeded. In addition, possibly increased size of compiled control software code should not lead to exceeded memory on the execution hardware.
- Inclusion of hardware and process behavior: Testing a system integrates all parts of the system, meaning software, hardware and the controlled process. To be able to assess a system's conformance to its specification, all parts should be as similar to the final system as possible, i.e. the software running on the final execution hardware, controlling the final version of the process.
- Manipulation of hardware and process behavior: The tests include manual manipulations of the hardware or technical process that cannot be performed by the software.
- No need for formalized functional requirements: The operator does not have to translate the requirements of the process to a formalized language. More manual actions results in a larger possibility of mistakes and additional knowledge.
- Finding untested behavior: to increase efficiency and quality during the quality assurance process of special purpose machinery by supporting the tester, who might be an experienced software engineer or inexperienced technician, when assessing the test adequacy.

Ovatman [17] gives an overview of model checking practices on control logic. This research shows that most of these checking practices do not fulfill all requirements given earlier. An area of concern in those approaches is that the transition to formal specifications is mostly a manual interaction. The analyzed techniques during this research are Sequential Function Chart (SFC), Petri nets, PLC-Automata, Timed Automata, Condition Event Systems, UML and MATLAB State chart. Research states that PLC program verification is an ideal target for model checking due to the medium criticality and the relatively simple programs. But semantics and syntax of PLC programs are complex, which makes it difficult for non-PLC experts to contribute to verification as the needed knowledge for PLC program verification is high. A bridge is needed between the formal verification researchers and the industrial control systems domain [18]. The conversion and automated testing of industrial control systems has some challenges. Research demonstrates the possibility to automate the generation of test cases based on a Timed Synchonizable I/O Automation [19]. The idea of using formal models in software testing is certainly not new. Multiple formal techniques for representing implementations, specifications and test suites are known. A popular approach is to use finite-state models, such as deterministic finite-state machines (FSMs) [20]. While simulation is open-ended and fraught with uncertainty, formal verification is definitive and eliminates uncertainty. In temporal-logic model checking, the correctness of a finite-state system with respect to a desired property is verified by checking whether the system satisfies a formula that specifies this property. A labeled state-transition graph that models the system is used for this verification. A key issue in the design of a model-checking tool is the choice of the temporal language used to specify properties. This language, referred to as the temporal property-specification language, is one of the primary interfaces to the tool. In linear temporal logics (LTL), time is treated as if each moment in time has a unique possible future. In branching temporal logics, like computation tree logic (CTL), each moment in time may split into various possible futures. The formulas can express reachability, safety and liveliness properties through several syntaxes. CTL is unintuitive for verification engineers, where LTL is considered as a more simple, natural language for verification engineers. Both languages require additional knowledge from the engineer to understand the formulas. One of the conclusions of earlier research is that LTL is not expressive enough to verify concurrent programs or assumptions about the environment in modular verification. The use of model-checking techniques modify considerably engineers work. It is absolutely necessary to think about the way to integrate them in the controller design workflow [21,22].

2.3. Proposed approach

The proposed approach combines the use of a digital model and formal verification techniques to reduce the manual activities during the commissioning phase of industrial control systems. This allows to automate the verification of the control logic, which reduces the required skills and still delivers a visual representation of the behavior of the system. A digital model contains a lot of information of the system. This information can be reused to define testing scenarios and conditions and to generate control logic. Also geometry and physical behavior like friction, gravity and forces are included in the emulation, resulting in a more realistic verification than other approaches. With this information, different states of the system or subsystems can be generated and visually represented in the digital model. The user can connect the different states and define the allowed states and transitions. The system is considered as the connection of all the parts in the model, which can be a whole production plant, a production line or just one machine. It is the upper model in the hierarchical structure of the model. A subsystem is a part of the system or a larger subassembly. This can be a subassembly of a machine, a unit for a subtask, parts responsive for a specific motion, etc. Applied to our system, a didactical Festo application, 5 subsystems can be identified: a warehouse, a first transportation element, a buffer unit, a second transportation unit and a sorting station. The second transportation unit can again be subdivided in subsystems: a horizontal movement, a vertical movement and a gripper.

The Inputs/Outputs (I/O) and structured lay-out allows to generate the possible states of the system. Some of the transitions, that are not immediately related to changes of I/O, cannot be generated automatically but they can easily be added and configured by the provided Graphical User Interface (GUI). Examples of these transactions are transitions based on counter values, analog signals and timers. Transitions related to changes of I/O are for example the detection of proximity switches, detection by inductive or capacitive sensors, etc. The user can graphically model the possible states and transitions by connecting them by lines. This approach reduces the needed knowledge of state machines and Petri nets, but still keeps a formal language and software independent notation, resulting in a low entry threshold. Besides the reduction of needed knowledge of the formal notation, the digital model interacts with the real control logic, reducing the amount of conversions. The automated verification of the transition reduces the influence of test expertise. To define the transitions, no knowledge of programming languages is needed. Table 1 shows that the proposed approach tries to combine the benefits of the current state-of-the-art and state-of-the-use.

Table 1. Summary of virtual commissioning approaches

	State-of-the-use	Proposed approach	State-of-the-art
Real control logic	Yes	Yes	No
Visual representation	No	Yes	No
Need for formalized requirements	No	Yes, but in background	Yes
Inclusion of real hardware	Possible	Possible	No
Additional modeling required	Not always	Yes	Yes
Additional modelling reusable later in project lifecycle		Yes	No
Additional knowledge required	Knowledge about system behavior	Limited by use of visual representation	Knowledge about system behavior and formal languages
Location in automation project	At the end, when physical set-up is available	During programming	During programming

A demonstration of this approach is developed in two software environments to illustrate that it is software independent. A first demonstration is available in Mechatronics Concept Designer (MCD). In this demonstration, the additional functionality is added by using the NXOpen application programming interface (API). The second demo is made in Visual components. This software tool also provides an API that allows to add functionality to the default program. Fig. 2 shows the interface in MCD to define the allowed states and transitions for each subsystem. This structure can be hierarchical, where a subsystem can be a part of a larger subsystem. In the additional screen, the states of the selected subsystem are shown. When no subsystem is selected (as in Fig. 3) or a larger subsystem is selected, the states and transitions of the whole system are visualized in the additional screen.

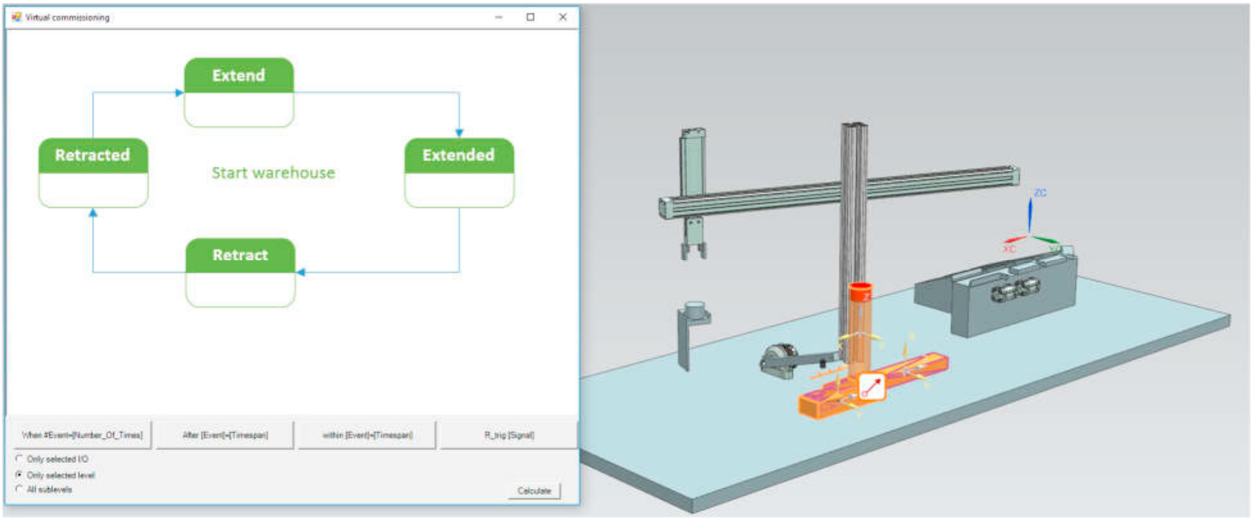

Fig. 2: Defining states and transitions on a subsystem

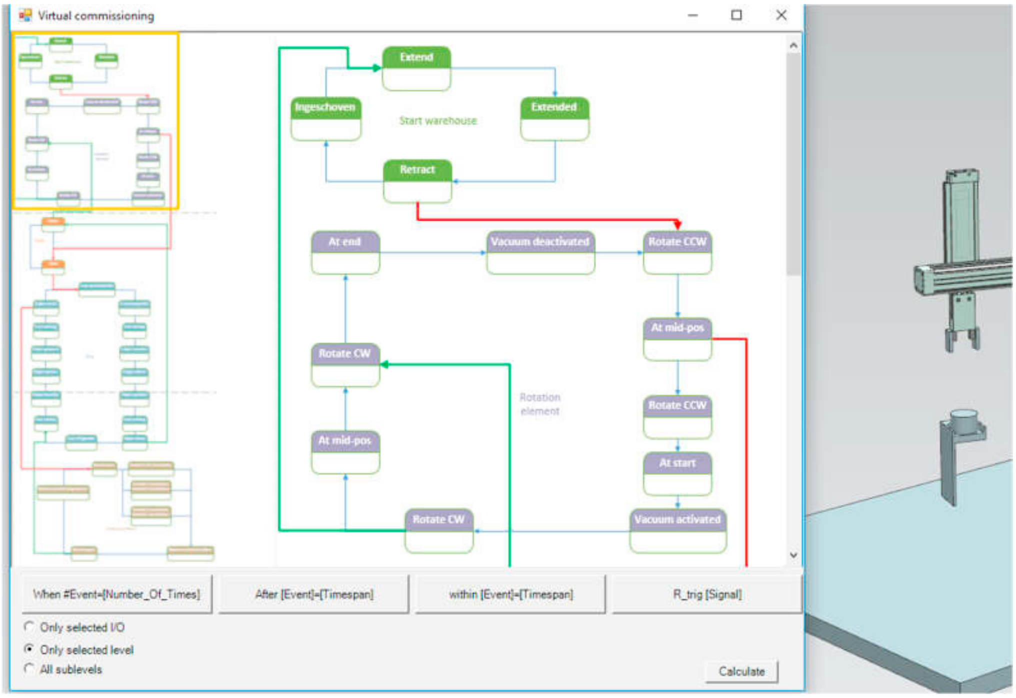

Fig. 3: Defining states and transitions between subprocesses, no subprocess selected

During verification, the current state of each of the subsystems is highlighted to give the operator some visual feedback. The software continuously monitors the behavior of the control system through the states and transitions. When a transition occurs that is not defined, it is detected as an error. A video is saved and a report is generated of the error. With this approach, the verification can happen in the background without having to watch and interpret the scenarios. The approach will automate the verification of the real PLC code. The automated verification checks all the known requirements on each run. This way, it will also reduce unintended mistakes by fixing or changes other steps in the program.

This proposed approach will be verified later by a small scale experiment. During this experiment, the benefits and disadvantages will be verified and quantified. To allow comparison, the same project will be programmed and verified by 2 groups, one this approach, the other group with the traditional approach. Values as programming duration, duration of virtual and real commissioning and the number of iterations until the control logic fulfilled all the requirements will be measured. User experience is also important, although it's still a concept demonstration, and the lay-out will change during the research. If this new approach shows clear benefits, larger industrial projects will be executed to verify the first results on real industrial cases of different size, different project type (material handling, process control,…), different industrial logic controllers and devices, etc. The difficulty of this second stage is that commissioning of a project can only happen once, so comparing results with a traditional approach in terms of duration is not straight forward.

3. Conclusion and outlook

Earlier case studies show the high potential of the use of a digital model of production systems, but they also address some hurdles that prevent the digital model to be integrated in the default engineering workflow. Our proposed approach tries to reduce the amount of manual effort needed during the virtual commissioning phase of an industrial automation project. To reach this goal, the benefits of a digital model are combined with these of automated formal verification, which results in the reduction of the disadvantages of both techniques. By still using the visual representation, the required knowledge about formal languages can be reduced. This approach was illustrated in 2 emulation software tools. A major part of the virtual commissioning approach is the interaction of the model with the real control logic, which eliminates logic conversions. Future steps in this research will be to quantify the benefits of this approach in terms of reduction in manual effort and improvement of the commissioning phase compared to the current traditional approach and to automatically generate control logic based on the digital model. To quantify the benefits of our approach compared to the traditional digital model virtual commissioning approach a series of experiments will be conducted. To compare both approaches, the reference group uses the traditional approach, while the test group uses the proposed approach on the same project. For both groups, the virtual and real commissioning time, the amount of virtual test scenarios and the number of iterations of the control logic needed to fulfill the project during real and virtual commissioning are logged. In the first phase, an industrially relevant, but small scale experiment will be conducted in a controlled lab environment. The experiment will compare the needed virtual commissioning time and the amount of unfulfilled requirements of both groups. The reduction of virtual commissioning time will be an indication of the reduction of spent time during industrial automation projects. This will give a first indication in what can be expected during industrial cases.

Later on, it will be possible to add automated code generation to the workflow. The information of the allowed states and transitions also enables the possibility to automatically generate the control logic or reuse code for components from a library. This is a second benefit of our approach and allows the reuse of the model. In first stage, not all the code will be generated automatically. During and after finalizing the control logic manually, the verification approaches discussed earlier can be used to test the manual changes in code and so close the loop. Later in this research, the code generation will be extended to generate more code automatically. To close the loop, the adjusted code can be verified with the digital model. During the verification, the model uses the states and transitions defined earlier in the engineering process to automatically verify the conditions. This automatic verification ensures that all the conditions are tested in all of the testing scenarios. After the verification, a report is generated with the fail/pass results of the testing scenarios, unmet conditions, throughput results, etc. Video files of all the tests are stored to give the operator a visual representation of the test results and problems. By using the real control logic, the possibility to connect the digital model to the real controller, simulation of the process behavior in the model and logging of tested transition, all the requirements identified in 2.2 are fulfilled. The formalization is done in the background, it is thus not noticeable by the automation engineer.

References

[1] M. Oppelt, L. Urbas, Integrated virtual commissioning an essential activity in the automation engineering process: From virtual commissioning to simulation supported engineering, IECON Proc. (Industrial Electron. Conf. (2014) 2564–2570. doi:10.1109/IECON.2014.7048867.
[2] R. Rosendahl, N. Schmidt, A. Luder, D. Ryashentseva, Industry 4.0 value networks in legacy systems, in: 2015 IEEE 20th Conf. Emerg. Technol. Fact. Autom., IEEE, 2015: pp. 1–4. doi:10.1109/ETFA.2015.7301598.
[3] B.A. Talkhestani, N. Jazdi, W. Schloegl, M. Weyrich, Consistency check to synchronize the Digital Twin of manufacturing automation based on anchor points, Procedia CIRP. 72 (2018) 159–164. doi:10.1016/j.procir.2018.03.166.
[4] T.H.J. Uhlemann, C. Lehmann, R. Steinhilper, The Digital Twin: Realizing the Cyber-Physical Production System for Industry 4.0, Procedia CIRP. 61 (2017) 335–340. doi:10.1016/j.procir.2016.11.152.
[5] W. Kritzinger, M. Karner, G. Traar, J. Henjes, W. Sihn, Digital Twin in manufacturing : A categorical literature review and classification, (2018).
[6] B. Kormann, D. Tikhonov, B. Vogel-Heuser, Automated PLC Software Testing using adapted UML Sequence Diagrams, IFAC Proc. Vol. 45 (2012) 1615–1621. doi:10.3182/20120523-3-RO-2023.00148.
[7] N. Shahim, C. Møller, Economic justification of virtual commissioning in automation industry, in: Winter Simul. Conf. (WSC), 2016, 2016: pp. 2430–2441.
[8] M. Ko, E. Ahn, S.C. Park, A concurrent design methodology of a production system for virtual commissioning, Concurr. Eng. Res. Appl. 21 (2013) 129–140. doi:10.1177/1063293X13476070.
[9] Z. Liu, N. Suchold, C. Diedrich, Virtual Commissioning of Automated Systems, in: Automation, InTech, 2012: pp. 131–148. doi:10.5772/45730.
[10] M. Schamp, S. Hoedt, A. Claeys, E.H. Aghezzaf, J. Cottyn, Impact of a virtual twin on commissioning time and quality ., Proc. 16th IFAC Symp. Inf. Control Probl. Manuf. (2018).
[11] O. Johansson, Testing and Evaluation of Virtual Commissioning Case study of an existing robot cell at Scania modelled with 3DExperience Master's thesis in Systems, Control and Mechatronics, CHALMERS UNIVERSITY OF TECHNOLOGY, 2017.
[12] M. Oppelt, G. Wolf, O. Drumm, B. Lutz, M. Stöß, L. Urbas, Automatic Model Generation for Virtual Commissioning based on Plant Engineering Data, IFAC Proc. Vol. 47 (2014) 11635–11640. doi:10.3182/20140824-6-ZA-1003.01512.
[13] C.G. Lee, S.C. Park, Survey on the virtual commissioning of manufacturing systems, J. Comput. Des. Eng. 1 (2014) 213–222.
[14] S. Seidel, U. Donath, J. Haufe, Towards an integrated simulation and virtual commissioning environment for controls of material handling systems, in: Proc. Title Proc. 2012 Winter Simul. Conf., IEEE, 2012: pp. 1–12. doi:10.1109/WSC.2012.6465081.
[15] G.F. Schneider, H. Wicaksono, J. Ovtcharova, Virtual engineering of cyber-physical automation systems: The case of control logic, Adv. Eng. Informatics. 39 (2019) 127–143. doi:10.1016/j.aei.2018.11.009.
[16] S. Ulewicz, B. Vogel-Heuser, Increasing system test coverage in production automation systems, Control Eng. Pract. 73 (2018) 171–185. doi:10.1016/j.conengprac.2018.01.010.
[17] T. Ovatman, A. Aral, D. Polat, A.O. Ünver, An overview of model checking practices on verification of PLC software, Softw. Syst. Model. 15 (2016) 937–960. doi:10.1007/s10270-014-0448-7.
[18] D. Darvas, E. Blanco Viñuela, G. Switzerland, What is Special About PLC Software Model Checking?, (2017). doi:10.18429.
[19] K. Pinkal, O. Niggemann, A new approach to model-based test case generation for industrial automation systems, Proc. - 2017 IEEE 15th Int. Conf. Ind. Informatics, INDIN 2017. (2017) 53–58. doi:10.1109/INDIN.2017.8104746.
[20] I. Buzhinsky, C. Pang, V. Vyatkin, Formal Modeling of Testing Software for Cyber-Physical Automation Systems, Proc. - 14th IEEE Int. Conf. Trust. Secur. Priv. Comput. Commun. Trust. 2015. 3 (2015) 301–306. doi:10.1109/Trustcom.2015.648.
[21] M.Y. Vardi, Branching vs. Linear Time: Final Showdown, (2007) 1–22. doi:10.1007/3-540-45319-9_1.
[22] A. Philippot, B. Riera, V. Kunreddy, S. Debernard, Advanced tools for the control engineer in Industry 4.0, Proc. - 2018 IEEE Ind. Cyber-Physical Syst. ICPS 2018. (2018) 555–560. doi:10.1109/ICPHYS.2018.8390766.