

Research Article

Design and Implementation of Runtime Verification Framework for Cyber-Physical Production Systems

Sungjoo Kang ¹, Ingeol Chun ¹, and Hyeon-Soo Kim ²

¹Artificial Intelligence Research Laboratory, Electronics and Telecommunications Research Institute (ETRI), Daejeon 34129, Republic of Korea

²Department of Computer Science and Engineering, Chungnam National University, Daejeon 34134, Republic of Korea

Correspondence should be addressed to Hyeon-Soo Kim; hskim401@cnu.ac.kr

Received 5 April 2019; Accepted 23 October 2019; Published 13 November 2019

Guest Editor: Nader Ben Amor

Copyright © 2019 Sungjoo Kang et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Traditional factories are turning into smart factories with the advent of various ICT technologies, and various control decisions are derived by AI technologies. In this circumstance, runtime verification of a control command is important for zero-defect manufacturing processes but challengeable because factories of the future are highly complex and heterogeneous systems. In this paper, we propose DigTwinOps, a Digital Twin framework for Runtime Verification of Cyber-Physical Production Systems (CPPSs). DigTwinOps features a Digital Twin Execution Engine (DTEE) that manages a Digital Twin Model to synchronize states of a real CPPS object in a production environment. With a monitoring and simulation combination process, a human worker can observe the states of the CPPS object and verify the effectiveness of control commands before applying it to a real production environment. The proposed framework is applied to a CPPS prototype production system, and the results show that the framework can work effectively in the controllability verification of control commands.

1. Introduction

Traditional factories are turning into smart factories with the advent of various ICT technologies such as wireless sensor networks (WSNs), artificial intelligence (AI), and cyber-physical systems (CPSs) [1]. With WSNs technologies, machines on a shop floor (a production environment) are getting smarter and more connected [2]. Big data collected from the network of smart machines are analysed in an operational environment of the factory, and autonomous decisions can be derived by AI technologies. However, research on how to build a new factory into a smart factory or how to convert an existing factory into a smart factory is insufficient. Therefore, the production environment (the physical world) and the operational environment (a cyber world) should be integrated to build a CPS-based production system, or so-called Cyber-Physical Production System (CPPS) which is an extremely promising technology of Industry 4.0 and an essential component of a smart factory [3].

One of the key issues in CPPS is the management of a control loop, which is the fundamental building block of industrial control systems [4–6]. It manages the decision cycle (observation, analysis, decision, and action) between the production environment and the operational environment. The control loop consists of all the physical and cyber components to autonomously adjust states of the production environment to equal the value of a desired state [7]. Various industrial control network technologies [8–10] make it possible to observe real-time states of the production environment. Recent advances in big data and AI technologies have led to the development of AI-based decision-making applications [11]. However, applying a control command decided by an AI application to the production environment could be dangerous when the controllability of the decided action is not verified. Controllability is defined as the ability of a control input to move the internal state of a system from any initial state to any other final state. The controllability verification of the control command which is scheduled to be applied to the production environment is important for

zero-defect manufacturing processes but challengeable because CPPSs are highly large-scale distributed and heterogeneous systems [12]. Several studies have been devoted to developing methodologies for verification of CPPS, e.g., Ptolemy [13], ACME [14], DEVS BUS [15], and FILCon [16]. Ptolemy, which is a design and verification tool for embedded system, provides functionality for analysing networked embedded system behaviours. However, it is difficult to integrate with real manufacturing objects for runtime verification. ACME, which is software architecture design and verification toolkit, expands functionality for representing and analysing heterogeneity of CPS's behaviours. However, it is only able to verify single system level and lacks precise analysis of physical dynamics. DEVS BUS provides simulation environment for networked discrete event systems. However, not only it requires external modules such as Simulink and HLA-RTI but also it requires additional implementation for runtime verification. FILCon is a MES-level framework which is able to support various manufacturing applications such as monitoring, simulation, and data analysis. However, it lacks precise representation and analysis of physical dynamics because it abstracts behaviours of manufacturing objects under MES level.

In this paper, a novel framework, DigTwinOps (Digital Twin framework for Operation of Cyber-Physical Production Systems) is described, which provides runtime controllability verification of a control command of a CPPS application. DigTwinOps manages the ECML-based Digital Twin Model that synchronizes the states of real machines in the production environment and provides monitoring and simulation services to both CPPS application and human worker for verifying the controllability of the decided control action.

2. System Model

2.1. CPPS Conceptual Model. Cyber-physical systems are coengineered interacting networks of physical and computational components. Figure 1 illustrates a CPS conceptual model defined by the National Institute of Standards and Technology (NIST) that explains what CPS is and how it operates between the physical and the cyber world [17]. According to the CPS conceptual model, CPS manages a series of connected control loops formed in different levels of objects, from a smart device to a single system or system-of-systems. In each control loop, the cyber system observes physical processes and controls the physical processes based on an interactive decision-making process with human actors.

Meanwhile, modern production systems have several layers that comply with control hierarchy levels (a.k.a. IEC 62264 [18]). IEC 62264 classifies the hierarchy of production systems as production environment, machine control, process control, and supervisory control. Since CPPS is a production system that adopts the CPS conceptual model, a conceptual model of CPPS can be illustrated as in Figure 2. Each level has a CPPS application that manages its own control loop that observes states of lower layers and performs interactive decision-making process with a human

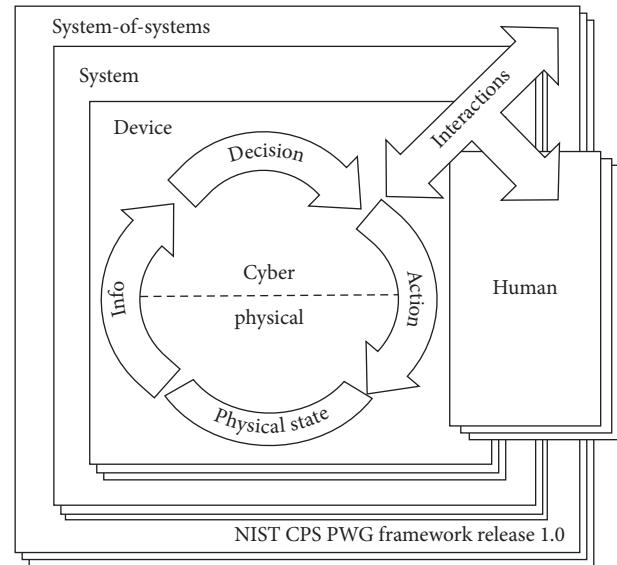


FIGURE 1: CPS conceptual model.

worker in the same hierarchical layer. Controllability verification of the result of the interactive decision-making is performed by DigTwinOps. To support controllability verification, DigTwinOps should provide two high-level functional requirements for the CPPS application and human worker. The first is monitoring of physical components to observe current states. The second is simulation capability for verifying whether the decided control action can change the state of lower-level components.

2.2. Dynamical Modelling of CPPS. All of the objects in CPPS from smart device to single system and system-of-systems are dynamic systems. Dynamical models are represented as a set of inputs, outputs, and state variables dependent upon past inputs along with the current input. Figure 3 illustrates the classification of CPPS objects into four types by governing equations and their relation to the IEC 62264-based control hierarchy level.

First of all, a continuous system (CS) operates in continuous time and its state and input/output variables are all real values. Examples are mechatronics operations in the production environment that are modelled by differential equations. Second, a discrete time system (DTS) is time varying but also periodic and its state variables are real values. Examples are machine components that have sensors and computation at the hardware level. This kind of system is modelled by difference equations. The next model is a digital system that is a computer system whose state and input/output variables are all discrete values. Digital systems (DS) such as a machine controller are modelled by a finite state machine (FSM). Most of the computer application programs in a single computing device are examples of this system. Lastly, a discrete event system (DES) is a discrete-state, continuous but event-driven system of which state evolution depends entirely on the occurrence of asynchronous discrete events over time. Distributed computing systems such as production systems and their process and

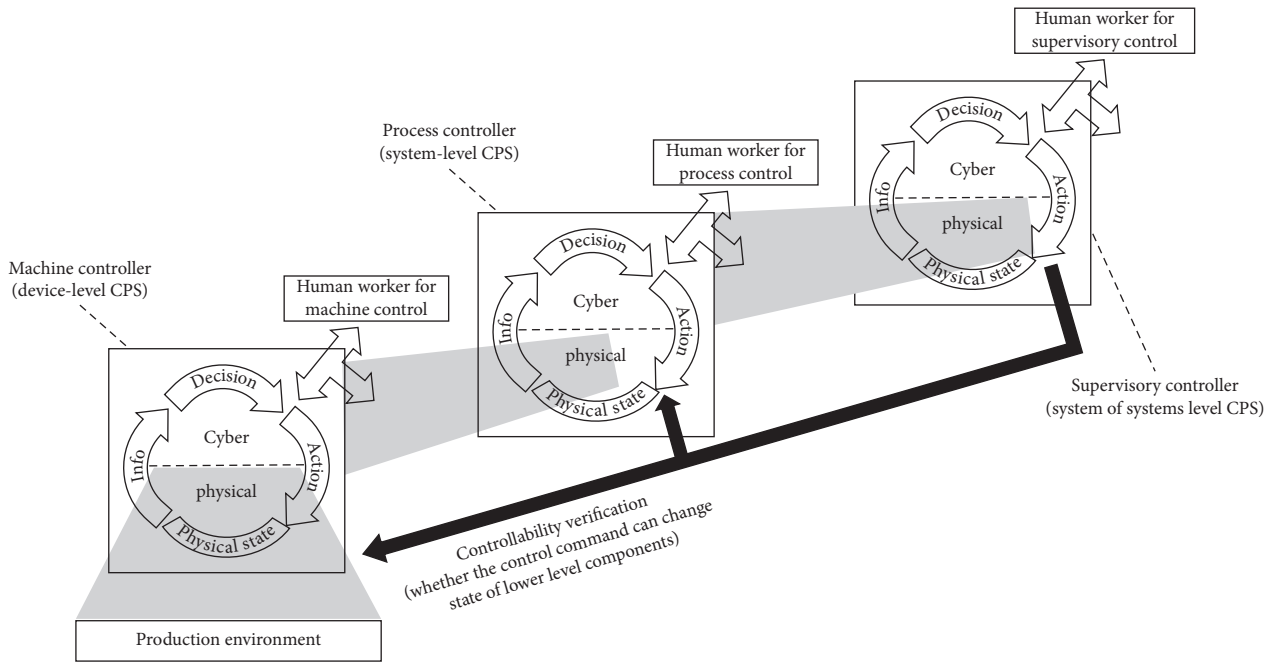


FIGURE 2: CPPS conceptual model.

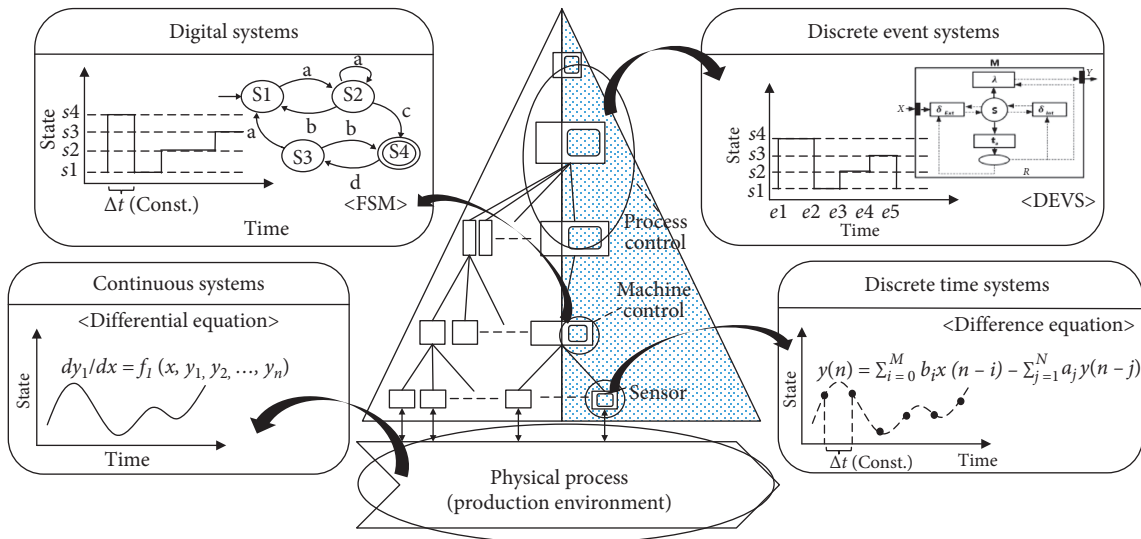


FIGURE 3: Dynamical models in cyber-physical production systems.

supervisory control logics are modelled on this level. For modelling DES, Discrete Event System Specification (DEVS) is used [15].

ETRI CPS Modelling Language (ECML) is a modelling language that supports modelling of the four dynamic systems in Figure 3 in a unified environment [19, 20]. ECML is intended to be a modular, hierarchical, and graphical language for the modelling, analysing, and simulation of systems. Components in each layer of CPPS can be described by continuous/discrete variables and continuous/discrete states with internal/external state transition rules. A target CPPS can be notated/expressed as a set of (a) CPS Structural Models (CSMs) and (b) CPS Behavioural Models (CBMs) in ECML. A CSM is composed of ports, couplings, and submodels

corresponding to CSMs and CBMs. A CBM consists of I/O ports, transitions, state variables, and constraints that update the values on its continuous properties. ECML employs the notions of conditional behaviour expressions, discrete-valued ports, continuous-valued ports, event ports, and constant properties, which would enable easier modelling of complex CPS and better performance in simulations. Figure 4 shows ECML and its representations example.

3. Design of DigTwinOps Framework

Digital twins are virtual representations of physical entities which became very popular in the manufacturing industry. With the advent of various ICT technologies, it is now

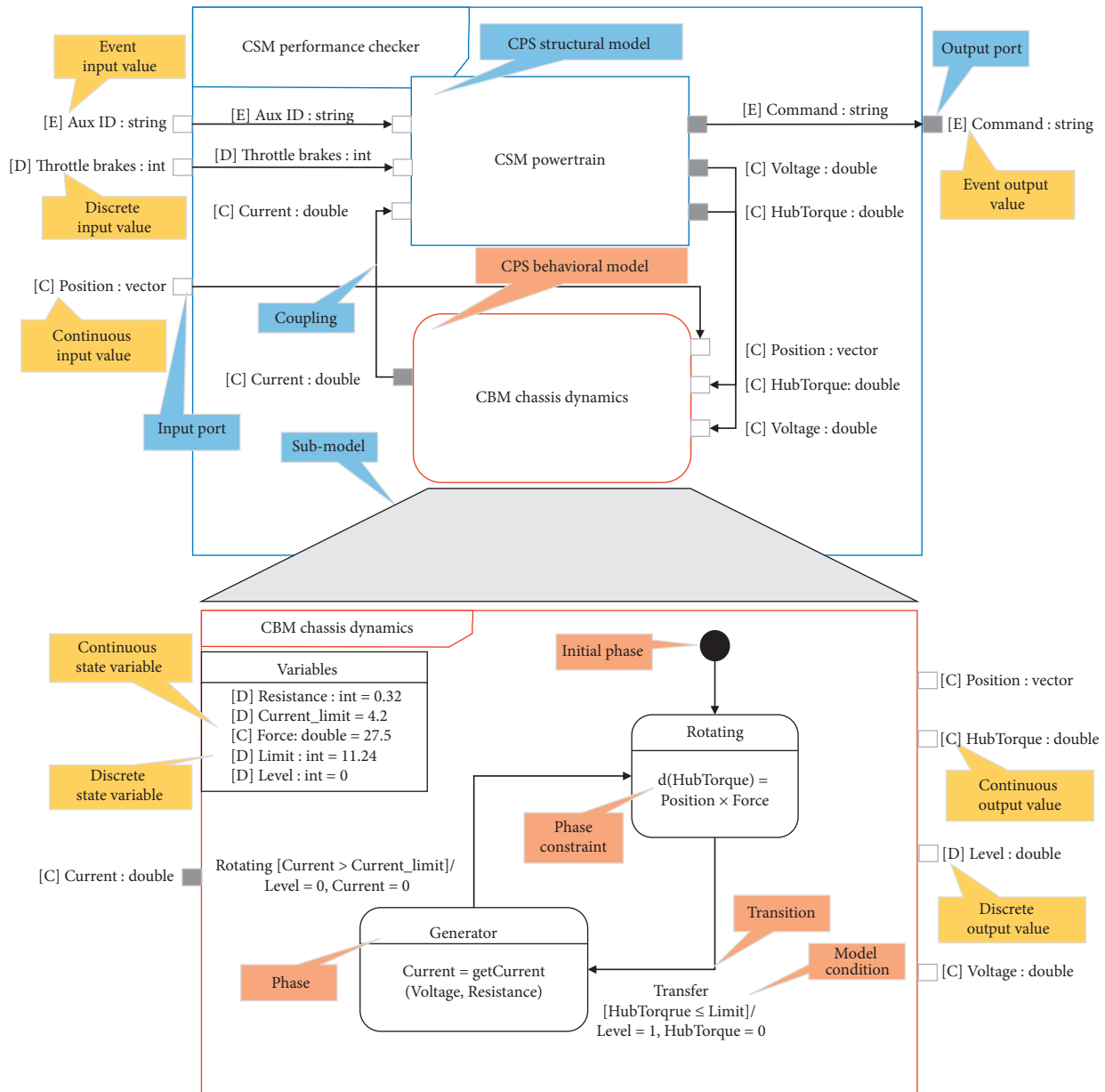


FIGURE 4: CPPS models based on ECML.

possible to enable the seamless transmission of data between the physical machinery and manufacturing execution system (MES) and to facilitate the cloud services to monitor, analyse, and optimize machines remotely [21]. In this section, we introduce DigTwinOps, a runtime controllability verification framework for CPPS. DigTwinOps features a Digital Twin Execution Engine (DTEE) which manages Digital Twin Models (DTMs), i.e., ECML models of the four dynamic CPPS objects. Each DTM synchronizes states of real CPPS objects in the production environment. Based on the management of DTMs, the DTEE provides monitoring and simulation services to CPPS applications and human workers.

The purpose of the monitoring service is state synchronization and condition checking. During this persistent

process period, DTEE collects various forms of data from the real CPPS object in the production environment and filters data using the conditions in ECML models. Figure 5 shows the monitoring service scenario of DigTwinOps framework. In this example, a 6-axis robot and a conveyor belt system are controlled by using a robot controller and conveyor controller, respectively, and a production management system carries out supervisory control for the production environment operation. Based on the four dynamical models, the robot and the conveyor system are the continuous system. Sensors in the robot and the conveyor controller are the discrete time system, while control software is a digital system. The production management system is a discrete-event system because it deals with asynchronous discrete events from distributed machinery objects and control systems. DTEE

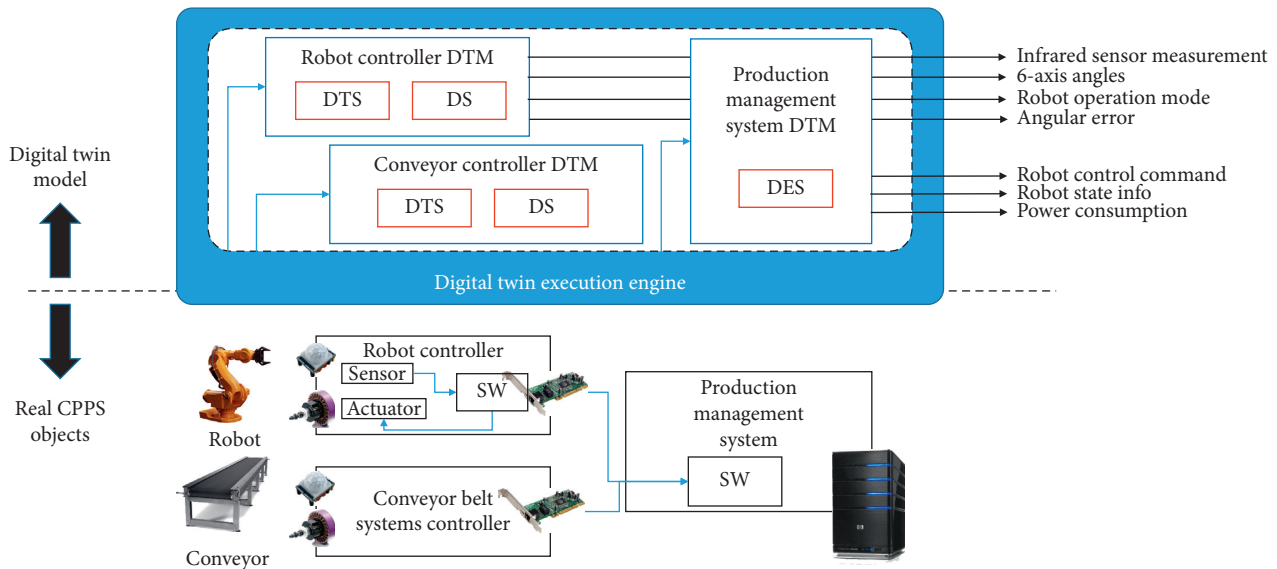


FIGURE 5: Monitoring scenario of DigTwinOps framework.

manages instance of each DTM and compares measured state value from the real CPPS object and predefined condition value in DTM. Finally, the result of condition checking is delivered to CPPS application.

The purpose of the simulation service is controllability verification. When a simulation request is arrived by CPPS application, each DTM stops synchronizing states with real objects and DTEE starts simulating DTMs. As the DTEE executes time advance functions, each DTM calculates its continuous or discrete state with time. Figure 6 shows the simulation service scenario of the previous production environment in Figure 5. At first, DTEE inputs a set of predefined control command to the production management system DTM. Then, the production management system DTM sends control signals to related DTMs such as the robot controller DTM and the conveyor belt system controller DTM. The robot DTM and the conveyor system DTM, subsequently, are moving on their relevant controller's command. When the simulation finishes, DTEE creates a simulation result, and the CPPS application and human worker can analyse the outputs of a CPPS model over time for verifying controllability of the decided control action.

Figure 7 shows a DigTwinOps framework design for the collaborative decision-making process of the CPPS application and human worker. The framework is composed of a CPPS controller and CPPS object (machine in the production environment). The CPPS controller is composed of the CPPS application and DTEE. The CPPS application processes control logic and provides visualization to human workers. The DTEE provides monitoring and simulation services to the CPPS application and human workers. The DTEE synchronizes the DTMs and its corresponding real object in the production environment. As the DTEE provides a stream of real-time state information of the target, the CPPS and human workers can monitor the status of the target. To provide monitoring service, DigTwinOps uses the MTConnect standard [22], which provides an HTTP/XML-

based data request/response mechanism between machine and application. MTConnect is composed of three components: adapter, agent, and application. The MTConnect adapter is attached to a real machine and transfers a sensor data stream to the MTConnect agent. The MTConnect agent is an HTTP server that manages collected sensor data and transfers requested data to the MTConnect application. The MTConnect application uses sensor data for various purposes in an operational environment. The DTEE is developed as the MTConnect application.

When the CPPS application requests the simulation of selected control commands with the decision-making process, the DTEE analyses the behaviours of the CPPS object when the control commands are executed in the simulation environment and sends the results of simulation data to support the verification process of the human worker. With this monitoring and simulation combination process based on the synchronization of the DTM and CPPS object, a human worker can observe states of the CPPS object and verify the effectiveness of control commands before applying it to the production environment. In the following section, the proposed framework is applied to a prototype production system.

4. Implementation of DigTwinOps-Based CPPS Prototype

DigTwinOps is applied to the prototype CPPS environment, a flexible motor assembly line. The purpose of the CPPS prototype is to produce daily production orders received from ERP and perform supervisory control of the entire CPPS. The production line is composed of a network of 14 fixed production cells (FPC) and one transfer robot (TR). Each FPC periodically senses whether a material to be operated has arrived. When the material is ready for operation, the FPC performs its own production process. When the operation finishes, the FPC calls the TR to transfer

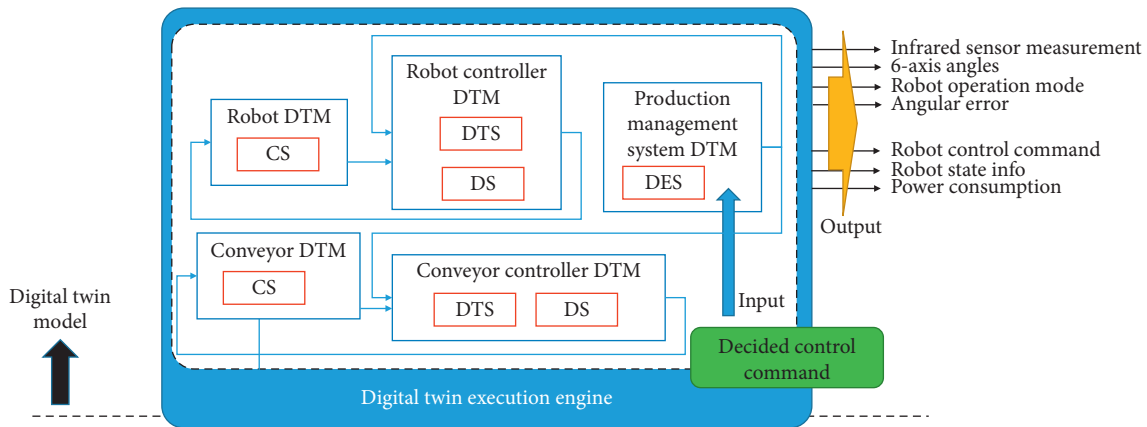


FIGURE 6: Simulation scenario of DigTwinOps framework.

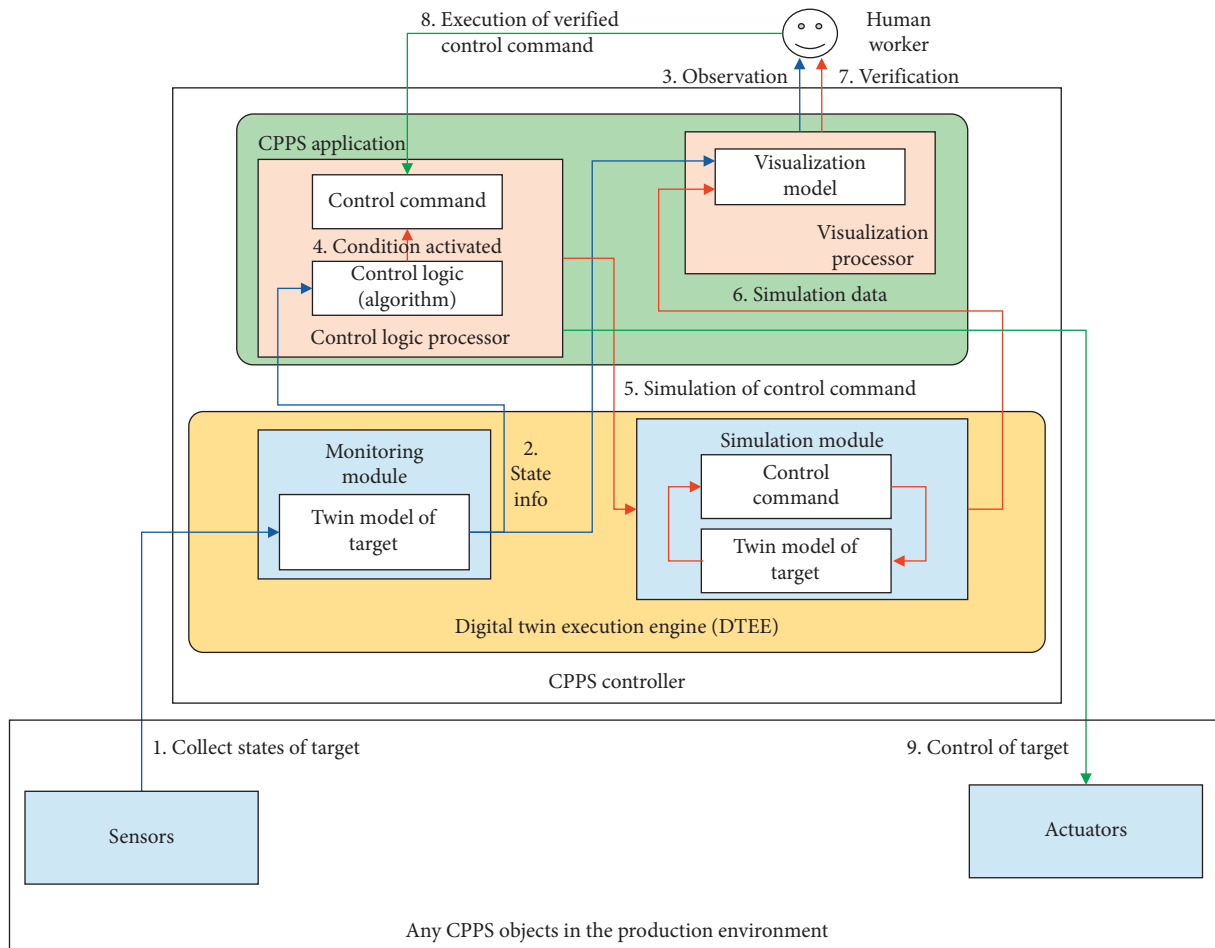


FIGURE 7: A DigTwinOps framework design.

materials to another FPC that is in charge of the next production process.

Figure 8 shows a bird-eye view of the prototype CPPS. The TR shuffles in four working positions. The prototype CPPS produces a number of motors, already ordered at the point of production that starts in the morning. Table 1 shows sample production plans and events that occurred in a day.

Production normally begins at 10 in the morning with daily production orders and inventory. Materials come in according to a daily plan, but additional orders come without notice.

Therefore, when additional orders come in, human workers should analyse the states of current production plans and decide whether the current production strategies

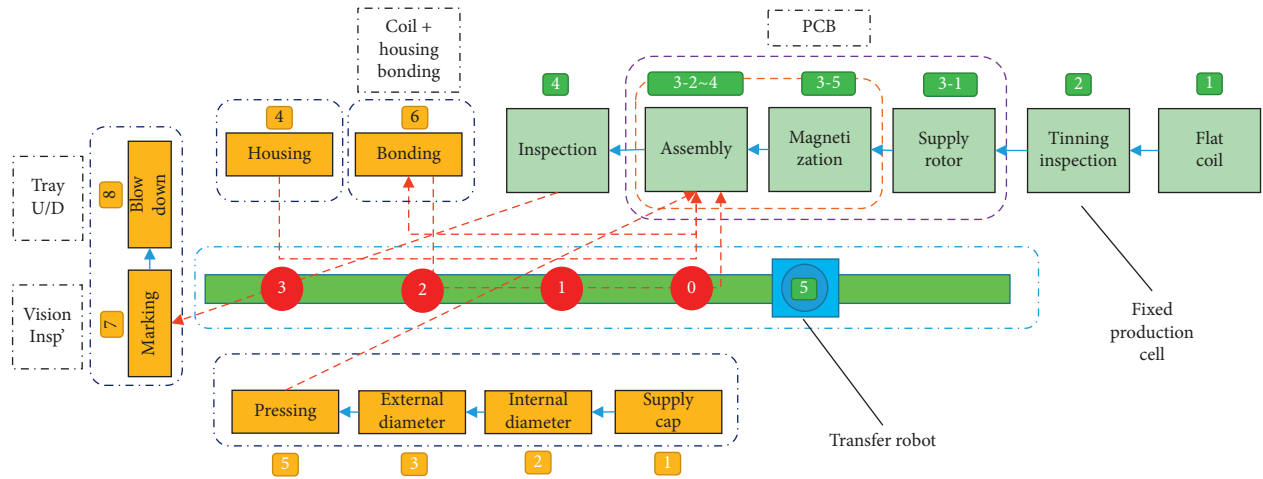


FIGURE 8: Bird-eye view of the DigTwinOps-based CPPS.

TABLE 1: Initial production plan and daily events.

Time	Production plans and events occurred
10:10:49	Production begins (# of order: 20/# of inventory: 10/# of production: 0)
12:53:01	Additional order comes (# of additional order: 4, # of total order: 24)
13:00	Additional inventory comes (# of additional order: 6)
14:30	Additional inventory comes (# of additional order: 6)
19:00	Expected production finish time

for the TR should be changed. Table 2 shows two pre-implemented production strategy models that a human worker can select between the two in the CPPS prototype.

All of the FPCs and TR are modelled by ECML, and they are synchronized with real objects in the production environment by the MTConnect standard. On the supervisory control layer of the CPPS prototype production system, the DTEE continuously observes the states of each FPC and TR by managing synchronized Digital Twin Models. When an additional order comes, the DTEE autonomously determines if decision making is required and the simulation module of the DTEE starts simulation with possible production strategy models.

Figure 9 shows Digital Twin Models of a CPPS prototype that are provided to human workers for supervisory control. It shows not only states of CPPS objects in the production environment but also the simulation process and results for human workers to compare and to verify the controllability of possible strategies. Experimental results for controllability verification are presented in the next section.

5. Experimental Results

In the CPPS testbed, the simulation service of the DTEE with two production strategies is activated when an additional order comes at 12:53:01. Figure 10 shows a graph of the expected completion time, comparing the simulation results of the original production strategy 1 to the alternative production strategy 2. The graph shows that the selection of

TABLE 2: Possible control commands.

Option	Production strategy for TR
#1	Transfer robot processes an earlier transfer request from all fixed production cells
#2	Transfer robot processes a request from nearest fixed production cell based on current working position (0~3)

production strategy 2 (20:34) will complete the production orders (24) earlier than the selection of the strategy 1 (20:52). The reason for this difference can be found by analysing the transfer route of the TR. In the case of strategy 1, the TR processes an earlier transfer request from all FPCs. In the case of strategy 2, on the other hand, the TR processes a request from the nearest FPC based on the current working position (0~3). According to Figure 8, there are three FPCs around working position 3. While the TR in strategy 1 that just finished transfer operations to one of the FPCs in working position 3 moves to the other working position (0~2) if there is an earlier transfer request, the TR in strategy 2 will answer a transfer request if it comes from one of the FPCs in working position 3. Figure 11 shows the transfer route of the TR over the daily production time.

The comparison of the simulated completion time shows that the strategy 2 has better performance for controlling the CPPS testbed. However, the earlier completion time cannot be the only reason for a better controllability metric to evaluate control commands. Therefore, we selected two performance metrics in ISO 22400 [23] for the comparison of two production strategies. The first is comprehensive energy consumption, which is the ratio between all energy consumption in a production cycle and the produced quantity (PQ). The second is production process ratio, which specifies the relationship between the actual production time (APT) over all work units and work centres involved in a production order and the whole throughput time of a production order, which is the actual order execution time (AOET).

Comprehensive energy consumption is calculated by analysing the location and travelling distance of the TR.

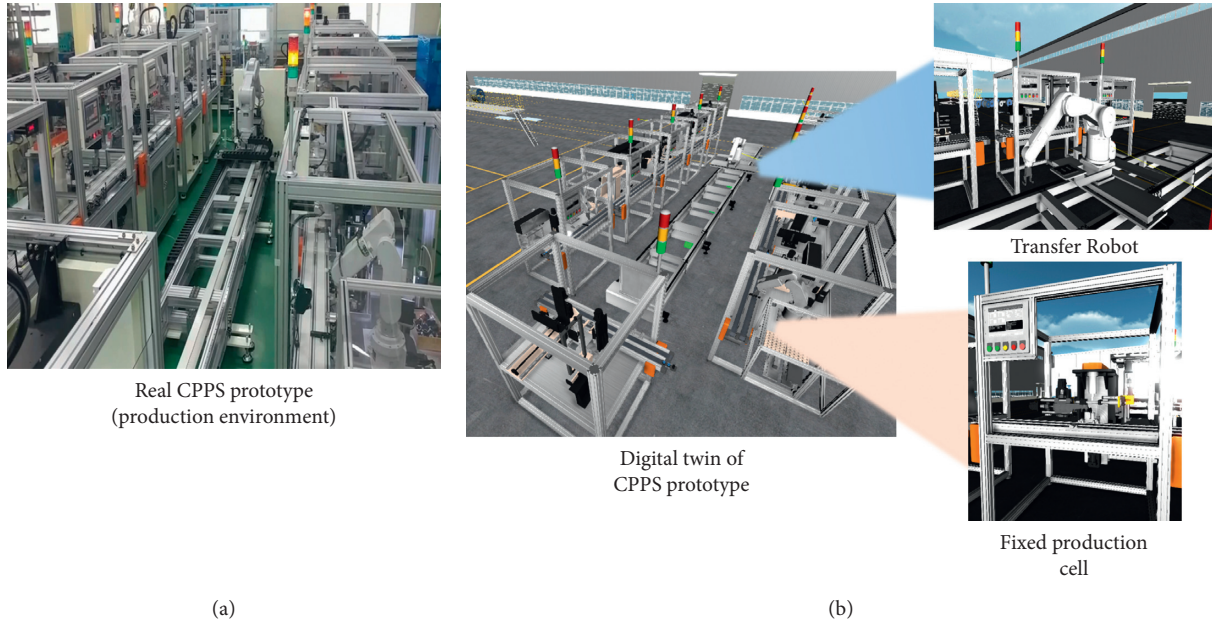


FIGURE 9: Digital Twin Model of CPPS prototype.

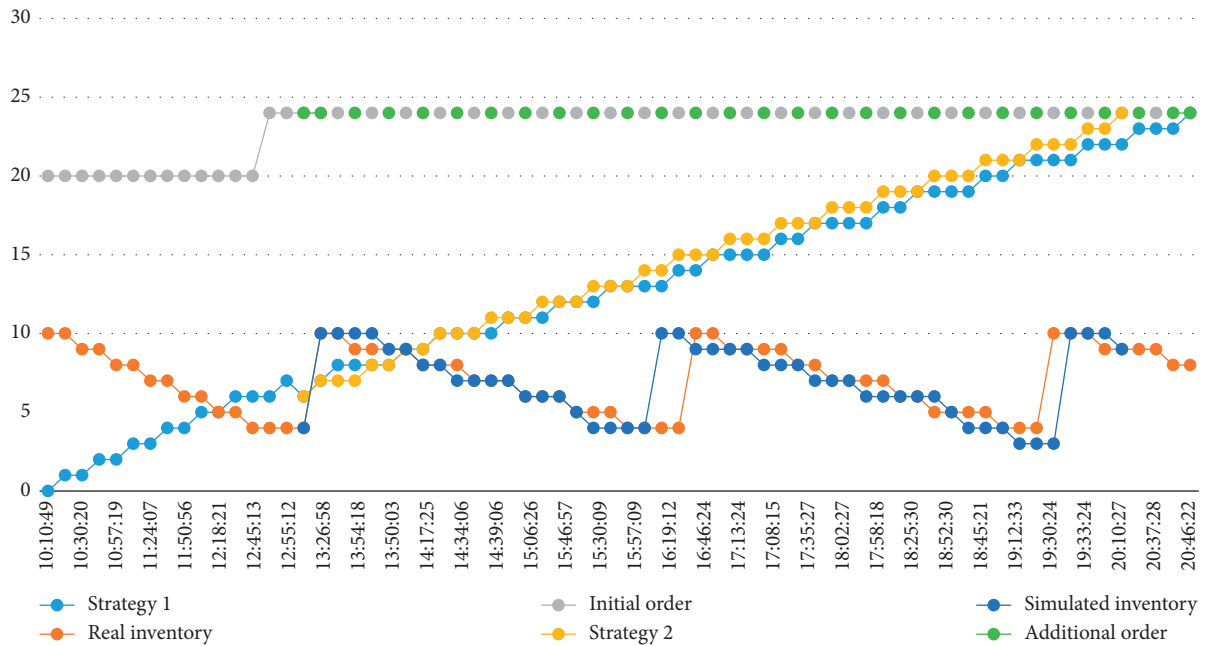


FIGURE 10: Comparison of simulation results for the completion time of production orders.

According to Table 3, the TR in strategy 1 travels 314 meters while completing 24 production orders. On the other hand, the TR in strategy 2 travels just 288 meters. This means the TR in strategy 2 consumes less energy than in strategy 1.

The production process ratio describes the efficiency of manufacturing facilities. In the CPPS prototype production system, both the initial production order time (10:10:49) and the additional production order time (12:53:01) are the same for the two production strategies. Moreover, the actual production time for one production order, which is the sum of throughput time of each FPC and mechanical operation of

the TR, is also the same. The only factor that affects the actual order execution time is the travelling time of the TR due to differences in production strategy. Therefore, the relative production process ratio can be obtained by comparing the average production time for each item's completion time. Table 4 shows the completion time per item and the average production time for the two strategies.

The simulation results show that it takes 27 min and 10 seconds (1630 seconds) to produce one item when strategy 1 is selected, while it takes only 26 min and 12 seconds (1572 seconds) when strategy 2 is selected as a control command.

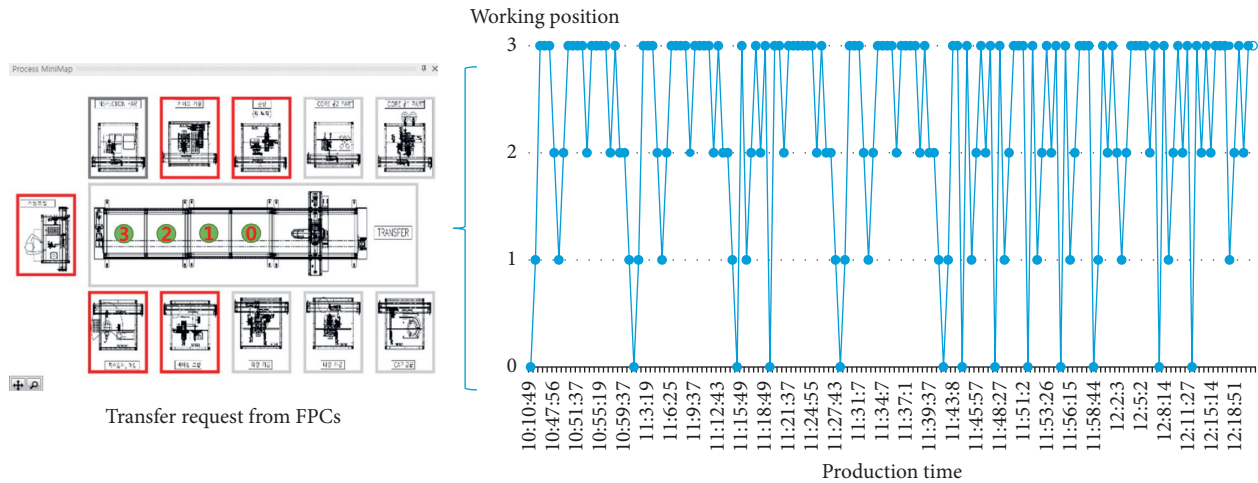


FIGURE 11: The transfer route of the TR over production time.

TABLE 3: Travelling distance and comprehensive energy consumption of the two strategies.

	Strategy 1	Strategy 2
Travelling distance	314 meters	288 meters
Comprehensive energy consumption	100	91.7

TABLE 4: Comparison of production process ratio.

Item	Completion time (S1)	Production process ratio (S1)	Completion time (S2)	Production process ratio (S2)
7	13:10:01	163%	13:26:58	62%
8	13:37:13	118%	13:47:03	85%
9	14:06:55	110%	14:15:13	91%
10	14:34:17	107%	14:42:30	93%
11	15:01:37	102%	15:04:04	98%
12	15:28:37	96%	15:22:09	104%
13	15:55:37	100%	15:54:57	100%
14	16:22:49	98%	16:19:12	102%
15	16:49:49	98%	16:44:12	102%
16	17:14:49	99%	17:13:24	101%
17	17:41:49	98%	17:35:27	102%
18	18:09:01	98%	18:02:27	102%
19	18:36:01	97%	18:25:30	103%
20	19:03:13	95%	18:42:21	106%
21	19:30:13	95%	19:10:21	105%
22	19:57:25	94%	19:32:36	106%
23	20:24:37	94%	19:58:24	106%
24	20:51:49	96%(overall)	20:34:27	104%(overall)
	Avg. production time of the strategy 1 27 m 10 s		Avg. production time of the strategy 2 26 m 12 s	

This means production strategy 2 has a comparative advantage in terms of production process ratio by 104%; in other words, it produced the same amount in 4% less time when strategy 2 is selected.

The experiment shows that the CPPS application and human workers are capable of observing the real-time states of the production environment and activating simulation when a predefined issue occurs (additional order requested). By comparing the simulation results for optional control commands (production strategy for the TR) based on the

criteria (completion time, production process ratio, and comprehensive energy consumption), the CPPS application and human workers can derive better control commands that help the CPPS promote more of the state, i.e., controllability verification.

6. Conclusions

Current smart factory research is only at the level of partial application of IT technology in production or operational

environments. Therefore, research on how to build a new factory into a smart factory or how to convert an existing factory into a smart factory is insufficient. In this paper, the hierarchy of the existing production system is modelled as digital twin, and the framework of DigTwinOps, which uses it to perform monitoring and simulation, is proposed. This framework allows interworking simulations of data from existing factory hierarchies and can be reflected in decision making based on the simulation results of possible control commands.

Usually, manufacturing companies are headquartered in the city, and factories are located on the outskirts. This framework will be located near the factory on the outskirts, which will enable rapid data collection and quick decision-making at the site through direct connections to the facility. To do so, however, the facilities will need to be equipped with 5G and other wireless technologies, as well as a data center near the factory where high-performance servers capable of processing and simulating data should be deployed. This is also the shape of the smart factory as part of the fourth industrial revolution. From a security perspective, since the proposed framework will be located in a private network close to the factory and provide abstracted data (e.g., simulated data, production output, and energy consumption) to the manager in the headquarter, there may be security issues in the data exchange between the headquarter and the site. We are considering a structure that stores sensitive data used by the proposed framework in the demilitarized zone (or DMZ that is a physical or logical subnet that separates an internal local area network (LAN) from other untrusted networks) and allows headquarters to access the DMZ only through VPN (virtual private network).

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

A part of tools, prototypes, and methodologies of this research are provided by the ICT R&D program of MSIP/IITP (IoT-based CPS platform technology for the integration of virtual-real manufacturing facility). This work was supported by research fund of Chungnam National University.

References

- [1] S. Kang, J. Jeon, H. S. Kim, and I. Chun, "CPS-based fault-tolerance method for smart factories," *At-Automatisierungstechnik*, vol. 64, no. 9, pp. 750–757, 2016.
- [2] D. Zuehlke, "Smartfactory—towards a factory-of-things," *Annual Reviews in Control*, vol. 34, no. 1, pp. 129–138, 2010.
- [3] J. H. Lee, S. D. Noh, H. Kim, and Y. Kang, "Implementation of cyber-physical production systems for quality prediction and operation control in metal casting," *Sensors*, vol. 18, no. 5, p. 1428, 2018.
- [4] D. Ding, Q.-L. Han, Z. Wang, and X. Ge, "A survey on model-based distributed control and filtering for industrial cyber-physical systems," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 5, pp. 2483–2499, 2019.
- [5] S. Spinelli, A. Cataldo, G. Pallucca, and A. Brusaferrri, "A distributed control architecture for a reconfigurable manufacturing plant," in *Proceedings of the IEEE Industrial Cyber-Physical Systems (ICPS)*, pp. 673–678, Saint Petersburg, Russia, May 2018.
- [6] M. László, "Cyber-physical production systems: roots, expectations and R&D challenges," *Procedia CIRP*, vol. 17, pp. 9–13, 2014.
- [7] E. A. Lee, "Cyber-physical systems: design challenges," in *Proceedings of the IEEE International Symposium on Object Oriented Real-Time Distributed Computing (ISORC)*, pp. 363–369, Orlando, FL, USA, May 2008.
- [8] P. F. S. de Melo and E. P. Godoy, "Controller interface for industry 4.0 based on RAMI 4.0 and OPC UA," in *Proceedings of the IEEE Workshop on Metrology for Industry 4.0 and IoT*, pp. 229–234, Naples, Italy, June 2019.
- [9] S. Fuchs, A. Gercikow, and H. P. Schmidt, "Monitoring of real-time behaviour of industrial ethernet for industry 4.0," in *Proceedings of the IEEE International Electrical Engineering Congress*, Pattaya, Thailand, March 2017.
- [10] B. Galloway and G. P. Hancke, "Introduction to industrial control networks," *IEEE Communications Surveys and Tutorials*, vol. 15, no. 2, pp. 860–880, 2012.
- [11] F. Alaieri and A. Vellino, "Ethical decision making in robots: autonomy, trust and responsibility," in *Proceedings of the International Conference on Social Robotics (ICSR)*, pp. 159–168, Kansas City, MO, USA, November 2016.
- [12] R. Rajkumar, D. de Niz, and M. Klein, *Cyber-Physical Systems: Chapter 5. Software and Platform Issues in Feedback Control Systems*, Addison-Wesley Professional, Boston, MA, USA, 2017.
- [13] E. A. Lee and S. A. Seshia, *Introduction to Embedded Systems, A Cyber-Physical Systems Approach*, MIT Press, Cambridge, MA, USA, 2017.
- [14] A. Rajhans, A. Bhave, I. Ruchkin et al., "Supporting heterogeneity in cyber-physical systems architectures," *IEEE Transactions on Automatic Control*, vol. 59, no. 12, 2014.
- [15] K. Lee, J. Hong, and T. Kim, "System of systems approach to formal modeling of CPS for simulation-based analysis," *ETRI Journal*, vol. 37, no. 1, pp. 175–185, 2015.
- [16] S. C. Yoon, *Factory-in-the-loop control approach to smart factory operation efficiency*, Ph.D. thesis, POSTECH, Pohang, Republic of Korea, 2017.
- [17] E. R. Griffor, C. Greer, D. A. Wollman, and M. J. Burns, *Framework for Cyber-Physical Systems: Volume 1, Overview*, National Institute of Standards and Technology, Gaithersburg, MD, USA, 2017.
- [18] International Electrotechnical Commission, *IEC 62264: Enterprise-control System Integration-Part 3: Activity Models of Manufacturing Operations Management*, International Electrotechnical Commission, Geneva, Switzerland, 2016.
- [19] J. Jeon, I. Chun, and W. Kim, "Metamodel-based CPS modeling tool," in *Embedded and Multimedia Computing Technology and Service*, vol. 181, pp. 285–291, Springer, Lecture Notes in Electrical Engineering, Springer, 2012.
- [20] H. Y. Lee, I. Chun, and W. Kim, "DEV&DESS-based cyber physical systems modeling language with uncertainty consideration," in *Proceedings of the Spring Simulation Multiconference*, San Diego, CA, USA, April 2013.

- [21] A. E. Saddik, "Digital twins: the convergence of multimedia technologies," *IEEE MultiMedia*, vol. 25, no. 2, pp. 87–92, 2018.
- [22] ANSI, *MTCConnect Standard Version 1.4.0 (ANSI/MTC1.4-2018)*, ANSI, New York, NY, USA, 2018, <https://www.mtconnect.org/standard20181>.
- [23] International Standard Organization, *ISO 22400: Automation Systems and Integration—Key Performance Indicators (KPIs) for Manufacturing Operations Management—Part 2: Definitions and Descriptions*, International Standard Organization, Geneva, Switzerland, 2014.

Copyright © 2019 Sungjoo Kang et al. This is an open access article distributed under the Creative Commons Attribution License (the “License”), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited. Notwithstanding the ProQuest Terms and Conditions, you may use this content in accordance with the terms of the License. <http://creativecommons.org/licenses/by/4.0/>