Synergy of multi-modelling for process control

Richárd Beregi[a,b*], Ádám Szaller[a,b], Botond Kádár[a]

[a] *Centre of Excellence in Production Informatics and Control*
Hungarian Academy of Sciences Institue for Computer Science and Control
Kende str. 13-17, H-1111 Budapest, Hungary (e-mail: richard.beregi@sztaki.mta.hu).
[b] *Budapest University of Technology and Economics, Department of Manufacturing Science and Engineering*
Műegyetem rkp. 3., H-1111 Budapest, Hungary

Abstract: In this paper a multi-modelling experiment is presented through which we have studied the possibilities of manufacturing process control supported by different digital simulation models. The main pillar of the study is a real, operating, research and demonstration cyber-physical production system which is detailed in the study. Our digital twin of the system in question includes two different virtual models; an agent-based model endowed with the ability of error handling, and a discrete-event simulation-based model for forecasting and supporting the error handling routine with evaluating bids. The experiment includes typical manufacturing processes with machine failures, which should be detected and recognized to invoke both simulations for re-forecast and error management.

Keywords: *digital twin, forecasting, multimethod modelling, discrete event-based simulation, agent-based simulation, error handling*

---

1. INTRODUCTION

One of the main challenges in manufacturing today is to design and operate systems producing a high variety of customized products as efficiently and quickly as possible, while dealing with uncertain and highly volatile demands. Managing manufacturing companies and systems requires both long-term and short-term decisions, which all deeply influence the performance of these firms [6].

The existence of validated, easily updatable and parametrizable models are one of the most important requirements in handling problems occurring in the operation of a manufacturing system and in having effective decision support on both planning and execution levels. From the modelling point of view, system's models may use different formalisms and approaches, depending on the characteristics of the considered problem and the expected results. Whether the system is a production line, a distribution network or a communication system, modelling can be used for:

- gaining knowledge about the system in different life-cycle phases
- evaluating certain features in the system
- predicting system performance
- comparing different alternatives
- detecting system problems
- evaluating and improving system performance.

As a modelling option, analytical models can be adopted which use mathematical or symbolic relationships to provide a formal description of the system [3]. The model is then used in order to derive an explicit expression of a performance measure or, in most of the cases, to define an algorithm or a computation procedure able to calculate the addressed performance indicators.

Applying simulation technology is another option to analyse and execute performance evaluation of production systems. Three major methodologies are known to build simulation models: discrete-event modelling, agent-based modelling and system dynamics. Simulation models represent the events occurring in a manufacturing system in its operation by a sequence of steps that are executed in a computer program [8]. This time lined sequence is generated with respect to a set of rules modelling the behaviour of the system. Accordingly, the characteristics and relationships between the elements in a production system can be described in detail. However, the higher the detail level is, the higher the required computational effort. If a simulation model is run for a sufficiently long time, then proper statistics can be collected, and performance indicators can be estimated.

Concerning the planning, one of the main drawbacks of today's production planning and control systems is that the decision makers rely on results achieved with static models that ignore important operating constraints/objectives of live shop operations. It is due to the lack of a close correspondence with the live status of executed processes and the data resulting from their real-time monitoring.

Another weakness comes from the fact that building effective, usable and valid models often results in a capital-intensive activity, even while adopting commercial software platforms. Nevertheless, the models are applied only once or very few times. These models are named “throw away” or “stand-alone” models because they are seldom used after the initial plans or designs have been finalized. An extensive study of the penetration and use of discrete event simulation in the UK manufacturing industry identified only 11% of sites out of a sample of 431 which were currently utilizing simulation as a decision support tool [13]. One of the limitations of its use for on-line decision-making is the considerable amount of time spent in gathering and analysing operational data. Consequently, this can result in decision-making processes relying on simulation primarily for off-line decision support and not for critical on-line decision-making. In real-time control, the three key issues are data acquisition, quick response and instantaneous feedback.

Usually, the traditional data update process in the simulation model-based decision making is carried out manually and if the control logic of the production system is changed, manual core changes are required in the simulation model (e.g. usually a new model is necessary).

As stated in [12] Cyber-Physical Production Systems (CPPS), relying on the latest and foreseeable further developments of computer science (CS), information and communication technologies (ICT), and manufacturing science and technology (MST) may lead to the 4th Industrial Revolution, frequently noted as Industry 4.0. Having the newest ICT enabling technologies, the real-time connection and update of the data and status of the real manufacturing system into its simulation model(s), thus achieving the exact mirror of the real system and its controlling part in the planning environment is currently possible. Such solutions, named usually as Digital Twin can provide a drastic change in the lifecycle of the decision making since it is expected that the model will be continuously used in parallel with the real manufacturing system, supporting the managers and the engineers to optimize their manufacturing processes, to react effectively in the case of disturbances and to discover potential future undesired situations in a proactive manner.

In the optimization and investigation of industrial production lines, digital twin will play a decisive role in the close future. It was proposed by Grieves in 2003 at University of Michigan, and defined as three parts: physical product in the physical space, virtual product in the virtual space, and the real-time two-way connection between them [4]. The virtual side isn’t just recording performances of the physical one, but also carries out optimization and prediction based on the stored historical data [14].

The evolution of smart interconnection and interoperability between virtual and physical space has four stages. At the first stage, the production depends on the physical manufacturing line, due to the lack of effective information, which may lead to low efficiency and accuracy. Then, at the second stage, with the appearance of different information technologies, interaction is created between the virtual and the physical space – but the interaction is weak, and the virtual model is not the real-time representation of the physical one. Now, at the third stage, owing to the increased usage of sensors, the reduction of calculation times, new communication technologies and the interaction exists. In the future (fourth stage), the two-way connection will be enhanced and additional services will be available [14].

In [14] the authors present a novel concept of digital twin shop floor. They divide the digital twin into four key components: physical shop floor, virtual shop floor, shop floor service system and shop floor digital twin data. According to this concept, while creating the interaction between the two spaces the following aspects should be considered and implemented [14]:

1. Data collection and order transmission at the field level
2. Data processing methods
3. Information systems to optimize the production process
4. Creation of a virtual environment
5. Establish interaction
6. Information security protocols

The most important part of the interaction is to update the status of the virtual space according to reality. Creating a real digital twin solves this issue: it fills the gap between the two spaces.

The authors in [1] present a digital twin architecture reference model for cloud-based cyber-physical systems (CPS). They describe the digital twin as a part of a CPS. According to the authors, the digital twin could be used for diagnostics, monitoring, and prognostics purposes and with the cloud infrastructure it becomes a bridge between the application and physical level of CPS. It is important because the physical layer can provide real-time information, and the cyber layer can extend that with delay tolerant applications. Digital twins could analyse the current state of the system, and recommend actions for the best outcome.

Zhang et. al. [16] describe a digital twin-based approach for designing and decoupling a hollow glass production line. They divide the iterative designing and decoupling process into three major steps:

1. Rapid individualized design based on reference models
2. Distributed semi-physical simulation
3. Decoupling of multi-objective optimization in design

However, digital twins are not only useful in connection of production lines. As for a different applicability, they could help reducing item return rates in the clothing retail trade [5], in aircraft real-time monitoring [15] or even faults can be diagnosed and problems solved before a trouble happens with a digital twin of a car.

In this paper, two of the previously mentioned methodologies are implemented in parallel for the examination of the same system: a discrete event-based simulation (DES) model in Tecnomatix Plant Simulation and a multimethod model that contains a distributed agent-based and also a discrete event-based layer in AnyLogic.

There is an essential difference between the two modelling techniques. In DES models the state variables of the system can change at separate points in time - when an event occurs. Events may change the system state, create or delete other events. The DES technique uses a dynamically changing event list to describe the behaviour of a system and does not deal with the time between the events.

In general, the agent-based simulation focuses on the actions and interactions of autonomous agents. An agent can be a product, a machine, or even a sensor with a defined behaviour which is in accordance with the cyber-physical systems concept [10]. The overall performance of the system is formed by the individual behaviours and interactions of the participants with themselves and with their environment [11]. This concept presents the incomplete knowledge-based decision-making factor, which is lifelike with economic competition between the “costumer” and the manufacturer and among the agents [9].

2. SYSTEM DESCRIPTION

The SmartFactory is a cyber-physical sample production system which is created for modelling and testing industrial issues. The system is capable of running complex simulation and scheduling experiments, meanwhile executing abstract manufacturing and logistical tasks. The system can be described as an experimental environment for Industry 4.0 and cyber-physical system related researches, while these concepts could also be demonstrated for industrial partners and the general public. Our last specified intention is to use the system for educational purposes - students can obtain experience on industrial equipment with real constraints. [7]

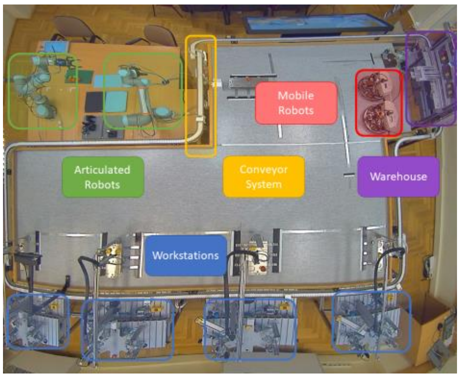

Figure 1. SmartFactory

In the SmartFactory, each and every workpiece is equipped with a unique identification tag for tracking and recognition purposes. This feature allows the workpieces to represent industrial components on an abstraction level. Workflows mostly start with every workpiece stored in the warehouse. After they are unloaded and transported to their processing destination, they are manufactured on one of the four identical workstations. Then they are transported back to the storage. The parts of the system are described below. [7]

2.1. Workstation

Four, perfectly identical Festo Modular Production Systems (MPS) can be found in the manufacturing cell. All of them have a stepper motor-propelled 6-positioned turntable as their central moving unit. The first nest of the dedicated positions is used for the loading and unloading the workpieces with a manipulator. At the second position, a pneumatic drill probe checks whether the workpiece has a pilot hole. After the tester there is a stamper tool, which prints a pattern (depending on the stamp) on the dedicated surface. The fourth position, which is the most easily accessible for humans, provides a possibility for manual manipulation of the workpieces. This position also has a button for the operator to signal after the task is finished. This is followed by a drilling position, which consists of a drill which can be lowered onto the workpiece for an adjustable time. At this nest the workstation performs material removal machining on the stamped surface. If there is no such surface, the drill sinks into the pilot hole. The sixth and final nest contains a pushing mechanism, which shoves the faulty workpieces onto a slide where it leaves the production system. [2]

2.2. Warehouse

The Festo Didactic corporation's uniquely manufactured high-bay warehouse has three levels and twelve palette places, and is an integral part of the SmartFactory system. Eleven of the twelve palette places are capable of storing four workpieces, while the remaining place is used to switch the palettes' positions. The palettes are moved by a two-pronged lifter, which is driven vertically by a numerically controlled (NC) servo motor propelled cogged belt, and moves on a pneumatic rail in the horizontal direction. The lifter is halted by pneumatic bumpers in order to stop at the right palette place. The system executes the vertical and horizontal positioning simultaneously. [2]

2.3. Transportation systems

In the demonstration system, the transportation between the warehouse and the different workstations can be realized by a conveyor system or by mobile robots. The conveyor system is composed of four FlexLink X45 type conveyor belt with individual motor drive, because the length and bending constrains. The warehouse and every workstation has a bypass unit for unloading workpieces from the conveyors. The mobile robot system is made of two omniwheel-driven automated guided vehicles (Festo Didactic Robotino). They determine their position on the table using their stepper motors' encoders and a gyroscope. The robots are also equipped with two inductive and an optical sensor, with which they achieve sensor fusion in order to more accurately position themselves at the buffer stations, and they also have a gripper for the transportation of the workpieces to the system components.

3. ANYLOGIC MODEL

The AnyLogic environment is a Java-based simulation tool for multi-modelling tasks. Owing to the well-known high-level programming language, it can be easily connected to the physical system with a TCP/IP based communication protocol over the dispatching unit described in the 5th section.

The AnyLogic model of the SmartFactory system is two-layered: one discrete event-based model for the realistic operation as in the physical system (Figure 2) and an agent-based order and job management layer, where each and every component of the system is endowed with the ability to make decisions for itself based on their interest. This abstract layer's main purpose is to manage the work organization without full knowledge of the capabilities and statuses of the components. The so-called "Management" agent is responsible for the control of the bidding procedure for each error handling job between the "Resource" agents and the message handling between the physical system. Each "Resource" agent has its own DES representation and capabilities for performing different manufacturing processes.

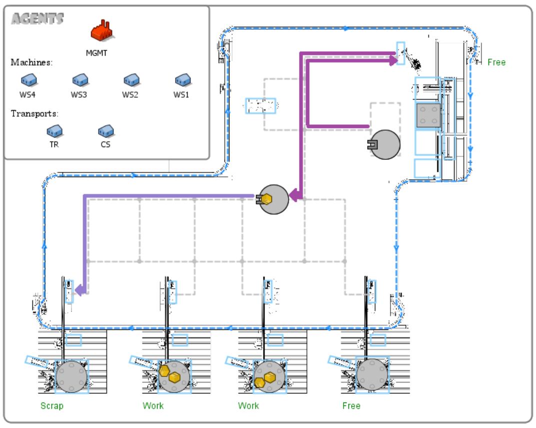

Figure 2. Runtime screen of the AnyLogic model

Following the Digital Twin concept, the simulation is linked with the real system and can react to errors without any user interactions. The simulation runs together with the real system by continuously mapping the process statuses to the DES layer. When an error occurs, the agent-based layer gains control over the DES layer and also over the SmartFactory. The application of the agent-based model implemented in AnyLogic is described in the 5th section.

4. PLANT SIMULATION MODEL

Plant Simulation is a discrete event-based simulation software, which is capable of simulating different manufacturing and logistic processes in an object-oriented way. While AnyLogic is developed for simulating processes in general, Plant Simulation is specialized for creating models about production systems – the toolkit (pre-defined objects with various setup options and attributes) makes the model building process easier.

The Plant Simulation model of the SmartFactory is depicted in Figure 3. The model is capable of importing the tasks and assigning the processes that have to be performed to the given products, and with going through all of the processes in the routing, forecasting the average expected lead times connected to each workstation. In the model, having an abstraction level, the mobile robots are modelled by human workers which are basic building blocks provided by the modelling environment, while the turntables are symbolized by separate workstations and short routes between them. The difficulty with creating the Plant Simulation model was the fact that the system must not reach a dead lock – which may happen when two workpieces try to reach the same loading position at the turntable from the conveyor belt and from last processing turntable nest. Since the simulation software is programmable, this deadlock is avoided by using different methods and "if-then" structures within the programming environment.

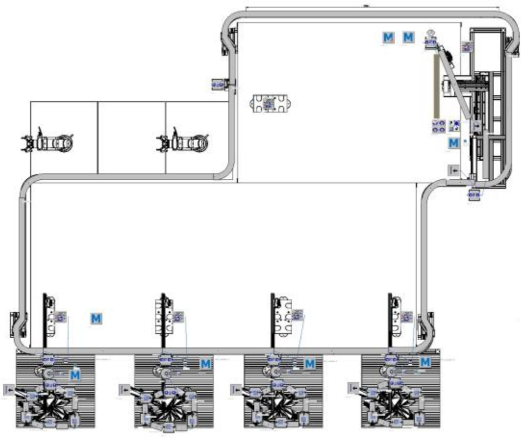

Figure 3. Runtime screen of the Plant Simulation model

The main aims of the discrete event-based model implemented in Plant Simulation is highlighted in more details in the next section.

5. SYNERGY OF MULTI-MODELLING

Our high-level model comprises six components which are used in the experiments (Figure 4). As mentioned before, the main pillar is the SmartFactory cyber-physical production system with all of the low-level controllers, actuators and sensors. The dispatching unit realizes a communication hub with a standardized protocol for status reporting and process controlling statements. Every low-level controller is reachable in a unique way from the hub (e.g. command interpreter statements over UDP protocol on LAN for every PLC of the workstations, JSON structured string message protocol on CAN for the microcontrollers of the bypass units), but for the other components of the system, they are accessible over a unified JSON based message structure over TCP/IP. This way the SmartFactory system's low-level commands are mapped to a more common higher-level form with a header containing a universally unique identifier, the target of the message and the current status of the task.

The controller functions as the driving force of the production in the SmartFactory: manages the processes. The routing of every workpiece is generated in a precedence-based graph form and is executed in this component based on the status reports. These reports have the same structure as the commands previously described but the status value of the task differs. We can identify acknowledgment from the controller as the actual beginning of the process. The status can have a value which means the task is finished without any problem or it can have an error value. Every process related information which reaches the dispatching unit is stored in a database in a structured form for later evaluation.

Both two simulation models are connected to the dispatching unit but receive different information. The routing of a certain workpiece is described in a task graph form, where the nodes are different operations that must be performed on a certain workpiece and the edges define the precedence constrains. The multimethod model is provided with the status reports to map the current state of the demonstration system. When an error occurs in the AnyLogic DES layer, it is also registered as a status report of the system, which means it reaches the controller and the error handling routine is activated.

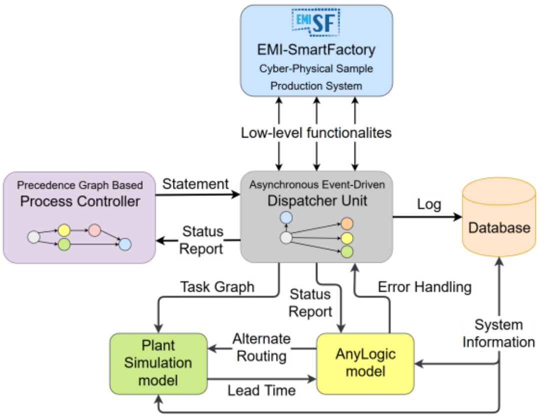

Figure 4. System functional structure

As shown in the flowchart in Figure 5, after the generation of the routing the production process starts with the controller's overview, while the Plant Simulation model obtains the task graph in a matrix form (the software is limited to table-based data inputs). The conversion is made by an intermediate script and transmitted to the simulation model through a socket interface. As already mentioned, the Plant Simulation model is capable of making forecasts about the expected average lead times based on the received task graph. The AnyLogic model runs together with the physical environment based on the information gained from the dispatching unit over another socket interface.

5.1. Error handling

When an error occurs (symbolized with a red "Error" node in Figure 5) the controller stops working, because the precedence constrains deny reaching the next step. Technically one edge (constrain) must be removed and an alternate branch, which substitutes the failed process, has to be inserted. Although the controller is not yet capable of accomplishing this functionality, so the multimethod model provides this new branch.

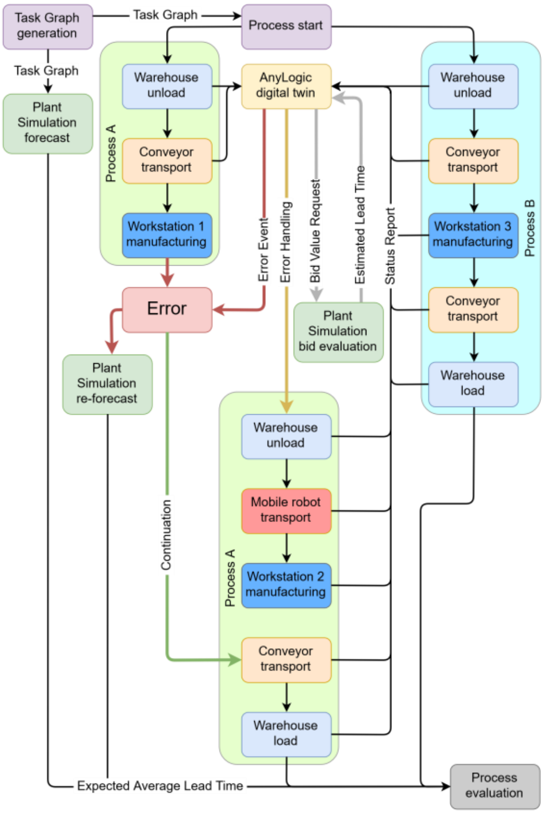

Figure 5. Process flowchart

In practice, if a workstation makes scrap based on the predefined probability in the DES layer, the agent layer is alerted. Thereafter the "Management" agent takes charge and generates a new order for a product based on the type of the failed job meanwhile removes the scrap workpiece from the workflow with the pushing mechanism at the workstation. The "Management" agent announces the work for the "Resource" agents and they can apply to take part in the tender. Every "Resource" replies for every job offer based on their own capabilities and they either drop it or bid to obtain the task. This bidding is supported with test simulation runs performed by the Plant Simulation model. When it receives the alternate routing possibilities from each "Resource" agent, an experiment is run in Plant Simulation for each bidding agent, and returns the forecasted lead times, which are the actual bid values. The decision-maker "Management" agent chooses the bidder with the lowest lead time offer. After the winner is published, in the case of an error, the transportation to the victor is done by the mobile robots (while in the other cases the conveyor system transports the parts). When the error is solved, the agent-based layer gives back the control to the SmartFactory process controller. This way the obstacles of the continuation are removed, and the controller can continue working.

In terms of the Plant Simulation model, when the production continues after the new routing created with the error handling branch, the simulation needs to restart itself from the beginning in the possession of the new task graph. It would be complicated to save the model state right before the error occurs, and continue the simulation run form that moment with a different task matrix – it is easier to run the simulation again (and it does not take much time and computational effort). This way the Plant Simulation model is a dual-purpose component of the system: its tasks are forecasting and bid supporting.

The main difference between the application of the multimethod modelling and the discrete-event approach is the following: when an error occurs, the Anylogic model uses agents that compete for executing the previously failed, unfinished task, while the Plant Simulation model forecasts show the diversion from the planned course.

6. FUTURE WORK

The error-handling method which is presented here is the first step of our research project in this area and shows a lot of potential, but it still needs further improvements. The outlined flowchart in the 5th section was executed with participation of every component and operated as expected and previously described. On one hand, the real-life error detection, which would be the base for industrial implementations, can not be achieved without the installation of proper sensor network for every possible failure. The software-based approaches (e.g. timeout, missing response from controller) are indirect indications of an arising error. Since the SmartFactory demonstration system is limited to a lower level of load and since the desired process complexity appropriate for the capabilities of the Plant Simulation model is higher, the second most crucial hardware development is to prepare the system for a long-time operation with a higher load of jobs.

7. CONCLUSION AND OUTLOOK

The experiment we proposed showed that the integration of multiple simulations can improve a manufacturing system's behaviour in relation with error handling and forecasting. The short-term distributed control can solve an arising problem without re-planning everything, while the invokable DES can predict the difference between the planned and the changed production parameters. We would like to underline that as we just started our research, the work presented in this paper is only the first step we completed. Nevertheless, we are continuing our activities to elaborate more complex scenarios and we are porting our multi-model approach to real industrial environments.

8. ACKNOWLEDGEMENT

This research has been supported by the GINOP-2.3.2-15-2016-00002 grant on an "Industry 4.0 research and innovation center of excellence".

REFERENCES

[1] Alam, K.M.; El Saddik, A. (2017): C2PS. A Digital Twin Architecture Reference Model for the Cloud-Based Cyber-Physical Systems. In IEEE Access 5, pp. 2050–2062. DOI: 10.1109/ACCESS.2017.2657006.

[2] Beregi, R. (2015): Design of a prototype production control system to the Smart Factory. In Master's thesis, Department of Manufacturing Science and Engineering, Budapest University of Technology and Economics.

[3] Colledani, M.; Tolio, T. (2005): A Decomposition Method to Support the Configuration / Reconfiguration of Production Systems. In CIRP Annals 54 (1), pp. 441–444. DOI: 10.1016/S0007-8506(07)60140-1.

[4] Grieves, M. (2014). Digital Twin: Manufacturing Excellence Through Virtual Factory Replication. [Online]. Available: http://www.apriso.com

[5] IET: E&T Magazine, Volume 11, Issue 04, May 2016 - Page 44-45.

[6] Kádár, B.; Terkaj, W.; Sacco, M. (2013): Semantic Virtual Factory supporting interoperable modelling and evaluation of production systems. In CIRP Annals 62 (1), pp. 443–446. DOI: 10.1016/j.cirp.2013.03.045.

[7] Kemény, Zs.; Beregi, R.J.; Erdős, G.; Nacsa, J. (2016): The MTA SZTAKI Smart Factory. Platform for Research and Project-oriented Skill Development in Higher Education. In Procedia CIRP 54, pp. 53–58. DOI: 10.1016/j.procir.2016.05.060.

[8] Law, A.; Kelton, D. (2000): Simulation modeling and analysis. McGraw-Hill, New York.

[9] Márkus, A.; Kis, T.; Váncza, J.; Monostori, L. (1996): A Market Approach to Holonic Manufacturing. In CIRP Annals 45 (1), pp. 433–436. DOI: 10.1016/S0007-8506(07)63096-0.

[10] Monostori, L.; Váncza, J.; Kumara, S.R.T. (2006): Agent-Based Systems for Manufacturing. In CIRP Annals 55 (2), pp. 697–720. DOI: 10.1016/j.cirp.2006.10.004.

[11] Monostori, L.; Valckenaers, P.; Dolgui, A.; Panetto, H.; Brdys, M.; Csáji, B.Cs. (2015): Cooperative control in production and logistics. In Annual Reviews in Control 39, pp. 12–29. DOI: 10.1016/j.arcontrol.2015.03.001.

[12] Monostori, L.; Kádár, B.; Bauernhansl, T.; Kondoh, S.; Kumara, S.; Reinhart, G. et al. (2016): Cyber-physical systems in manufacturing. In CIRP Annals 65 (2), pp. 621–641. DOI: 10.1016/j.cirp.2016.06.005.

[13] Ryan, J.; Heavey, C. (2006): Process modeling for simulation. In Computers in Industry 57 (5), pp. 437–450. DOI: 10.1016/j.compind.2006.02.002.

[14] Tao, F.; Zhang, M. (2017): Digital Twin Shop-Floor. A New Shop-Floor Paradigm Towards Smart Manufacturing. In IEEE Access 5, pp. 20418–20427. DOI: 10.1109/ACCESS.2017.2756069.

[15] Tuegel, E.J.; Ingraffea, A.R.; Eason, T.G.; Spottswood, S.M. (2011): Reengineering Aircraft Structural Life Prediction Using a Digital Twin. In International Journal of Aerospace Engineering 2011 (3), pp. 1–14. DOI: 10.1155/2011/154798.

[16] Zhang, H.; Liu, Q.; Chen, X.; Zhang, D.; Leng, J. (2017): A digital twin-based approach for designing and decoupling of hollow glass production line. In IEEE Access, p. 1. DOI: 10.1109/ACCESS.2017.2766453.