

Received 27 November 2022, accepted 20 December 2022, date of publication 23 December 2022,
date of current version 29 December 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3232061

APPLIED RESEARCH

IIoT Device Prototype Design Using State Machine According to OPC UA

PETER STRELEC^{ID}, TIBOR HORAK^{ID}, SZABOLCS KOVAC^{ID},
EDUARD NEMLAHA^{ID}, AND PAVOL TANUSKA^{ID}

Faculty of Materials Science and Technology in Trnava, Institute of Applied Informatics, Automation and Mechatronics, Slovak University of Technology in Bratislava, 91724 Trnava, Slovakia

Corresponding author: Tibor Horak (tibor.horak@stuba.sk)

This work was supported in part by the Scientific Grant Agency of the Ministry of Education, Science, Research, and Sport of the Slovak Republic and the Slovak Academy of Sciences, Proactive Control of Hybrid Production Systems Using Simulation-Based Digital Twin, under Grant VEGA 1/0176/22; and in part by the Proposal of Identification and Monitoring of Production Equipment Parameters for the Need for Predictive Maintenance in Accordance with the Concept of Industry 4.0 Using the Industrial IoT Technologies under Grant VEGA 1/0193/22.

ABSTRACT This article proposes a digital twin design using a finite state machine (FSM) that can be integrated and used for the simulation requirements of a production process. The FSM used to introduce the proposed concept can create a fully fledged digital twin. The design of the state machine ensures the functionality of the control systems, parts of the production line, and the associated Internet of Things (IoT) devices. The concept can be extended using an interface that can communicate at the level of the required protocol, thereby ensuring the possibility of integration into the production line. The digital twin can be used as a model and basic platform to develop an IoT hardware device. The advantage of this system is that its implementation can be verified with or without a physical prototype. It is possible to test hybrid implementation, where hardware elements are partially integrated into the software model created based on FSM. The design of the IoT device was verified on a production line, where it was adaptively controlled the delivery rate of material for sorting. It introduces an opportunity to test or simulate a possible planned change in production infrastructure, in which it is possible to better understand the solution and verify the expected changes and their effect on production. The design can be fully used in accordance with Industry 4.0.

INDEX TERMS Finite state machine, digital twin, the IoT device, simulation, production line.

I. INTRODUCTION

The Industry 4.0 framework transforms production from standalone automated units into a fully integrated automated and continuously optimized production environment [1].

This will allow companies to create “smart products” that are clearly identifiable and localizable knowing not only their history and current status, but also understanding the alternative methods that could create the final product. Vertical production processes are horizontally interconnected

The associate editor coordinating the review of this manuscript and approving it for publication was Jesus Felez^{ID}.

within industrial systems that respond flexibly in real time to immediate and changing demand for products [2].

These vertical production processes will be able to respond to individual customer requirements and based on these requirements, will also be able to produce products efficiently.

The production process is continuously optimized and can respond to unexpected changes caused by the failure of a production facility [3]. The starting points of the fourth industrial revolution arise from the implementation of new models and approaches to human work activities using the Internet of Things (IoT).

This is related to new socio-economic behavior, new modes of thinking, and changes in human society. However, these changes also require necessary steps in terms of technological preparation using the latest cybernetics and other modern technologies and methods [4].

This has created new global networks based on the interconnection of production facilities in the form of cyber-physical production systems (CPPSs). CPPSs are becoming an essential building block of “smart industries” which are characterized by the ability to exchange information autonomously, to be able to trigger the necessary actions in response to current conditions and to ensure mutual, independent control.

The sensors, machines, and Information Technology (IT) systems involved in CPPSs are interconnected within a value chain that transcends the boundaries of individual companies. CPS (Cyber-Physical System) connected in this manner can interact and analyze data using standard IoT-based communication protocols that allow them to foresee possible errors or faults and configure and adapt themselves to changing conditions in real time [5].

CPPSs, which change the field of production management and present new possibilities for integrating entities in the production process, make it possible to change the flexibility of the production process as well as to incorporate new production elements and define their mutual substitutability and communication requirements [6].

In the Industry 4.0 concepts, processes are optimized even before they are implemented in practice, thus enabling the verification of parameter changes in the simulation environment and the preparation of suitable parameters for the production process [7].

Simulation and digital twins are used in the industry to achieve set production goals, improve integration, and reduce risks when changing production process parameters. Specific production decisions are the parameters of a planned operation that guarantee the achievement of the production goal and can be characterized as job scheduling problems [8].

These parameters were verified using a digital twin. This approach uses a set of attributes and properties of existing physical devices. This approach records and updates data throughout the life cycle of the system. The identification and processing of data for the preparation of the digital twins during the integration of the IoT device into the production line and the data synchronization among the digital twins make it possible to describe the behavior of the existing system [9].

By integrating the digital twin concept with model-based systems engineering (MBSE), hardware solutions can be developed based on the digital twin software model [10].

The basic characteristic of a device developed in this manner is a set of autonomous elements that can communicate with each other. Simultaneously, these elements should be able to adapt to the incoming changes in production and should be able to record changes in the environment, as well as to respond to them autonomously [11], [12].

All of these elements lead to the best possible management processes. Performance indicators are the main parameters for usability and quality of management. Every change in production represents a risk that needs to be eliminated, and owing to the variability of solutions, this task is not always easy. Significant efforts have also been devoted to meeting these requirements through machine learning [13].

To achieve these goals, and especially to achieve the flexibility represented in Industry 4.0, it is necessary to pursue the agile implementation of changes required using the production process. A comprehensively solved task can slow down the implementation of an eventual solution. Faster adaptation and acceleration of development can be achieved using an agile approach, which can help in the development of software and hardware devices [14].

To achieve this goal, it is necessary to classify and identify areas that can be prepared in parallel so that individual teams can focus on their assigned tasks. The technical side of this represents an area where it is possible to use a digital twin and the CPPS system so that the individual subsystems of the solution can be solved separately.

A. RELATED WORKS

The concept of a finite state machine (FSM) is relatively complex, and its deployment presents a challenge. When creating an FSM model, it is necessary to conduct an accuracy test for the model. Testing and issues relating to the deployment, synchronization, and identification of states, as well as their verification, fault detection, and the evaluation of testing, were described in the work of Bongjoo Koo et al. in 2020 [15].

The design and implementation of a digital twin model with the possibility of managing a production system dealing with the development of the model and a systematic framework for developing Digital Twin (DT) based CPPS have been proposed previously [12]. In this study, the possibility of integrating and developing a digital twin as a possible development prototype was verified.

The communication itself was processed through protocols related to a specific situation that occurs in asynchronous communication. The solution to this problem was inspired by the work of Muttersbach et al. [16], in which synchronous modules were equipped asynchronous wrappers that adapted their interfaces according to the required timing.

The preparation of the integration of the state machine model was based on the principles of CPPS, for which the configuration, automatic and integration elements, and subsystems within the entire level of production are described in the work of L. Monostor et al. [17].

The integration of IoT devices, applications of this approach in Industry 4.0, the issue of safety communication, and its integration into the production process are summarized in the work of Kim [18].

The versatility of the proposed solution is based on the concept of Industry 4.0, and it is possible to choose a suitable communication protocol as needed.

B. CONTRIBUTIONS

IIoT systems are an integral part of every modern company in Industry 4.0. IIoT is a term for modern devices that can be controlled remotely via the internet [19].

The Internet of Things (IIoT) refers to technologies that enable cheap wireless connections and the communication of various sensors and devices to automate, speed up, and streamline processes; measure distances; act as remote controls; increase comfort; and enable better quality of life, with many other uses in areas such as agriculture, medical care, recycling of waste products, virtual real estate production, and movement in the gaming industry [20].

The design of an IIoT device and its physical or software implementation can be divided into subtasks. The basic element for ensuring such an implementation approach is the development and implementation of an IIoT device, which is described in this article.

The idea is to create a model that provides better transparency and simplifies the design of the IIoT devices. The design of the model and methods used were inspired by the OPC UA, which is the Industry 4.0 standard. The project framework is the realization of a digital twin as a finite-state machine (FSM) [21].

The FSM was used in the OPC Unified Architecture (OPC UA) to describe the internal functionality of the device. This design of the model appears advantageous, and it has been verified that it is possible to evaluate the properties by which production can be influenced by monitoring the parameters in the production process. It is also possible to monitor the degenerative conditions during production [22].

Each state defined in the FSM models also presents the possibility of solving a specific problem through an independent team, thus allowing the incorporation of software and hardware solutions into the final product. The advantage is the parallel implementation of several modules and the testing of the partial solution using the created model of the FSM [23].

The model can also be used for simulation or integration into the production process in a prototype design, where part of the tasks can be performed only by software or using a hybrid approach. In such an approach, the individual parts of the proposed solution can be combined with the hardware. This procedure is enabled by the application of input and output variables that affect the assignment of states to an FSM [7], [24].

II. MATERIALS AND METHODS

According to the Industry 4.0, it is possible to design an IIoT solution or Programmable Logic Controller (PLC) program in the form of a finite-state machine [12].

The proposed solution represents the principles of the device. This device can facilitate integration into the production process and allow the device to be tested and simulated without the need to intervene or shut down the production process. In the traditional understanding of Industry 4.0, the

digital twin is presented as a faithful copy of an existing production process or technical device that serves as a real existing model [25].

When designing a simulation or IIoT device, it is often necessary to test or provide a solution as quickly as possible in relation to the actual product during development. After exploring these possibilities, we focused on the idea of preparing a digital twin as a prototype of the proposed solution and enabling the preparation of the intended extensions, which would accelerate the development, testing, and deployment of real equipment without the need for prior interventions in the production process.

To design this solution, it is advantageous to use the industry standard OPC UA, which ensures the compatibility and variability of the proposed equipment for use in the industry. One level of the OPC UA standard involves a description of the function of the device as a final-state machine [26].

Programmed devices go through a prescribed sequence of states during their execution. This is typically used to describe the function of a device, and it provides a description of the behavior of the device. According to the OPC UA standard, FSM elements describe the phases of the program's execution in terms of valid transitions between a set of states, as well as describing the stimuli or causes of these transitions and the resulting effects of the transitions [27].

Therefore, in this article, we examined the possibility of creating an IIoT device design according to the OPC UA methodology. First, a model is created on which the possibility of agile management of device development is tested. The result of this process is an IIoT device integrated into the production line. FSM was tested at all developmental stages.

According to our assumptions, the development process set up in this manner should facilitate the development of IIoT devices. Simultaneously, the FSM model must guarantee the parallel implementation of individual states. Each condition can be implemented as required by an additional FSM.

The defined design system of the final IIoT device should satisfy the attributes of Industry 4.0, in which the creation of its digital twin is ensured, and the software component can also be used for physical testing or verification of functionality without the creation of a physical device that can be validated in practice.

Verification was carried out on the design of the device to improve the sorting quality of the material in the production line. The model created can exist as a software digital twin, where parts that are entered parametrically for simulation purposes can be replaced with sensors, physical or logical interfaces that supplement data from the production line.

In this way, a prototype IIoT device can be deployed and verified that has been designed as a model using FSM and the OPC UA interface has been used for solution compatibility.

A. THE PRODUCTION LINE

Manufacturing with IoT elements in industry is one of the most important steps on the road to Industry 4.0. Integrating the production environment with IoT devices significantly increases the efficiency and quality of production and minimizes unnecessary costs [28].

As shown in Figure 1., we use a real production line for industrial material sorting. Specifically, this part of the production line is used for the output quality control of materials, and its task is to separate defective materials from those that meet the prescribed requirements.



FIGURE 1. Real production line.

This system can be used for sorting various materials, such as in the production of screws or in the food industry, sorting corn, wheat, and other materials. It is beneficial for this part of the production line to sort the defective material from acceptable material as efficiently as possible.

The individual components of the production line are connected by a switch and controlled by a Siemens type S7-300 PLC controller. The production line contains Human Machine Interface (HMI) panels, through which the operator controls the production process.

The operator can determine the speed of the conveyor belt using these HMI panels [29].

In our case study, the production line involved a sorting device that evaluated the color of the sorted product using a Red, Green, Blue (RGB) sensor (see below). The entire production process is recorded and archived using a server.

This server is equipped with a manufacturing execution system (MES) built on the Wonderware platform. The Wonderware system enables efficient control and management and enables the archiving of production operations.

The Wonderware platform is equipped with the Historian subsystem, which enables efficient storage and archiving of all production processes, on the basis of which it is possible to analyze this data to adjust and streamline the production process [30].

B. INDUSTRIAL IoT

The IoT improves manufacturing operations by improving connectivity, device management, production monitoring, and customer relationships. The IoT connects devices between each other and to people and collects and shares large amounts of data [31].

Other main benefits are

- **The monitoring of production.** Because IoT connects intelligent machines and collects and shares large amounts of data, it is possible to monitor production in real time. This allows an immediate response to production interruptions, helps eliminate downtime, and reduces the process inventory. The planned production can be compared with the actual production in real time. The speed of the machine and line can be changed in real time, and the amount of process stock can also be adjusted. IoT allows the production to be completed on time and allows synchronization with both process stocks and raw material stocks [32].

- **The management of a remote device.** Connecting machines compatible with the IoT to the network allows remote device management. The worker can manage or monitor devices from anywhere and is not limited by having to be directly in front of a device that requires maintenance. For a better diagnosis of this problem, it is advisable to equip the device with intelligent sensors. It is possible to set up protocols to actively manage the device, thus saving energy costs and reducing the overall operating costs [33].

- **The maintenance of the devices.** Condition-based maintenance alerts can easily be implemented with the IoT. The IoT is a key factor in performing maintenance based on its reliability and the use of machine learning to support predictive maintenance. This is positively reflected in increased production volumes, reduced downtime, lower maintenance costs, high machine reliability, and a higher return on investment in the form of better machinery use and production volume [34].

- **Continuous improvement through data analysis.** Six sigma and other continuous improvement concepts require several data points to function properly. From this viewpoint, industrial IoT is important. The IoT helps aggregate product data and gather them for analysis. Subsequently, people working in continuous improvement can identify problems, work out the root causes, implement improvements, and verify that these improvements are working [35].

III. RESULTS

The proposed solution was implemented on a production line, with the main goal of increasing the quality of the sorting module. The production line had a sorting device that evaluated the color of the sorted product based on an RGB sensor; in the case of an unwanted particle, which was represented by gray, this particle was removed.

To improve the process and adaptation of the sorting of several materials, it is necessary to ensure the adaptation of the particle feed rate to the sorting line input. This

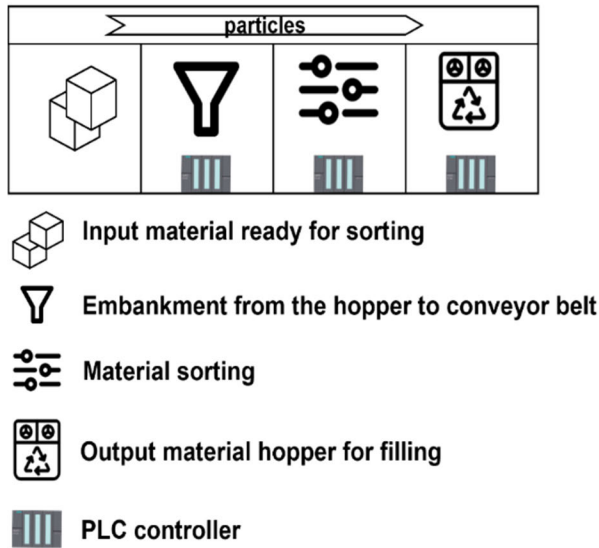


FIGURE 2. Sorting module of the production line.

requires the design of an IoT device that is capable of performing this task. For testing, it was necessary to verify the proposed solution as simply as possible. Because the production line was overworked, it is impossible to intervene in the production process for the purpose for streamlining the production process.

Therefore, it was necessary to design a digital twin model for this part of the production line, which dealt with material sorting, as shown in Figure 2.

This part of the production line, which was used for sorting, was extended using the prepared IoT device, which, based on the sorted material, sent a signal to change the feed speed of the conveyor belt, which fed the material to the sorter input. This parameter also changes in practice and is sufficient to increase the quality of sorting. The quality classification of the sorting process was determined by assessing the sorted samples at the outlet of the sorter based on the collected waste.

If the sorting was not of the required quality and the waste also contained particles that were evaluated as satisfactory or, conversely, if the sorted output also contained particles that did not meet the quality requirements, it was necessary to manually change the feed speed rate. However, changing the feeding speed rate does not always sufficiently resolve this issue. To better demonstrate the efficiency of material sorting, the digital twin of an existing device was first realized.

Production process control was implemented using the Manufacturing Execution System (MES) of the Wonderware system. The data were collected using the Historian system, and the Message Queue Telemetry Transport (MQTT) protocol was used, which enabled the data to be read. It was possible to use the data from the real environment and apply them to the digital twin model.

As shown in Figure 3. These data can be used as live data. This type of data can also be used in the Historian subsystem

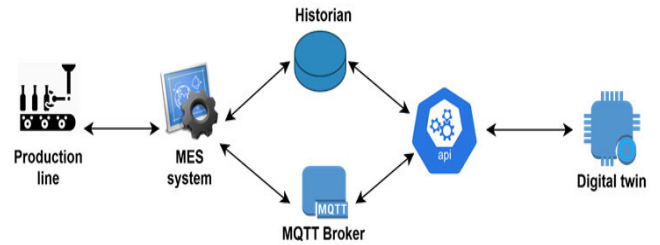


FIGURE 3. Schematic integration of the FSM digital twin model.

and then delivered to the MQTT broker, from which the data are drawn by the proposed digital twin.

The structure of the data blocks named Tags, which are provided via MQTT or a Historian, is shown in Table 1. Tag is defined by System Integrators. It represents a set of attributes, keys and values with a specified type of the data. It's used for recording and displaying the status of a machine, sensors or equipments. The basic definition is represented by TagKey, which is a unique identifier and TagName and additional values, which are defining limits, the expected values, quality and raw values. There is a possibility of having a parameter for the sampling interval for retrieving data.

The MQTT broker was configured with subscriber isolation settings and VerneMQ software was used for this purpose. For each Tag, it was necessary to define the subject under which the data were transmitted to the MQTT broker [36]. The MES system that provided the integration of the sorting modules of the production line was used to collect the data. Each sector was controlled by a PLC system and the following Tags were collected:

- Monitoring the filling status of the input bin for sorting
- Start signal for material shift.
- Material feed speed, where the reading frequency 2Hz
- Feed rate of the conveyor belt.
- Stop of the feed process
- Start signal for sorting
- Number of sorting hits
- Stop signal for sorting
- The frequency of extraction of sorted material.

This process was derived for the RGB sensor and a signal was generated for each sorting action to record the sorting activity of the material. The change in conveyor belt feed speed setting was manually adjusted according to the experience of the operator. The measurement of sorting quality was carried out on the test samples. The material sorting quality testing set was set up in 90% correct samples to 10% samples to be sorted.

The frequency of image capture depends on the belt speed offset. Distances are visible on Figure 4. It is also possible to use video sequential processing, which showed a lower accuracy rate.

The calculation of the displacement speed and frame rate was calculated as follows:

TABLE 1. Structure of the data block.

Column name	Description	Type
TagKey	Unique numerical identifier for the tag	String
TagName	The tag name is a unique name of the tag	String String
Description	Description	
Datetime	Datetime	Datetime
Value	Current value of read data	Float, Integer Float, Integer
Value	Computer value	Datetime
Statetime	Time zone	
Minraw	The minimum value of the raw acquired value. Analog tags	Integer
Maxraw	The maximum value of the raw value acquired. Analog tags	Integer
Mineu	The minimum value of the tag	Measured in engineering units
Maxeu	The maximum value of the tag	Measured in engineering units
Unit	Unit	String
Quality	Quality of data	Integer
Qualitydetail	Data quality is the degree of validity for a data value	Integer
Qualitystring	QualityDetail bit layout	Integer
Opcquality	OPC quality	String
Wwretrieval mode	Read data setup	String
Sourcetag	Source system tag	String
Sourceserver	Source server	String
Wwcyclecount	The number of cycles for which data were retrieved	Integer
Wwresolution	The sampling interval for retrieving data	Integer
Percentgood	Stored with good quality	Integer
Startdatetime	Start date of reading interval	Timestamp

Length of the area for taking picture:

$$L_t = L - (i - 1) * L_s \tag{3}$$

Speed of feed is:

$$v = \frac{i * \frac{L + L_t}{2}}{t_n} \tag{4}$$

From the known speed, we can determine the time t_n

$$t_n = \frac{i * \frac{L + L_t}{2}}{v} \tag{5}$$

D - diameter

L - axial distance

i - number of frames in the row

L_t - length of the scanned area

L_a - It is the distance of the i-th member of the decreasing arithmetic series

t_n - time required to move the area to the next frame

v_{max} - maximum allowed speed of feed

Figure 5 shows how the IoT device senses the sorting quality at the output. Where a case of passing mis-sorted material and its identification is shown.

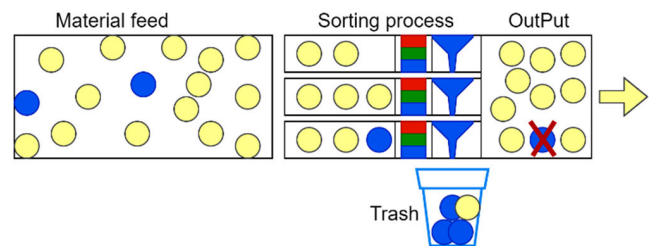


FIGURE 5. Sorting of materials.

Images that contained multiple poorly sorted elements were counted as a floating average. This is used to calculate the feed rate control of the belt. All numbers that express the rate of displacement take only positive values. To control the velocity, it is necessary to vary the nature of the change smoothly and for this purpose, the geometric mean of the evaluated values is used.

The geometric mean x_g is defined as the n-th root of the product of n numbers $x_1, x_2, x_3, \dots, x_n$.

$$\bar{x}_g = \sqrt[n]{x_1 * x_2 * x_3 \dots x_n} = \left(\prod_{i=1}^n x_i \right)^{\frac{1}{n}} \tag{6}$$

Figure 6 shows a comparison of the mixed samples as the feed control was performed by the Operator OPS and by the added IoT device - AI. The range of speed control is given in percentage.

The IoT device, is configurable and the post-rotation speed is increased by a value of 5% up to the detection of mis-sorted material. At that point, a reduction in speed occurs. At each detection of a mis-sorted material, the sorting process is evaluated and the feed rate is changed. From the specification, the belt speed has been defined.

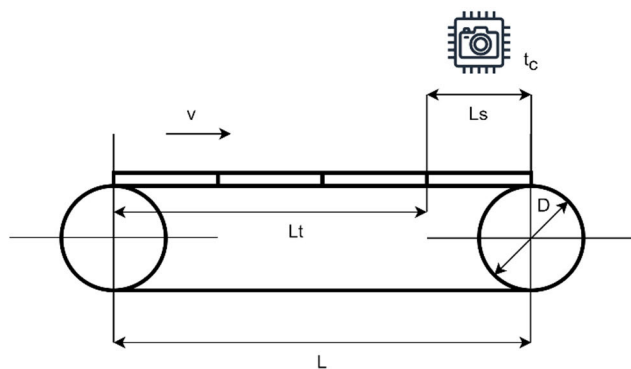


FIGURE 4. Scheme for calculating the scan rate of the sorted material window.

Drive member speed:

$$n = \frac{v_{max}}{\pi * D} \tag{1}$$

from the circumferential velocity per second, it is possible to determine the speed:

$$v = \pi * D * n \tag{2}$$

A. BASIC FSM MODEL

The digital twin is implemented in Python. A finite state machine (FSM) was used to model the digital twin.

A model created in this manner can ensure the behavior of an IoT device via software, and it can be easily used for simulation. Implementation was performed using a transition library [37].

When designing an IoT device, its implementation can be considered through an FSM. The first level of the state machine is the device itself. When START is switched on, the start and setting of the initial states occur. STATE 1 represents the functionality itself, and END is the shutdown process of the device (Figure 6).



FIGURE 6. Basic FSM.

For a real design, it is necessary to create a nested STATE1 model and describe its internal operations. Each condition represents another FSM. Modeling a real prototype usually leads to a sophisticated system, and a system of individual nodes is created in which each node represents a separately solvable task.

Transitions are shown as arrows that can provide data for the inputs and outputs of the states; in reality, they represent a means of communication and the possibility of connection using Application Programming Interface (API) or by allocating memory addresses. The first abstraction and the FSM levels for the IoT function are shown in Figure 7.

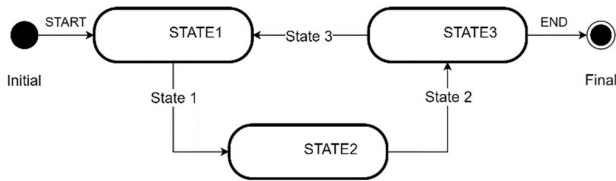


FIGURE 7. The basic function level of FSM.

The node that representing the next FSM is designated State 2, as shown in Figure 8, and can be represented as an individual FSM. It is possible to create a tree of FSM, in which every state has its own start section, process states, and end state.

The input data are handled by the start state, and they represent the reception of the input data. The tasks are represented as states 1, 2, and 3. The final state, that is, the end state, ensures the transfer of the parameters to its superior state. A description of this principle is shown in Figure 8.

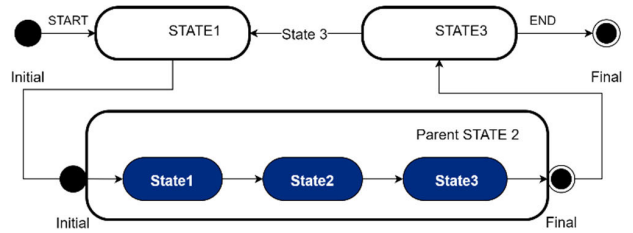


FIGURE 8. An example of a simple FSM tree in which STATE3 is the next FSM.

B. BASIC COMPONENTS OF FSM

Therefore, it is possible to prepare an FSM for this type of device design. The basic components used to create the finite-state machine are states and transitions. States define states, and transitions provide links and logic as states change with respect to the ongoing operations. Transitions can generate signals and call back signals. To model a state machine, it is sometimes necessary to provide transitions from multiple states. A trigger can be connected to multiple transitions, some of which can begin or end in the same state.

Reflexive transitions are transitions that have the same state as the source and destination from multiple states and are available for more complex state changes.

They are used when the same reflex trigger is required in multiple states. Each state can perform a function triggered by a trigger. Under trigger conditions, it is possible to imagine changes in the state as well as changes in values.

After modeling the state machine, particularly in the case of multiple transitions, it is appropriate to validate the FSM model. The library allows for the checking and validation of the prepared model. The basic element of the using of state machines is the possibility of implementing conditional transitions. For real deployment in the production process, it is used to change the set parameters and influence the behavior of the state machine.

Callbacks are another critical type of signal. These provide feedback when moving from one state to another. In this solution, it is possible to use two states that specify the transition of the state: BEFORE and AFTER. Both usually contain the methods or calls that need to be executed. In this way, it is possible to respond to state changes and ensure that the model responds to incoming data that affect the behavior of the final state machine.

With callbacks and transitions, interfaces that can connect a digital twin model to the real world can be added. In the proposed solution, this property is used to ensure communication during the integration of a digital twin into the production line.

For the simple verification of functionality, because the MES system was available, the MQTT broker provided live data, and historized data were loaded from the relational database of the Historian subsystem.

In Figure 9, a block diagram of the design is provided. The proposed model was created using the Python software.

Mmodel consists of a simulation interface that provides an interface only for compatibility with simulation platforms.

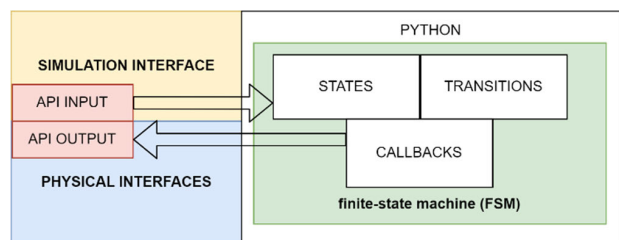


FIGURE 9. Block scheme diagram of a digital twin design using FSM.

C. INTERFACES AND COMMUNICATION

A physical interface is the layer responsible for communicating with the outside world. Both the simulation and communication layers rely on an API with input and output methods in place to communicate with the real world. The simulation layer may be independent of the physical interface, and the API may vary as needed; thus, they have been visibly separated.

The original proposal did not contain this distinction and only one API was prepared. During development, however, the reasons for modifying the simulation and physical interfaces were discovered. The reason for this change in the concept was that the digital twin can be used as a prototype to which it is possible to connect physical parts and thus create a hybrid digital twin, in which parts of the data are generated by a physical device and others by software simulation. As an example, the design of an IoT device for detecting the sorting quality, in which the entire model was created as a software digital twin, is discussed.

In the proposed model, an IoT device with a real camera was tested, and its image signal processing offers the possibility of connecting a combination of nodes. These nodes are implemented using software and a modular connection of a physical device, which can be replaced by software.

For the camera, the original implementation consisted of a selection of random images that were ready for artificial intelligence (AI) learning.

Similarly, the AI module, which processed the signal and evaluated the quality of the sorting as long as it was communicating and receiving data, could integrate and enable it in the digital twin or simply simulate its function, for example, by using synthetic parameters of shift speed control.

A schematic of the prototype 4.0 hybrid digital twin connection is shown in Figure 10.

Digital twin communication can occur in two ways. Live data generated during production are available. The MQTT broker is used for connectivity with the production line and as a connector between the production line, MES system, and the FSM model. The MQTT broker should be configured in a manner that provides communication according to the first in/first-out (FIFO) queue.

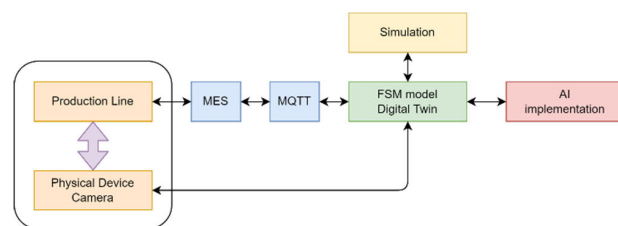


FIGURE 10. The connection of the model along with the physical device of the camera.

This is a prerequisite for configuring the queue because the FSM requires an exact sequence of the data. The goal is to guarantee that the messages sent to the MQTT broker will have the same exact order. However, the state machine expects exact signals to implement changes in its state. In the case of the simulation, this phenomenon is relatively easy to maintain.

The sequence of data entering the input can be easily observed. For practical use, however, it is better to create an identifier and not rely on MQTT to ensure a FIFO message queue. The timestamp, as a sequence identification parameter, may not have the expected effect, as the MES system does not read the timestamp directly on the controller, but it generates its own during data loading.

For example, time is generated according to the system time during which the MES system is running. For this type of communication, it is advisable to introduce an identifier as a sequence that identifies and guarantees the implementation of the state changes in a guaranteed order.

The Industry 4.0 concept enables event-based generated data that can be considered as a reliable source of data, as the event is triggered by a state change, in which case the internal timestamp is relevant and represents the correct value.

In a manufacturing process that relies on cyclic data retrieval, a timestamp must be generated when the controller state changes. Usually, however, a timestamp is used, which is generated on the side of the system that collects data.

D. DATA COLLECTION PROCESSES AND ANOMALY DETECTION

Another source of data is the Historian system, in which data from an already completed production process are stored.

Because these data were already recorded, it was only necessary to ensure their loading. For simplicity, the data were exported from a Structured Query Language (SQL) database, applied to a General Data Protection Regulation (GDPR) filter, and subsequently exported to comma-separated values (CSV) before use. The data assigned in this way are suitable for processing, but simultaneously, it is possible to detect a state in which the read data are missing or have not yet been saved.

The error state blocks the FSM, as it is impossible to decide which subsequent state is necessary to move. It is possible to detect the conditions and check the data sequence.

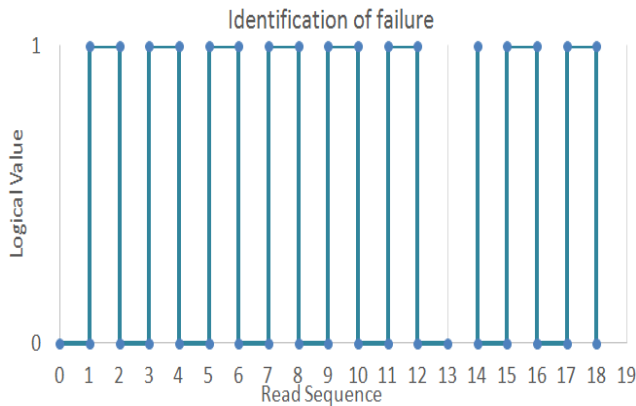


FIGURE 11. Identification of failure.

If the error that occurred is relevant, it is possible to continue to processing this information and decide whether the situation needs to be solved at the level of production control and possibly correct the program, IoT, or PLC component. Failure identification is shown in Figure 11. Failure occurred between sequences 13 and 14.

Sequence read number 14 indicates the start of a logical value of one. The previous value is generated at a logical value of zero, which explains why the state machine does not register a state change.

It is possible to decide on the implementation of a state machine to fix this behavior, and there are several options to solve the problem. A simple solution is to rely on the data generated using the production process. Here, each error manifests itself as it occurs on the production line. The state machine reports an error when, for instance, the maximum time required to execute an activity on a given node is reached.

The IoT device remains in this state. It is necessary to perform manual interventions, which may be related to the production process. This can also serve as a simulation for operators of how the production process will change and what needs to be provided to treat and resume production in a failure situation.

If it is necessary to ensure the continuity of the state machine, it is possible to turn on the option that allows the process to continue by defining the maximum time during which it is possible to wait for input data for a given state of the state machine and to set its behavior to ensure the possibility of continuation.

In the case that this state occurs in the connection, when the model is a function of an integrated digital twin in the production process, this step is performed using the parameter called max lifetime. In principle, a configuration parameter is always available, which defines the task of the proposed model and its functional mode.

The mode can only be determined through the simulation and testing of its integration into the production process. Another possibility is the simulation of a device with prepared data from the MES system Historian, which can also test

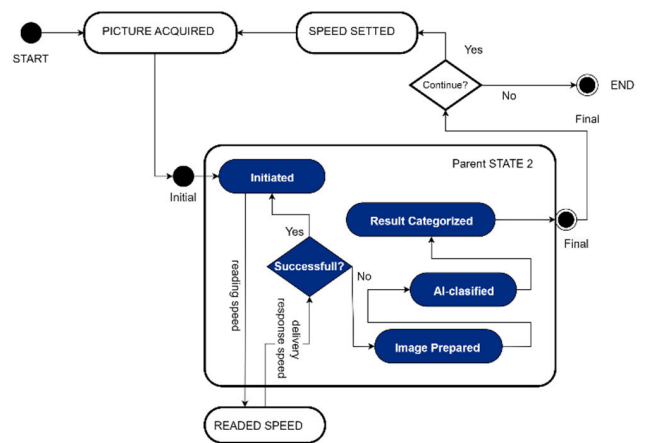


FIGURE 12. The basic model of state machine is used.

borderline situations and collisions arising from their possible use in the production process.

For basic settings, it is possible to specify how the mean data processing time for each state change is applied.

To meet these requirements, it is advantageous to use data recorded during production, and based on their average values, it is possible to determine the execution time of a given operation.

For a more accurate simulation, it is appropriate to identify and quantify defects that may occur during the manufacturing process.

In this way, it is possible to determine the percentage of failures, and for possible use as simulation entities, it is appropriate to implement a parameter that will ensure the representation of failures in the required number. This parameter can be specified by the mean failure time, such that the possible time between failures corresponds to the actual production process or is completely dynamically configurable.

Furthermore, it can be linked to other production parameters as required, and it can follow simulations that may affect the parameter. All the above parameters can be applied to stream data used in real-time models.

In the case of a connection to a real production process, it is possible to reconfigure the behavior of the state machine with respect to the time priority of the given operation. In a situation where this is a real error in the production process, in terms of a delay of several seconds or minutes, it is advisable to decide again how the state machine will behave and allow the configuration of its behavior.

Here, it is required that the behavior used for the simulation or testing does not occur; rather, the model works only with the data generated by the manufacturing process.

E. IMPLEMENTATION OF THE IIoT AS A DIGITAL TWIN

The proposed model is verified using an IIoT device. The IIoT device evaluated the sorting quality and, based on the

detected errors, provided feedback on the change in the feed rate parameter of the sorted material. The basic model of the state machine is shown in Figure 12.

The initial state, marked START, executes the start of the system, sets the initial values, and resets the model to the basic state. The get-picture state is responsible for capturing the image or video sequence and sending it to the artificial intelligence (AI) system, where an erroneous sorting process is identified based on the detection of an object that is otherwise color-coded in our implemented case.

AI evaluation identifies the number of defective elements at a given feed rate. This parameter is passed to the new speed setting for evaluation.

The speed setting is done in a step-wise manner at the beginning. It starts from a zero feed rate and continues until the output starts detecting mis-sorted elements. It records the geometric mean for the short-term sorting cycle. This detects a short-term change in sort quality and indicates whether a change in speed is required. The long-term sort quality is recorded in the arithmetic mean and is recorded according to the time window of the captured image.

The AI model was then trained for object shape and color recognition. In the test case, it involves detection of marbles and their color at the output of the sorting process. A pre-trained model was used which was fine-tuned for bulk material sorting.

The proposed Digital twin and prototype based on this principle is implemented as a software solution. When data available from the manufacturing process can be used, the initial settings that affect the system behavior will contribute to the system. The operation times, amount of data, required evaluation speed, and other parameters can then be set during the start of the START node.

To simulate a PICTURE ACQUIRED state, sets of images were prepared. These images are also needed for learning the AI module; therefore, it is sufficient to prepare sequences that will simulate the connected physical device, which, if ready, can be directly integrated into the state machine model, thus creating a hybrid solution. The functionality was verified by a test set of pictures.

In this way, it is possible to verify the basic algorithm of the proposed solution, determine the deviations, verify the over regulation, and determine the time required to achieve a stable setting of the feed rate of the sorted material.

The final decision is based on the AI system results. The calculation of maintaining the speed parameter is completed and there is done decision base on geometry mean.

This condition indicates a parameter change that repeats the process of monitoring the quality of sorting and re-evaluation.

The entire solution can be advantageously tested during an ongoing production process. It does not have to be a direct part of the system; however, by using the data created by the production process, it is possible to read these values and monitor the behavior of the system. Based on real data,

it is possible to monitor how this affects the settings of the production process.

IV. DISCUSSION

The implemented model was tested on a project that worked with test datasets from a production line. A state machine is designed for this study. Therefore, a functional prototype is available. This makes it easier for development teams to define tasks that can solve the tasks entrusted to them. Simultaneously, they have the opportunity to test the developed part of the solution, allowing them to better understand and imagine the final product.

The advantage is that when defining the AI interface for states in the FSM, the model determines exactly how communication will occur, and their assigned tasks can be validated against the prepared model. The disadvantage of such a design may be changes in the states in the FSM or changes in the initial design of the model, which are required from time to time.

Figure 13 shows the course of speed adjustment manually by the production operator, which is shown in blue. The second waveform is calculated according to the verification of the output control by AI detection. The readjustment occurs in both cases. The control deviation was larger for manual adjustment. Simultaneously, the additional manual tuning and then the steady-state sorting setting, which is set by the production operator, can be seen. When the sorting quality is detected for the proposed solution, faster speed adjustment but also speed sorting correction can be observed, during the entire processing time of the material batch.

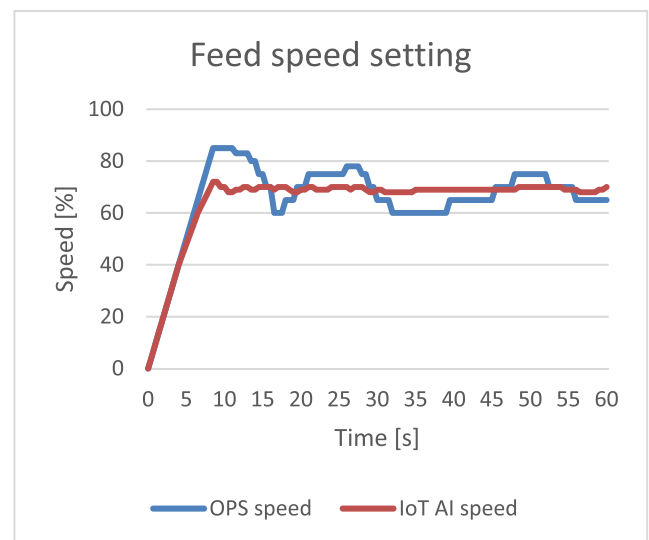


FIGURE 13. Comparison of speed adjustment manually by the operator and using IoT.

The model is implemented during the solution design. IoT devices are mostly activity oriented; therefore, the FSM, given the proposed IoT solution, is relatively simple. For correct model design, it is appropriate to consider that

each state represents one result of comprehensive activity. If necessary, each state, action, or condition can be described as a unique FSM. The proposed model was also tested from the viewpoint of project management, with a focus on parallel implementation. The model of the proposed state machine was initially implemented only as a software model, in which each state had defined input and output parameters and, if possible, their address space or protocol was defined by data that were exchanged between states. All states in the prototype had preset data that were selected from the MES system or generated randomly. The goal was to capture the behavior as truthfully as possible, according to reality. Randomly generated values can only be used to verify the functionality. It is up to the teams to decide which data will be prioritized. The test model was divided into six tasks, as listed in Table 2.

TABLE 2. Defining states and division of into tasks.

State	Description	Task
Picture Acquired	Receive image or video stream, prepare it for AI	Camera interface
Readed speed	Read speed setup via MES system, export parameters to MQTT Broker	Read speed parameters from PL
Speed setted	Set the speed setup via a MES system, export parameters to MQTT Broker	Set speed parameter to PL
start	Read config and setup initial parameters	Initial Setup
end	Save reports, shut down the device	Stop sequence
(AI FSM sub state)	Receives images or video streams and detects the quality of the sorted material based on pre-prepared examples.	Detect the quality of sorting AI
(AI FSM) Image-Prepared	Prepare images for processing	Image preparation
(AI FSM) AI-classified	AI implementation for image classification and detection of incorrectly filtered number of particles	AI-classification
(AI FSM) Result Categorized	Result of categorization with the detection of whether speed should be changed or not	Result preprocessing

The FSM is illustrated in Figure 12. The AI state is a sub-FSM that causes other tasks to occur. For the input and output in AI, it is possible to omit START and END or solve them separately.

Here, we omit them because they represent only one communication channel for data exchange.

The states designated as read speed and set-speed were implemented as MQTT subscribers with push and pull functionalities. Each of these was implemented independently. The AI part was performed as an image recognition process and split into sub-FSMs; out of this, the next three parallel tasks were created.

Altogether, it was possible to prepare tasks for seven independent teams, and the model provided the ability to start working on their tasks with a better understanding and the possibility of testing all their phases of development.

However, the agile style of the project was limited by the use of the created model. It can be a disadvantage in those teams do not have the possibility to discuss the design, but on the other hand, it brings more stability into the development process, providing each team with a basic definition and full overview of the process. Subsequent decisions were made by the teams, and they could prepare their own solutions, but these were supported by the model.

An advantage of this model is that the representation of the FSM can be used to devise an OPC UA solution by default. However, if an improvement is needed in the basic concept of the model, it leads to a situation in which the model evaluation is needed and changes are required. In this scenario, there should be another validation process and a decision on how to continue with the development.

Nevertheless, not all states usually need to be changed, and improvement is needed for only a few of them. The described design for developing of an IoT system using an FSM according to the application development delivered the expected benefits and made it possible to plan and develop each part of the process as a unique task. It can be performed in parallel.

The model was tested using several designs. The solution described here represents only one possible way to implementation. The requirements defined in manufacturing factories are not always clearly specified because of the time and speed of adaptation of the manufacturing process; therefore, the implementation was based on agile project management.

The model was tested several times in such an environment. Cicirelli et al. [38] described the IoT proposal using an intelligent metamode environment model. In the case of implementation through agile development, the goal of involving entire teams is defined as one of the problem in the design process. After an initial analysis, the topic of how to create and assign multiple subtasks was developed to allow scrum teams to work in parallel.

The FSM model was designed to provide stability and transparency of the tasks so that each team could process or select a specific part to implement or continue. The tasks were defined as sprints, for which the time was set for a two-weeks period.

Then, in the process of agile development called retro, a revision of tasks and implementation was performed. It was determined here which parts of the task are processed well or not well enough, the areas that need improvement were identified here as well as changed design and improvement of the implementation, or whether another change is needed as well. Comparisons of time, task, and error rates were made for the same teams that worked on assignments, with and without the model. An overview of this process is provided in Table 3.

The tasks solved were similar. The basic overview describes the time and number of tasks created, and the number of registered errors.

The measurability of the impact and the impact of the experience gained by the teams before implementation

TABLE 3. The design of IoT with or without FSM.

Description	Changes in design	Duration in weeks	Task	Teams	Bugs
IoT design with FSM	3	6	6	4	10
IoT design without an FSM	8	9	8	4	19

and the definition of the exact differences in the project may represent differences. Regardless of these inaccurately defined parameters, the use of the FSM model and how the model works were presented with positive feedback from the teams.

Project planning and comparison of agile and „waterfall“ project management are beyond the scope of this article, and so it can be further analyzed.

The model can be used in both forms of management, and can help identify and create tasks or subtasks, and those after the implementation of the software model can be implemented in parallel. Understanding the functionality and tasks can improve the development of IoT devices.

The implemented solution was tested on a production line. Small balls were selected for testing. They differed in color and weight.

The feed rate is set manually by the production operator. When changing the size of the sorting material, poor-quality processing occurred, and the required quality was not achieved.

The operator then changed the feed rate according to the subjective perception of the sorting quality. A test set was prepared in which 90% of the material was acceptable and 10% represented the defective particles to be sorted out. Three different sets of materials are available, and their particle weights are listed in Table 4.

TABLE 4. Particle weights of the material.

Weight [g]	Color	Shape
0.20	Yellow/Gray	Balls
0.30	Yellow/Gray	Balls
0.40	Yellow/Gray	Balls

This set was used to evaluate the quality of the sorted feed. Moreover, the model is into the production process. The result is expressed as the evaluated speed of the conveyor belt feed, which fed the material to the RGB sorter. A subjective comparison of the sorting quality set by the operator and that of the prepared model is shown in Figure 14.

The quality of the AI setting is represented by the speed setting, which never reaches 100%. Thus, it was possible to experiment with and evaluate or teach speed control according to the weight or size of the sorted particles.

These results were obtained using a hybrid solution, in which a camera was connected to the production process

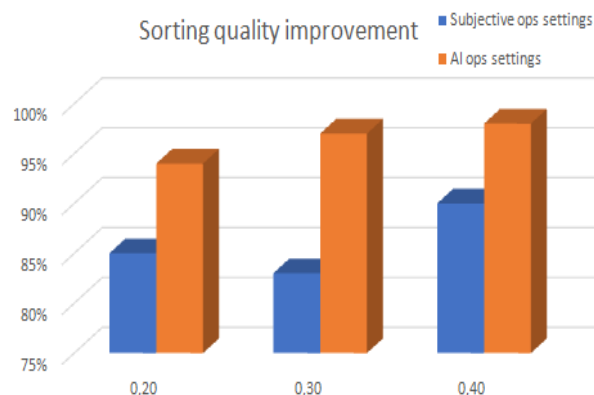


FIGURE 14. Comparison of success and improvement in material sorting quality.

and the other parts were implemented only in software without the final hardware solution.

The results always showed an error rate, which was caused by the adaptation of the AI to the situation. When processing large batches, the percentage of error was insignificant because inaccuracy occurred only when the wrong material was fed. There is no guarantee that a faulty or correct particle will enter the sorting system.

The simulation of the IoT device was prepared in the first part of the design. In the case of implementation, the parameters obtained were immediately ready for practical use in the simulation process. Each node in the state machine was provided, either configured or based on real datasets that were needed for the simulation. In the case of software implementation, it is appropriate to set the goal of achieving the proposed solution of the IoT device.

This also involves a solution that ensures connectivity using the selected protocols for simulation software development, performance verification, and the possibility of an open simulation interface (OSI) [39].

A design prepared in this manner will be easier to use for complex tasks and will be sufficiently easy to integrate into simulation processes.

V. CONCLUSION

The development of an IoT device based on the proposed model makes it possible to verify the functionality of the solution and demonstrate the capabilities of the intended solution. Simultaneously, it is possible to use such a software solution as part of the simulation and create a digital twin of the IoT system based on the FSM model.

The proposed solution is suitable for the implementation and design of a physical device and creates an opportunity to verify the proposed method before the final implementation. It represents a flexible design, which makes it possible to ensure the agile development of various parts of the system, and it also represents an ideal opportunity to approach the development of individual tasks in parallel.

A disadvantage of our solution is its inability to verify the hardware requirements. The prototype itself still allows for a functional solution, but no additional data collection options are in place, such as the amount of data sent between nodes, which can provide essential information when selecting a hardware IoT design. Another desired parameter is the power required to perform individual operations at a given node.

This requirement is not currently addressed; therefore, the prototype itself does not yet allow users to classify and evaluate the requirements for computing power and the possible power consumption of the device itself. Conversely, the advantages of a rapid prototype solution and its implementation can help address various manufacturing scenarios, and its ease of integration and adaptability can help decide whether to integrate IoT systems and Industry 4.0 into the manufacturing process.

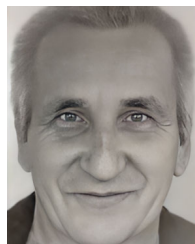
REFERENCES

- [1] M. Piccarozzi, B. Aquilani, and C. Gatti, "Industry 4.0 in management studies: A systematic literature review," *Sustainability*, vol. 10, no. 10, p. 3821, Oct. 2018.
- [2] J. M. Müller, D. Kiel, and K.-I. Voigt, "What drives the implementation of industry 4.0? The role of opportunities and challenges in the context of sustainability," *Sustainability*, vol. 10, no. 1, p. 247, Jan. 2018.
- [3] J. Ang, C. Goh, A. Saldivar, and Y. Li, "Energy-efficient through-life smart design, manufacturing and operation of ships in an industry 4.0 environment," *Energies*, vol. 10, no. 5, p. 610, Apr. 2017.
- [4] C. Chute and T. French, "Introducing care 4.0: An integrated care paradigm built on industry 4.0 capabilities," *Int. J. Environ. Res. Public Health*, vol. 16, no. 12, p. 2247, Jun. 2019.
- [5] H. Meissner and J. C. Aurich, "Implications of cyber-physical production systems on integrated process planning and scheduling," *Proc. Manuf.*, vol. 28, pp. 167–173, Jan. 2019.
- [6] M. Hermann, T. Pentek, and B. Otto, "Design principles for industrie 4.0 scenarios," in *Proc. 49th Hawaii Int. Conf. Syst. Sci. (HICSS)*, Jan. 2016, pp. 3928–3937.
- [7] V. Salauyou and M. Ostapczuk, "State assignment of finite-state machines by using the values of input variables," in *Proc. IFIP Int. Conf. Comput. Inf. Syst. Ind. Manag.*, in Lecture Notes in Computer Science, 2017, pp. 592–603.
- [8] P. Vazan, Z. Cervenanska, J. Kotianova, and G. Krizanova, "The impact of selected priority rules on production goals," in *Proc. 20th Int. Carpathian Control Conf. (ICCC)*, May 2019, pp. 1–6.
- [9] K. Židek, J. Pitel, M. Adámek, P. Lazorič, and A. Hošovský, "Digital twin of experimental smart manufacturing assembly system for industry 4.0 concept," *Sustainability*, vol. 12, no. 9, p. 3658, May 2020.
- [10] A. Madni, C. Madni, and S. Lucero, "Leveraging digital twin technology in model-based systems engineering," *Systems*, vol. 7, no. 1, p. 7, Jan. 2019.
- [11] T. Lins and R. A. R. Oliveira, "Cyber-physical production systems retrofitting in context of industry 4.0," *Comput. Ind. Eng.*, vol. 139, Jan. 2020, Art. no. 106193.
- [12] C. Liu, P. Jiang, and W. Jiang, "Web-based digital twin modeling and remote control of cyber-physical production systems," *Robot. Comput.-Integr. Manuf.*, vol. 64, Aug. 2020, Art. no. 101956.
- [13] B. Farooq, J. Bao, and Q. Ma, "Flow-shop predictive modeling for multi-automated guided vehicles scheduling in smart spinning cyber-physical production systems," *Electronics*, vol. 9, no. 5, p. 799, May 2020.
- [14] M. Jovanovic, B. Lalic, A. Mas, and A.-L. Mesquida, "The Agile approach in industrial and software engineering project management," *Istrazivanja i Projektovanja za Privredu*, vol. 13, no. 4, pp. 213–216, 2015.
- [15] B. Koo, J. Bae, S. Kim, K. Park, and H. Kim, "Test case generation method for increasing software reliability in safety-critical embedded systems," *Electronics*, vol. 9, no. 5, p. 797, May 2020.
- [16] J. Muttersbach, T. Villiger, and W. Fichtner, "Practical design of globally-asynchronous locally-synchronous systems," in *Proc. 6th Int. Symp. Adv. Res. Asynchronous Circuits Syst.*, 2000, pp. 52–59.
- [17] L. Monostori, B. Kádár, T. Bauernhansl, S. Kondoh, S. Kumara, G. Reinhart, O. Sauer, G. Schuh, W. Sihn, and K. Ueda, "Cyber-physical systems in manufacturing," *CIRP Ann.*, vol. 65, no. 2, pp. 621–641, 2016.
- [18] J. H. Kim, "A review of cyber-physical system research relevant to the emerging IT trends: Industry 4.0, IoT, big data, and cloud computing," *J. Ind. Integr. Manage.*, vol. 2, no. 3, Sep. 2017, Art. no. 1750011.
- [19] J. Nagy, J. Oláh, E. Erdei, D. Máté, and J. Popp, "The role and impact of industry 4.0 and the Internet of Things on the business strategy of the value chain—The case of Hungary," *Sustainability*, vol. 10, no. 10, p. 3491, Sep. 2018.
- [20] I. Zambon, M. Cecchini, G. Egidi, M. G. Saporito, and A. Colantoni, "Revolution 4.0: Industry vs. agriculture in a future development for SMEs," *Processes*, vol. 7, no. 1, p. 36, Jan. 2019.
- [21] D. Jones, C. Snider, A. Nassehi, J. Yon, and B. Hicks, "Characterising the digital twin: A systematic literature review," *CIRP J. Manuf. Sci. Technol.*, vol. 29, pp. 36–52, May 2020.
- [22] R. Jihin, F. Kogler, and D. Söffker, "Data driven state machine model for industry 4.0 lifetime modeling and identification of irrigation control parameters," in *Proc. Global IoT Summit*, 2019, pp. 1–6.
- [23] N. Walkinshaw, R. Taylor, and J. Derrick, "Inferring extended finite state machine models from software executions," *Empirical Softw. Eng.*, vol. 21, no. 3, pp. 811–853, 2016.
- [24] W. Zamojski, J. Mazurkiewicz, J. Sugier, T. Walkowiak, and J. Kacprzyk, Eds. *Theory and Applications of Dependable Computer Systems: Proceedings of the Fifteenth International Conference on Dependability of Computer Systems DepCoS-RELCOMEX, June 29–July 3, 2020, Brunów, Poland*. Cham, Switzerland: Springer, 2020.
- [25] C. Cimino, E. Negri, and L. Fumagalli, "Review of digital twin applications in manufacturing," *Comput. Ind.*, vol. 113, Dec. 2019, Art. no. 103130.
- [26] G. Caiza, A. Nuñez, C. A. Garcia, and M. V. Garcia, "Human machine interfaces based on open source web-platform and OPC UA," *Proc. Manuf.*, vol. 42, pp. 307–314, Jan. 2020.
- [27] *OPC UA Documentation*. Accessed: Apr. 18, 2022. [Online]. Available: <https://reference.opcfoundation.org/v104/Core/docs/Part5/B.1/>
- [28] K. Zidek, D. Janacova, A. Hosovsky, J. Pitel, and P. Lazorič, "Data optimization for communication between wireless IoT devices and cloud platforms in production process," in *Proc. 3rd EAI Int. Conf. Manage. Manuf. Syst.* Ghent, Belgium: European Alliance for Innovation, 2018, p. 118.
- [29] *Tempest: Production Line Description User Manual*, Tempest, Chennai, India, 2014.
- [30] G. Cwikła and K. Foit, "Problems of integration of a manufacturing system with the business area of a company on the example of the integrated manufacturing systems laboratory," in *Proc. Web Conf.*, 2017, p. 11.
- [31] I. Bedhief, L. Foschin, P. Bellavista, M. Kassar, and T. Aguilí, "Toward self-adaptive software defined fog networking architecture for IIoT and industry 4.0," in *Proc. IEEE 24th Int. Workshop Comput. Aided Modeling Design Commun. Links Netw.*, Sep. 2019, pp. 1–5.
- [32] M. Aazam, S. Zeadally, and K. A. Harras, "Deploying fog computing in industrial Internet of Things and industry 4.0," *IEEE Trans. Ind. Informat.*, vol. 14, no. 10, pp. 4674–4682, Oct. 2018.
- [33] T. M. Fernández-Caramés and P. Fraga-Lamas, "Use case based blended teaching of IIoT cybersecurity in the industry 4.0 era," *Appl. Sci.*, vol. 10, no. 16, p. 5607, Aug. 2020.
- [34] H. Boyes, B. Hallaq, J. Cunningham, and T. Watson, "The industrial Internet of Things (IIoT): An analysis framework," *Comput. Ind.*, vol. 101, pp. 1–12, Oct. 2018.
- [35] A. Jayaram, "Lean six sigma approach for global supply chain management using industry 4.0 and IIoT," in *Proc. 2nd Int. Conf. Contemp. Comput. Informat. (IC3I)*, Dec. 2016, pp. 89–94.
- [36] K. Chooruang and P. Mangkalakeeree, "Wireless heart rate monitoring system using MQTT," *Proc. Comput. Sci.*, vol. 86, pp. 160–163, Jan. 2016.
- [37] A. Neumann and T. Yarkoni, Pytransitions Transitions. Germany. Accessed: Apr. 28, 2022. [Online]. Available: <https://github.com/pytransitions/transitions>
- [38] F. Cicirelli, G. Fortino, A. Guerrieri, G. Spezzano, and A. Vinci, "Metamodeling of smart environments: From design to implementation," in *Advanced Engineering Informatics*, vol. 33, 2017, pp. 274–284.
- [39] *Open Simulation Interface (OSI)*. Accessed: May 7, 2022. [Online]. Available: <https://opensimulationinterface.github.io/osi-documentation/index.html>

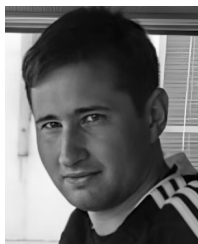


PETER STRELEC received the B.S. degree in graphical data processing, in 2003, and the M.S. degree in information technology and automation in industry, in 2005. He is currently pursuing the Ph.D. degree in process automation and informatization with the Faculty of Materials Science and Technology in Trnava, Slovak University of Technology in Bratislava. He is an Engineer with Telco Company, Austria. His research interests include automation, the development of the Internet of

Things, machine learning, wireless sensor networks, and data processing and data mining.



EDUARD NEMLAHA received the Ph.D. degree from the Faculty of Materials Science and Technology in Trnava, Slovak University of Technology in Bratislava, in 2010. He is an Assistant Professor of applied informatics, automation and mechatronics. He has considerable experience in the design and implementation of database systems and internet technologies. His research and development interests include automation and control of production systems, management and design of safety systems, and PLC applications.



TIBOR HORAK received the Ph.D. degree from the Faculty of Materials Science and Technology in Trnava, Slovak University of Technology in Bratislava, in 2021. He is with the Faculty of Materials Science and Technology in Trnava, Slovak University of Technology in Bratislava, where he teaches the subjects, such as computer networks, information technologies, and internet technologies. He is an author of several professional and scientific articles. His professional

interest includes cyber security of the Industrial IoT devices integrated in manufacturing processes based on the identification of vulnerabilities by implementing security strategies (rules) to prevent potential cyber threats in various industries in accordance with the concept of Industry 4.0. He is a member of several research teams dealing with scientific projects (VEGA, KEGA, and young researcher) focused on the research of the IoT devices from the perspective of security applied in the industry.



SZABOLCS KOVAC received the master's degree in automation with a specialization in process automation and informatization in industry from the Faculty of Materials Science and Technology in Trnava, Slovak University of Technology, where he is currently pursuing the Ph.D. degree in process automation and informatization. His research interests include artificial intelligence, signal processing, and load forecasting.



PAVOL TANUSKA received the Ph.D. degree from the Faculty of Materials Science and Technology in Trnava, Slovak University of Technology in Bratislava, in 2000. Since 2013, he has been a Professor with the Institute of Applied Informatics, Automation and Mechatronics, Slovak University of Technology, Trnava, Slovakia. He has published over 120 scientific and technical papers in national and international conference proceedings and journals. His research and development interests

include big data, data collection, processing, and analysis of an extremely large amount of data in order to knowledge discovery for the needs of production systems control. He has been a member of IACSIT, since 2009, and IAENG, since 2010.

...