



Article

A Discrete Process Modelling and Simulation Methodology for Industrial Systems within the Concept of Digital Twins

George Tsinarakis * , Nikolaos Sarantinoudis  and George Arampatzis 

Industrial and Digital Innovations Research Group (INDIGO), School of Production Engineering and Management, Akrotiri Campus, Technical University of Crete, 73100 Chania, Greece; nsarantinoudis@isc.tuc.gr (N.S.); garampatzis@pem.tuc.gr (G.A.)

* Correspondence: tsinar@dpem.tuc.gr

Abstract: A generic well-defined methodology for the construction and operation of dynamic process models of discrete industrial systems following a number of well-defined steps is introduced. The sequence of steps for the application of the method as well as the necessary inputs, conditions, constraints and the results obtained are defined. The proposed methodology covers the classical offline modelling and simulation applications as well as their online counterpart, which use the physical system in the context of digital twins, with extensive data exchange and interaction with sensors, actuators and tools from other scientific fields as analytics and optimisation. The implemented process models can be used for what-if analysis, comparative evaluation of alternative scenarios and for the calculation of key performance indicators describing the behaviour of the physical systems under given conditions as well as for online monitoring, management and adjustment of the physical industrial system operations with respect to given rules and targets. An application of the proposed methodology in a discrete industrial system is presented, and interesting conclusions arise and are discussed. Finally, the open issues, limitations and future extensions of the research are considered.

Keywords: process modelling; digital twins; cognitive manufacturing; industrial systems; Industry 4.0; discrete event systems; Petri nets; Internet of Things; optimisation; analytics



Citation: Tsinarakis, G.; Sarantinoudis, N.; Arampatzis, G. A Discrete Process Modelling and Simulation Methodology for Industrial Systems within the Concept of Digital Twins. *Appl. Sci.* **2022**, *12*, 870. <https://doi.org/10.3390/app12020870>

Academic Editors: Manuel Davila Delgado and Tom Beach

Received: 29 November 2021

Accepted: 12 January 2022

Published: 15 January 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

The intended objective of this research is the definition and development of a generic well-defined methodology for the implementation of the dynamic process models of discrete industrial systems and their operation for simulating alternative scenarios (what-if analysis). After the specification of certain, well-defined and described parameters of the industrial systems under study, and with the application of appropriate tools, the proposed methodology enables the construction of efficient models. Typically, the process models can be used for the calculation of key performance indicators describing the behaviour of the physical systems under given conditions and for what-if analysis and performance prediction.

The proposed methodology is applied in the context of digital twins (DT) that expands and updates significantly the importance and the usability of the models. A DT is a virtual representation that reflects and simulates a physical object (e.g., a product, a process, a service or a system), its environment and interaction, providing a picture as accurate as possible of how that object behaves in real time. In DTs, process models are no more just passive design or forecasting tools that mirror the state of the physical systems, but active components that interact with the physical systems bidirectionally. The current state of the model is updated using information received from the installed sensors, and with the use of appropriate decision tools, such as Optimisation, machine learning and Analytics, the models can enable, monitor and evaluate the application of different types of strategies for certain periods through the performance of corrective and preventive actions enabled from actuators installed in the physical system. Digital twin relies on a core component,

called the digital shadow, that enables the management and analysis of near real-time data coming from the actual physical counterpart [1].

The main scope of this research is to define a Petri net model of a steel reinforcement industry, using a well defined Petri net model synthesis procedure that can be utilised either offline or online in the context of digital twins. In addition to it, the model construction procedure is defined analytically as a sequence of stages; the data and information needs as well as the outputs of each stage are described, in addition to the way these outputs can be utilized from process modelling tools and methodologies and associated services, such as optimisation and analytics. There is no extensive literature related to the online operation of the PN model in the context of a digital twin, so in the current research, all the main concepts related to the bidirectional exchange of data, system state mirroring, KPIs calculation, evaluation of the current system state and corrective action performance as well as the functional inclusion and enabling of these capabilities on the implemented Petri net models are discussed.

The rest of the paper is organized as follows: Section 2 introduces industrial systems, process modelling and simulation and digital twins concepts. Section 3 focuses on the description of the main definitions, types of characteristics and flows of the tools used for process modelling and simulation in the current paper. Section 4 introduces analytically the steps as well as the operation of the proposed process modelling and simulation methodology. An interesting part of this section has to do with the analysis on the role of process modelling and simulation services in the context of digital twins. Section 5 presents and explains the application of the proposed process modelling methodology in a specific discrete system case. Finally, Section 6 concludes the paper and discusses some open issues, challenges and possible future steps.

2. Research Methodology

This research paper deals with the problem of developing a process modelling method for a steel reinforcement installation and applying it not only with the traditional way, but also online with the physical system as a digital shadow. The development of the proposed methodology was performed in closed cooperation and interaction with the production manager executives of the industry, with experts from optimisation and analytics services as well as from information and communication technology (ICT) experts.

Initially, the overall problem was specified, and then tools from different areas of expertise that could deal with it were recognized. The characteristics of the industry under study, after extensive review of the literature, led us to the selection of Petri nets as a process modelling tool. The initial dataset was provided from the ICT department of the steel industry to the information and communication technology experts of the project who were responsible for data cleansing, storage, classification, selection and formatting. Then the finalized data were given to all service providers. After extensive discussions and interviews with the experts of the industry, an initial solution was suggested and revised to cover all the assumptions set and the needs of the industry. The modularity of the proposed approach is a fundamental need, as it allows the management and study of complicated cases and also enables the update of the process models in cases of changes on the production procedure. In addition, the proposed methodology was developed with features, making possible its interaction (through exchange of data in JSON file format using APIs) with the other services. Then the proposed method was applied in examples of real data and was validated, justified and then evaluated by ICT providers, industry and us before getting finalized. It is important to note that the method was considered both as an offline process modelling tool as well as in the context of digital twins online with the physical system. No further description of the technical part of this interoperability is given, as it is beyond the scope of the current research.

3. Relative Literature

3.1. Industrial Systems

According to [2], a system is a part of the real world with well-defined physical boundaries, which is affected by its surroundings or environment via its inputs and consequently affects its surroundings with its outputs. Specifically, industrial systems are those systems which include the context, resources, activities, processes, actors, and interdependencies that support the creation and delivery of products and services [3]. In order to model any industrial system, its state space needs to be defined in detail, i.e., the variables that govern the behaviour of the system with respect to the metrics being estimated. There are two main categories of industrial systems: discrete event systems, where the state changes occur at discrete instances in time and continuous process systems, where state changes occur continuously over time.

3.2. Process Modelling and Simulation

The main objective of a process model is to develop a scientific understanding through quantitative expressions of knowledge of a system by displaying what we know, but may also reveal things that we do not know. It is also ideal for testing the effects of changes in a system, making predictions and helping in decision making, including both tactical decisions by managers as well as strategic decisions by planners. In order to underline the significance and the widespread application of the process modelling and simulation approaches in the present industrial and social systems, some of the most popular and promising application areas, as specified from our extensive research and academic activity on the domain of digital twins and applications, are highlighted.

- **Smart Industries:** Cyber physical systems manage processes and monitor each other, while interconnected software manages the local operating and interacting parts (entities) of a company.
- **Global Virtual Factories:** Network of businesses from multiple regions that resolve the issues of processes, dependencies and interrelations, data and material flows between companies and between customers and suppliers.
- **Asset Performance Management:** Improved reliability and performance of equipment and assets is achieved through better visibility, availability and predictability enabled from tools and algorithms from different scientific fields.
- **Process and Quality Improvement:** Yield and productivity of manufacturing operations is optimized, covering the operations from design through warranty support.
- **Resource Efficiency Optimisation:** Optimising energy efficiency, facility productivity, product reliability, quality, safety and yield, while reducing costs, resource idleness and production time.
- **Supply Chain Optimisation:** Improving visibility and insights to build a dynamic supply chain that accelerates innovation, efficiency and performance taking advantage of resources and state-of-the-art technologies.
- **Predictive Maintenance:** The condition of in-service equipment is monitored and evaluated in order to estimate when maintenance should be performed with the intention to minimize the consequences from equipment disorders (such as loss of productivity).
- **Anomaly Detection:** Patterns, events, behaviours or observations which raise suspicions by differing significantly from the normality of the observed and stored data.

To model any system, we first need to define its state space, i.e., the variables that govern the behaviour of the system with respect to the metrics being estimated. If the variables continuously change with time, it is called a continuous system. If the system state instantaneously changes at discrete points in time instead of continuously, it is called a discrete system. It is important to define and detect all the critical parameters that govern the behaviour of the system in order to understand it and set an efficient study mechanism. In the modelling and simulation procedure, there are three basic entities: the model, the simulator and the experimental frame.

The model is the system specification, defining the static and dynamic structure and behaviour needed to generate data comparable to real-world data. Typically, the structure of a model represents all the entities involved in its operation as well as a set of possible states in which the physical system can be found after the appearance of certain sequences of events. Practically different levels of detail can be considered in a model in order to make it applicable in alternative cases, and for this, certain entities can be neglected (this can be done for reduction in the model complexity or for the study of certain aspects of the system). The selection of the best model comes down to the problem of choosing the appropriate level of detail, and it is considered one of the most difficult aspects of the modelling process. A model that is too simple will be unrealistic, and its results will be of little use. Considerable resources are required to build a complex model and so, if the model is too complex, constraints on resources may make it inefficient. It is much harder to understand the relationships contained in a complex model; this makes the interpretation of the results more difficult, leading to incorrect conclusions being drawn. A complex model is probably more likely to contain errors, as it is harder to verify that it is working as intended [4].

Events define the mechanism for change of systems state and refer to interactions between entities of the system and between the system and the external environment. A significant difference between these categories of events is that it is much more difficult to control or eliminate the events in the external behaviour and their effects in the system under study (such events are called uncontrollable according to the literature, and in many cases the outcomes of their appearance may take extra time to be observed).

In addition to the structure of the model, its dynamic state must be defined. This represents, at any time instant, the active state and enables the set of possible upcoming events (active state enables certain possible upcoming events). Generally, a model consists of a set of instructions, rules, equations and constraints for generating input/output behaviour. In many cases, also a graphical representation of the model is added, but this is not necessary and is mainly for representation and better understanding.

The simulator is a kind of agent (any computation system) that executes the instruction of the model, generating the system behaviour. Simulators generally use algorithms and equations to define the way parameters of the model are updated with reference to the performance of events or pass of time. In order to produce realistic results through simulation, initially, scenarios referring to the typical system behaviour should be simulated for validation and justification purposes.

Finally, the experimental frame refers to the specification of the conditions under which the system is observed. A complete virtual run of a system's model (a scenario) for a sufficient virtual (simulated) time period (a full production cycle needed to produce a distribution batch/shipment of the product or an extended period if probabilistic system events—internal or external—must be taken into account). Throughout that period, model flows and process state changes are estimated, and system status statistics are calculated (synthesized) and aggregated to produce KPIs for assessment. System events (even including real-time monitoring inputs and monitoring system behaviour) can also be simulated by using suitable probability functions. Simulations are performed for understanding the behaviour of the system, for testing the effect of changes in it, making predictions and aiding decision making as well as for comparative evaluation of alternative strategies for systems operation.

3.3. Digital Twins

The precursor of physical and digital representations was reported by NASA's Apollo program, where mirrored systems were created to monitor and manage unreachable objects and find solutions to problems they faced. The term digital twins gained momentum in 2002 by Michael Grieves, at University of Michigan executive course on product lifecycle management (PLM) [5]. Grieves' DT model is composed of three main components: (1) a real space containing a physical object; (2) a virtual space containing a virtual object; and

(3) the link for data flow from real space to virtual space and for information flow from virtual space to real space [6]. DTs are also closely related with cyber physical systems (CPS) from many perspectives, including engineering practices, cyber–physical mapping, and core elements. The CPS term was introduced in 2006 at the National Science Foundation (NSF) to describe increasingly complex systems that could not be effectively illustrated using traditional IT terminology [7].

The popularity of DTs arose with the introduction of Industry 4.0, a concept introduced by the German government, which encapsulates a set of technological changes in manufacturing and sets out priorities of a coherent policy framework with the purpose of maintaining the global competitiveness of German industry [8]. Industry 4.0 mainly refers to the concept of factories in which machines are augmented with smart and autonomous systems enhanced by the Internet of Things (IoT), 3D printing, artificial intelligence (AI), machine learning, big data, augmented reality, etc. [9]. Industry 4.0 represents the revolutionary age of computers, machines and human interconnectivity and interaction for higher manufacturing efficiency, larger production scale, sustainable environmental outcomes and improved quality of life [10]. The main criterion of the fourth industrial revolution has to do with the automation of decision-making and problem-solving processes. Many authors support that the fifth industrial revolution has already begun and concerns the integration of the swerving strengths of cyber–physical production systems (CPPS) and human intelligence to create synergetic factories that enable personalisable autonomous manufacturing [11].

The state of the art of the DT research, concerning the key concepts and components of DTs, the current development of DTs, the major DT applications in industry categorized, as well as promising areas can be found in [12]. In [13], the authors provide a state-of-the-art review of DT history, alternative definitions and models, as well as 6 types of key enabling technologies, while Ref. [14] surveys how the digital twins technologies are integrated into and promote the smart manufacturing system design. Further applications of DTs, may be met in the fields of greenhouses, refineries, telecommunications, buildings, healthcare, vessels, asset management, lifecycle management, traffic management, product design, project management, education and many more.

In the literature, there are many definitions of DTs, as their popularity has grown exponentially in recent years. A DT is often described as a computer model that virtually reflects and simulates a physical object (e.g., a product or a system), its environment and interaction, providing a picture as accurate as possible of how that object behaves in real time. A DT creates transparency and allows the optimisation of processes and the speed up of operational decision making. The latter is what conventional model-based decision support systems try to achieve. The difference of a DT is the connection and the real-time synchronisation between the physical object and the model that makes possible the analysis of data to tackle problems before they even occur, prevents downtime, develops new opportunities and even plans for the future by using simulations. Digital twin relies on a core component, called the digital shadow that enables the management and analysis of near real-time data coming from the actual physical counterpart.

The increased interest for digital twins in industrial systems becomes evident from the development of a number of standards referring to digital twins. IPC-2551: International Standard for Digital Twins, the first international standard on digital twins enables manufacturers, design organisations and solution providers to initiate application interoperability to create smart value chains [15]. International Standardisation Organisation (ISO) released in 2021 ISO 23247-1:2021 a four-part standard series referring to digital twins [16]. This standard series comprises a set of protocols for implementing and maintaining different aspects of digital twins as scope and objectives, analysing modelling requirements, promoting common terminology usage, implementing a generic reference architecture and supporting information modelling of observable manufacturing elements (OMEs) and information synchronisation between a digital twin and its OME [17]. Other standards include IEC TS (The International Electrotechnical Commission Technical Specifications) 62832, Digital

Factory Framework, IEEE P2806, System Architecture of Digital Representation for Physical Objects in Factory Environments, IPC (Institute for Interconnecting and Packaging Electronic Circuits) 2551, International Standard for Digital Twins [18] and NISTIR 8356 (Draft) Considerations for Digital Twin Technology and Emerging Standards [19], a rather explanatory report on the current trends and standards on DTs.

3.4. Justification of the Use of Petri Nets as Modelling Tool

The main reasons that justify our selection of using Petri nets as the modelling and simulation tool in the context of digital twins can be summarised as follows:

- As stated in [20], Petri nets can describe the system and analyse the system behaviour in a very efficient manner through strict mathematical expression and intuitive graphical expression and are widely used in modelling and simulation analysis of discrete systems. The data-driven Petri net models can simultaneously simulate state-driven behaviour and the impacts of real-time working conditions, operating parameters and production load on the behaviour of the individual state.
- The existence of a wide variety of Petri net extensions and variations increases the modelling power of the initial formalism and makes them most suitable choice for a number of applications with special features or needs.
- In [21], the authors claim and discuss the fact that manual simulation modelling is not an option with the modern manufacturing systems that undergo numerous and rapid reconfigurations during their lifetimes. For this reason, they propose a data-driven development of simulation models for smart manufacturing systems, as a basis for their digital twins. In addition, data-driven digital twins will enable integrated ongoing model validation that can be started as soon as parts of the model are being extracted. The methodology presented in the current paper is compatible with the results and conclusions of the work [21].
- Most commercial simulators are not programming languages but rather graphical modelling systems. This means that they have limited flexibility according to the needs of the users concerning input, output management as well as feature calculations (e.g., KPIs). Petri nets and extensions with the integration of tools such as PYTHON (see [22]) enable the exchange and management of inputs and outputs from other tools using common files (such as JSON) for the needs of a digital twin platform. In addition, with the use of auxiliary structures, they enable the calculation, monitoring and improvement of a variety of measures.
- Petri net models are ideal to represent sequences of events in the interacting entities of a system, represented as a scenario. These scenarios can be of increased complexity according to the specified needs. Such scenarios can be considered for many reasons, including what-if analysis of the cyber subsystem of a digital twin in order to specify corrective actions or to evaluate alternatives. In addition, such scenarios can be produced automatically (through optimisation algorithms to check alternative schedules), and their implementation using a Petri net model is easier than through graphical simulation software.
- Petri nets are ideal for developing modular techniques with reuse of the models and for automatic model synthesis and automatic control. Typical examples of automatic Petri net model synthesis can be found in [23–25] and a review on applications of Petri nets for automatic control in [26]. In such cases the construction of the overall models using Petri nets is a quick and accurate procedure.
- The cost of the most popular modeling and simulation platforms is not negligible, as it fluctuates from EUR 5000–20,000, while for their test versions, there are strict limitations concerning the complexity of the models that can be constructed and of the events simulated. Concerning Petri nets, there are many open-source and freely distributed tools with advanced capabilities, such as WoPed, PIPE2, CPN, etc.
- According to [27], comparison between Petri nets and a commercial simulator regarding structural and computational complexity, running times, accuracy, construction

of scenarios, modelling power, and modelling abstraction was performed, and Petri net models were proven to be more efficient than the respective models implemented with Arena.

The use of Petri nets in the context of digital twins has also certain limitations and disadvantages that can be considered minor compared to their usefulness. Some of the most important of them concern the need for intermediate tools for data transformations to the appropriate formats, the increased complexities of models consisting of multiple entities or/and representing numerous events, the need for adding auxiliary structures for calculations (as in many cases they do not refer to parameters) and the strong mathematical skills concerning the modeller and model operators.

4. Theoretical Background—Petri Net Theory

4.1. Introduction

In this section, the fundamental theory of Petri nets with extension arcs are presented. Petri nets are selected as a process modelling and simulation tool, as they can describe efficiently the structure and the dynamic condition of systems typically considered discrete event and analyse their behaviour using well-defined tools. Petri nets provide strict mathematical, in parallel with intuitive graphical, expression that improves the understanding of the system.

Today, Petri nets are considered a popular mathematical and graphical tool widely used for modelling, analysis, synthesis, performance evaluation, simulation and control of processes and systems typically considered a discrete event. They allow the representation and study of the structure as well as of the dynamic behaviour of systems and processes and have been proven to be a powerful tool for studying system concurrency, sequential, parallel, asynchronous, distributed deterministic or stochastic behaviour, resource allocation, mutual exclusion and conflicts [28,29].

Representation of the dynamic state of the modelled system enables the simulation of certain scenarios as defined from the definition of the initial state of the system and the state change rules (discrete event simulation). Popular fields of Petri nets include production and manufacturing systems, project management, computer networks, software development and engineering, traffic monitoring and control, power systems and robotic tasks. Except the typical engineering applications Petri nets have been used for studying chemical and biochemical processes, medical and healthcare tasks and cognitive, educational and learning procedures.

4.2. Petri Net Theory

In the late 1950s, Carl Adam Petri developed an innovative approach to the theory of the concurrency and synchronisation applied to the modelling and study of concurrent distributed discrete event systems [26]. The introductory work on this new tool was his Ph.D. thesis (1962), titled *Communication with Automata*, and in this, PNs were given a rather qualitative than quantitative notion of time. Since then, Petri nets have changed significantly; a lot of variations, extensions, parameters and features have been added to the initial formalism in order to increase its modelling power, extend its area of application and make it suitable for problems with special features, specifications, structure and needs generally.

The initial formalism, ordinary Petri nets (OPNs), is bipartite directed graphs formally defined as the five-tuple: $PN = \{P, T, I, O, m_0\}$. The respective sets for the two types of nodes are $P = \{p_1, p_2, \dots, p_{np}\}$, which is a finite set of places, and $T = \{t_1, t_2, \dots, t_{nt}\}$, which is a finite set of transitions. $(P \cup T) = V$, where V is the set of vertices and $(P \cap T) = \emptyset$. In Petri nets, places describe conditions (e.g., for control purposes) or resource availability. Transitions represent events or actions and arcs direct connection (that may have weight equal or greater than one), access rights or logical connection between places and transitions. Places are the passive element of the PN, while transitions are the active ones. The element I of PN represents an input function, while O denotes an output

function. Finally, m_0 is PN's initial token distribution referred to in the literature as marking and representing the initial state of the elements of the Petri net. Places, transitions and arcs define the structure of the model and can be used explicitly for modelling purposes.

Places and transitions are connected interchangeably, while T defines node partitions of the network (i.e., nodes of the same type are not linked). The most important PN properties (reachability, safeness, k-boundedness, conflicts, liveness, reversibility, persistency, deadlock-freeness, P -invariants and T -invariants) capture precedence relations and structural interactions between system components [30].

Inclusion of time delays (constant, following certain distributions, or random according to the actions) in the transitions of the initial formalism implements T-timed PNs (TPNs). An analytical presentation of different classes of TPNs (such as bounded and unbounded) can be found in [30]. In the general case, TPNs are defined as $\{P, T, I, O, m_0, D\}$, with the first five elements of the formal definition responding exactly to the same features as in the case of OPNs and D representing time delays that are functions from the set of non-negative real numbers [29,31].

The main structural elements (building blocks) of a Petri net are the following:

- **Discrete Places** (\bigcirc) that describe system states (represent conditions, e.g., for control purposes) or resource availability.
- **Discrete Transitions** (\blacksquare) that represent events or actions that change system states.
- **Tokens** (\bullet) describe the dynamic condition of the system. Tokens reside in places, move through transitions and define the marking of the net. Tokens and their movement through transition firing enable the simulation of the models.
- **Connection Arcs** define relations, such as physical connection, access rights, logical connection or sequence in the performance of events between system components. Three distinct arc types are the most commonly met in the literature:
 - Standard arcs, usually drawn as (\longrightarrow), representing connections between entities, define flows of materials, sequence of events, etc.
 - Inhibitor arcs are represented by arcs whose end is marked with a small circle (\dashrightarrow). Inhibitor arcs disable transitions when input places connected to them contain number of tokens equal to their weight.
 - Test arcs are represented as arcs with dotted line (\dashrightarrow). Test arcs enable certain transitions when input places connected to them contain a number of tokens equal or greater to their weight and allow the firing of such transitions without consumption of the tokens residing in these places.

Simulation is performed through the firing of enabled transitions and causes a change of the system state. This firing represents the performance of an event and “consumes” from each input place the number of tokens equal to the weight of the arc connecting the input place with the transition, while at the same time “producing” to the output places a number of tokens equal to the respective arc weight. Transition firing updates the set of reachable states through the newly enabled transitions. When the set of reachable states becomes empty, a deadlock is reached.

5. Proposed Methodology

5.1. Process Modelling and Simulation Role

In the context of the current research, process simulation model (PSM) denotes a generic model with all the related methods, algorithms, mechanisms, services and tools it directly uses, integrated into an overall modelling application or platform. In any individual model, these methods, algorithms and mechanisms do not change, but the process model (digital) representation per se is adapted to the specific needs and features of the use case. PSM interconnects and interoperates with external AI tools, optimisation tools, analytics tools, etc. These may also change from subclass to subclass. However, PSM itself does not change from application case to case. Figure 1 outlines the communication between the physical system and its digital shadow as well as the procedures taking place in the context of the implemented services.

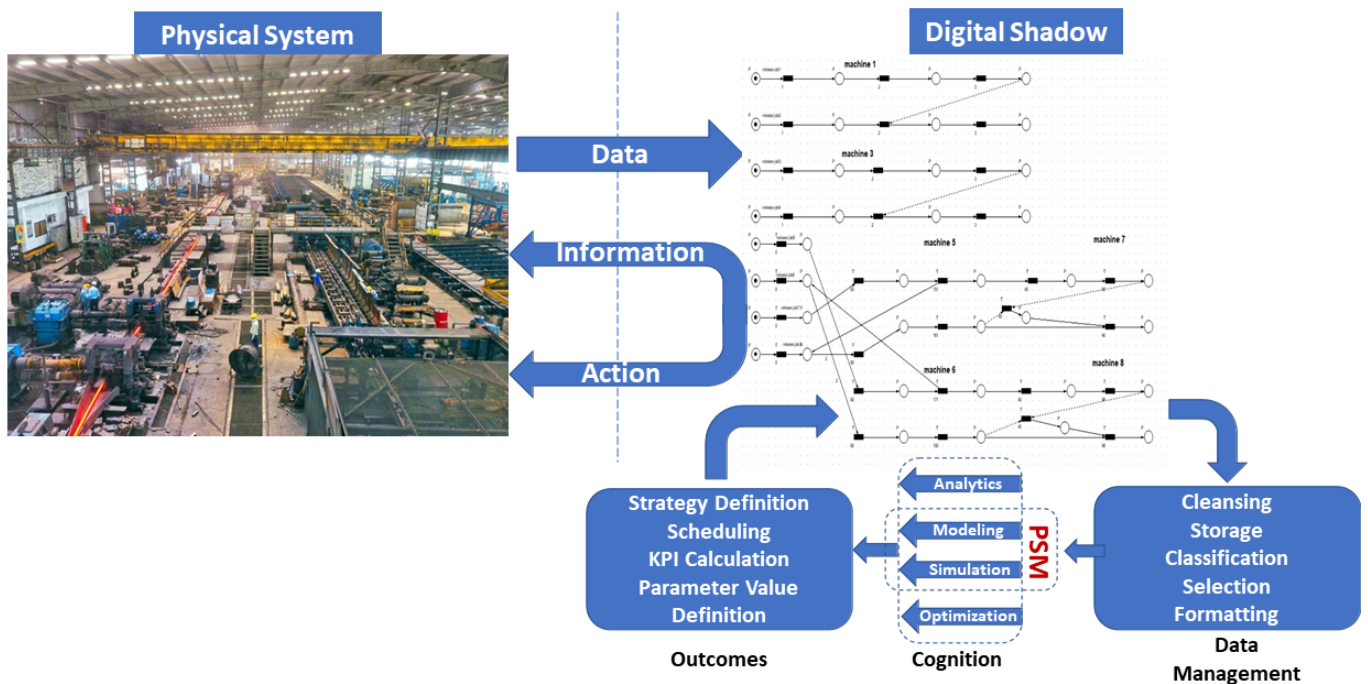


Figure 1. Physical system—digital shadow communication and services interoperation.

Generally speaking, the PSM can be used in conjunction with other tools and methods for predicting and cognitive purposes that serve decision making. Predicting system abilities can be used to support the physical system's dynamic adaptation, e.g., by predicting product demand or resources' availability and matching accordingly production schedule and process parameters (i.e., product supply). An entirely different type of prediction will be to anticipate the next process/system state change(s) (including potential problems), in order to perform some action or notify some operator. The cognitive behavioural enhancements of the model add elements of reasoning capacity, understanding of system knowledge and self-awareness, with the main focus put on the anticipation (and corrective action).

The PSM uses the analytics service to estimate KPIs for assessing (a) the performance of the current system (or single processes), either at regular intervals or on request), and/or (b) the performance of a simulated scenario(s). In addition, analytics provide sequences of events to be simulated and approach the characteristics (delays and appearances) especially of uncontrollable and unpredictable events using statistical and AI tools. Given that the AI tools detect an anomaly, PSM can either verify or refine (or both) this AI inference running a scenario of the current system in isolation from the current shadow twin, in order to test some hypothesis about the anomaly. A simulation may be carried out either (i) in the present to repeat processes with modified parameters, (ii) in the past to retrace the system's/event's history, or (iii) in the future to predict consequences. Regarding the interaction with optimisation service, there is a bidirectional interchange of data, to be used for running both services with common assumptions in order to define the optimal policy, evaluate and cross evaluate its efficiency. In addition, the process model can be used to evaluate a subset of feasible alternative schedules through simulation in order to provide the final selection to the decision makers, especially in the cases that no optimal policy can be analytically defined.

Ideally, the role of the central (production system) modelling component can be defined in relation to the cognition process, as follows:

1. Create a dynamic digital shadow of the physical system, providing an accurate image of its operation at any time, updated by getting real-time monitoring data from the

analytics platform. The dynamic image can be also used by man–machine interfaces for system monitoring, control and management.

2. Accurately model all production processes as well as all entities along the product's value chain and all inputs and outputs, providing a comprehensive systemic view and operational statistics.
3. Use KPIs and business rules and benchmarks to assess the performance of the entire system and each component; also compare estimated and real-time data for self-assessment.
4. Provide systemic knowledge (entities, relations, material flows, process states, and performance) and operational data to the knowledge engineering component (e.g., knowledge graph) and the machine-learning tools, while using the stored knowledge to make the model more intelligent.
5. Provide support for root-cause analysis, risk analysis and hypothesis testing to AI problem inference tools.
6. Provide use case models and data to support optimisation algorithms and tools.
7. Provide base model and support for demand–supply and other prediction tools or in-built routines.
8. Support system adaptability by building, running and assessing system adaptation scenarios.

5.2. Process Modelling and Simulation within the Concept of Digital Twins

The use of process models is typical for engineering system analysis, design, and control development, during preliminary design stages or for performance evaluation and optimisation during the operational phase of a system. Their use is wide, as they enable virtual experiments that provide realistic results when physical experimentation is either too expensive, time consuming, infeasible, or even impossible to conduct [32].

As already mentioned, with the advanced capabilities arising from Industry 4.0, sensors and Internet of Things, process models can be used online as part of a digital twin. In this case, process models can be treated as living models of a system under study that can continually be updated with incoming data from the operating environment (through sensors or data feeds) to monitor and mirror the current status and evaluate its performance [33]. In addition, the projection on the future of the current state of the system process model can be efficient in predicting the future behaviour of a physical system in conjunction with big data management, analytics, artificial intelligence and optimisation techniques and tools using real-time data and certain types of information (such as historical and strategic). Finally, process models acquire an interactive relationship with the physical system, as they can be used for modifying the system's future behaviour. This can occur through the realisation of certain actions and commands to real system components through appropriate actuators installed on it.

This new use of process modelling and simulation aims to aid and improve the efficiency of engineering support, the decision-making process, and evaluation of the impact of production changes by providing fast reaction on time [33]. This procedure in the context of a digital twin is continuous, so there is no meaning in defining a specific scenario or the duration of a simulation. All the above mentioned certify why process models, through the advanced capabilities they acquire in the context of digital twins, become a critical factor and a key trend in the way of the realisation of intelligent manufacturing [34].

5.3. Process Modelling Methodology

5.3.1. Types of Input Data

For the implementation of the model structure and the definition of its parameters, several input data are required. These data are generally of two types: static data, that refer to characteristics of the production system that do not change or are updated periodically, such as machine types, machines, resources, etc.; and dynamic data, that refer to the scenario considered, such as jobs, products, processes, moving and setup times, maintenance

activities, etc. Certain types of data have to be provided to the PSM module by the information system of the production system under study (orders, release dates, time horizon, etc.) and others from modules such as analytics (data connected with breakdowns, setup and processing times) and optimisation (that calculates the schedule or else a heuristic used for this purpose). A superset of types of parameters is necessary in order to construct a model and simulate specific production scenarios (in certain problems, parts of these data can be omitted if events as the transfer of parts are considered of negligible duration or if certain characteristics are studied). This list is specified according to our needs as determined through the interaction with the steel reinforcement installation, and includes the following:

- Flow charts. They are used to study the connection between machines and moving equipment (resource allocation) and create connections between the entities of the system according to this.
- Resources catalogue. It refers to all the resources of the production line as well as the possible use of each one for the production of final products.
- Initial inventory for a case. This refers to the initial levels of internal buffers of the system as well as parts in the machines. This practically is part of the initial marking of the net (initial token distribution) and is used to define the initial state of the system together with the states of the machines.
- Production orders and jobs. In many cases, a production order is composed of different jobs that need to be processed in different machines and produce parts with different characteristics. Product quantities, due dates, priorities between orders, whether they can be split in suborders, product special features and resources used for each should be known.
- Machine families. Machines with similar characteristics that can be used for the performance of certain tasks. In addition, constraints regarding the performance of certain types of processes in machine sets must be included in order not to consider cases that are not applicable.
- Product families. Products belonging to a family may have common characteristics as certain processes received, common setup times, machines used for the performance of specific processes, etc.
- Schedule per machine. According to this, the number and type of different processes performed in each machine is listed in order to define the exact structure of its model. This is received from the optimisation algorithms and can be different for every machine or common for every production line with respect to the specific characteristics of the case considered. The proposed schedule might be different if the objective function changes.
- Machine categories and constraints. The types, multiplicities of machines and the processes performed in each case are fed. In addition, possible constraints that refer to specific machines must be known especially for use during scheduling.
- Process duration. The theoretical delays associated to the assignment of each type of process in each machine are listed. Machines typically are different in their speeds, have constraints in parts they can process and have breakdowns that follow different distributions.
- Maintenance-related data. These activities can be considered to be of two types, the scheduled ones and the non-scheduled ones that refer to machine breakdowns. In both cases, the time between two consecutive events or distribution followed, if such data exist and are not considered negligible, should be known in order to simulate a specific scenario as well as the delay representing the duration of such an event.
- Priorities. The priorities concerning processes performed in each machine when this is not specified by a certain schedule are given.
- Machine setup-related data. The delays representing the machine setup when the type of processes performed have to be known. According to the problem considered, these

delays can be sequence dependent, non-sequence dependent, common for product families or different for each type of process performed.

- Part movement-related data. The delays representing loading/unloading and transfer of parts from buffers to machines and vice versa are fed to the system. According to the case, these data can be part specific or independent of the characteristics of the parts. In the first case, optimisation has to consider these data in detail while in the second one they can be represented as lag times.
- Place capacities. The maximum capacities of places (especially of the ones representing internal buffers) as well as the maximum availability of certain resources are introduced to the system.

5.3.2. Model Synthesis and Operation Procedure

The methodology followed for the modelling of discrete processes is a modular one, as this serves the already mentioned use of hierarchical models. A module is a fundamental subsystem with a set of input and output connection arcs defining interactions with other modules and a set of internal discrete and continuous relations that define the module's internal hybrid state. By using a modular approach, it is possible to simplify the description, qualitative (properties and invariants) and quantitative (mean production cycle, WIP, etc.) analysis of a complicated system by splitting its complexity into small and simple entities that can also be considered and analysed independently of each other [35] and to overcome problems relative to the systems behaviour (e.g., some conflicts types) [36]. In addition, modular representation enables the reuse of the specified models and even the implementation of algorithms for automatic model synthesis with respect to system features [17].

According to the proposed approach, the following parts-levels are considered.

- Material processing level represents the main part of the model. In this, the processes performed in order to transform raw materials in final products are represented. The entities composing the production system, such as machines, moving equipment, etc., are represented, and interactions between them are defined. Additionally, resources of the system are represented, and their use through the system is defined.
- Control level is implemented by adding extra characteristics to the model. Certain quantitative parameters are defined (such as buffer capacities, process conditions, etc.) and ancillary structures are added in the model in order to control its behaviour. These parameters and structures constrain the model's behaviour and limit the state space (possible reachable states during simulation) within given limits. Typically, through the implementation of the ancillary model structure, sets of mutually exclusive resource states are defined, and through the prioritisation of certain actions, the behaviour of the system can be controlled to follow given constraints.
- Decision making is implemented through the definition of quantitative parameters and the addition of auxiliary structures in the model. Such typical decisions concern priorities between the occurrence of certain events when conflicts are met, the performance of the schedule calculated from optimisation tools and resource allocation through time. The use of an inhibitor and test arcs can model the performance of events concerning, for example, change of the system state when buffer levels increase or decrease with respect to a given limit.

The overall Petri net model building procedure is depicted in Figure 2. Currently, it can be used for the synthesis of Petri net models of production systems that can be partially or totally considered production lines or have parallel machines that can perform the same type of processes in the given stages of the production procedure. This procedure typically concerns a number of common concepts that are considered the stages of the proposed methodology. The most important of these are the following:

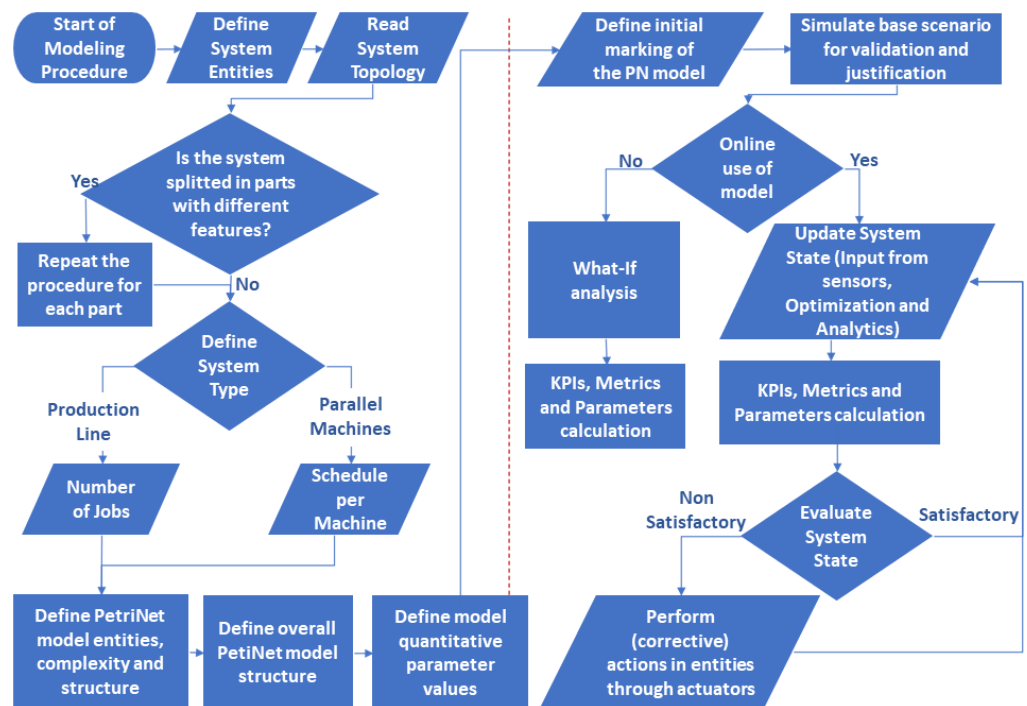


Figure 2. Petri net model building procedure.

- Definition, description and representation of the numbers and types of entities (such as machines, people and equipment) that compose the system under study as well as their position in the operation procedure (stage, position, inputs and outputs, interconnections and process performed). This part is the initial part of the modelling procedure, as it is critical for the definition of the structure of the model. In order to transform the provided flow diagrams in diagrams of components, this part has to be defined accurately. Such information can be acquired in the initial design part of the model and is considered static data that can change only periodically when structural changes in the system take place (probably once or twice a year), as they represent decisions of high cost and complexity (rather strategic ones).
- Representation of system topology. In particular, the way models of the specific entities are interconnected in order to represent the flows of information, materials and other resources in the system is defined according to the topology and the geometry of the real system. The interactions between entities are important for the simulation, as they increase modelling power, controllability, represent realistic cases and through the addition of randomness may lead to different outputs through the reach of different states. This step mainly allocates the structure of the overall system model, using as structural components the individual models of the entities composing it and defines the flows of material and data in the system. Flows of information are critical for the operation of the model as a digital twin, as the different services exchange different types of information in real time.
- Split of the system in subsystems with common production and operation characteristics (e.g., transfer lines, and parallel machines). Each subsystem may consist of multiple entities (for example, process stages) and in this stage, the internal structure of each subsystem is confirmed, and the operational limits of each subsystem are defined. This is an iterative process that splits the overall system in subsystems with common operational characteristics.
- Definition of the industrial system type. Not all subsystems share common characteristics and these characteristics play significant role on the way system is analysed and studied. In the current research, two major types of production systems are considered, transfer lines and parallel machines in production stages. In the case of a

transfer line, scheduling is common for all machines composing it and all parts visit machines mandatorily and with the same sequence. In this case, only the schedule for the first machine has to be provided. When in certain production stages there are parallel machines, that can perform the same operation (with constraints or not), the sub-schedule should be provided per machine (since parallel machines share the parts they have to process). For simplicity reasons, in the current research, production systems can be considered parts (segments) of the above-mentioned types and for each part, the respective strategy will be followed. Generally, the problem can become more complicated when routing flexibility is considered, as in this case, the schedule per machine can be totally individualized.

- Construction of the Petri net models of the individual entities. In order for this to become possible, the complexity and the exact structure of the models has to be defined with respect to the events that occur in the system and changes of its state as well as of their characteristics. For example, the number of different types of processes performed in an entity as well as their sequence has to be defined in this stage. For this reason, the applied policies and strategies, such as the schedules calculated from optimisation algorithms, or rules or heuristic methods, have to be taken into account. Orders should be fed to the respective machines or lines in given sequences. Generally, the events considered can be of different categories, as some of them are fully controllable (e.g., change of process performed in a machine, and machine setup), some are partially controllable (preventive maintenance that should be considered in a given time window), while others are not at all controllable (e.g., machine breakdowns, raw material shortage or external to the system events) and change the state of system not in the direction of improving its performance. Non-controllable events cannot be modelled accurately and are approached using distributions that give partially satisfactory results with respect to previous data.
- Construction of the structure of the overall system Petri net model with respect to the respective models of the entities. This is a typical bottom-up modelling approach. The individual models have to be linked according to the topology represented in stage 2. Connection points are created by the fusion of output buffers of preceding processes with the input buffers of the subsequent processes in common ones.
- Definition of the characteristics of the jobs considered for process. The number of jobs plays a significant role in the complexity of the overall PN model, as different branches are considered for each process with its own quantitative characteristics. In addition, the way that jobs are operated must be defined (by part or by batch) if batches are split or if they are treated as a unity and many more decisions and assumptions concerning the execution and simulation of the model through the performance of events in it.
- Once the structure of the overall model is confirmed, the quantitative parameters of the system have to be added. Typically, such parameters refer to time (delays) and quantities (e.g., resources participating in a process). In the case of controllable events, the time instants at which they will occur are specified (or the delays associated to their occurrence), while in the case of uncontrollable events, other parameters should be specified (e.g., the distribution followed, and consequences in specific entities). Once again, times should be calculated with respect to the way jobs are processed (by part or by batch) in order to have realistic results. In addition, in this step, certain variables that are necessary in order to represent the state of the system as well as to measure certain KPIs have to be defined. KPIs are not always common, as they have to do with areas of interest of the specific application and in many cases ways to calculate them or record them during the simulations must be specified, as this is not straightforward. This part is very important also for the performance evaluation of the system under the scenarios considered. Typical such indicators can be mean idleness time of the machines, percentage of the time a machine is operational, percentage of the time a machine produces, mean number of parts in buffers, maximum tardiness, mean tardiness, energy efficiency connected indicators and many more.

- Representation of the initial state of the system. This refers to the initial inventory levels (raw materials, semi-processed parts and products), initial states of machines, resource availability, as well as to the orders that must be processed. Initial state of the system is very important, as it affects the complexity of the models constructed as well as their execution through simulation, since it defines the initial reachability state set. The set of initial parameter values is referred as a scenario, and the study of alternative scenarios is critical between others in order to forecast the behaviour of the model or even optimize it when no analytical optimisation tools can be considered (by comparative simulation and evaluation of results). In the case of Petri net models, the initial state is defined as a token distribution in the places of the model.
- Initially, simulation of the base scenario and evaluation of the results obtained is performed. Comparison of these results with the realistic ones validates and justifies the practical use of the received model and enables its usage for the study of other cases. The base scenario typically describes a well-known production case whose results can be compared easily with the ones from the simulation. If the results are similar, then the model is considered accurate, validated and verified and can be used for studying alternative scenarios, realistic or planned. As described in [37], validation itself is viewed as a process and evidence for “building the right model”. In addition, verification, which is also a process, is concerned with “building the model right”.
- The further use of the model can be offline for what-if analysis and evaluation of alternatives or online, in the context of a digital twin serving as a digital shadow of the real system. In this case, the simulation is not terminated as the active state of the system is updated through inputs received directly from sensors or indirectly after the application of tools and methods from analytics, optimisation and AI. These inputs are “translated” in tokens generated in certain places of the model. Functionally, in the Petri net model, this is possible through the inclusion of input transitions in the respective parts of the model. In addition, KPIs and measures are calculated that monitor the operational efficiency of the entities or of the whole system. The calculated data are continuously evaluated from decision-making tools, and if they are not considered satisfactory, corrective actions are performed through actuators in the real system. These actions cause more state changes in the composing entities and are also mirrored in the digital twin model. In this case, also simulation can be used to forecast the behaviour of the system with respect to its active state and examine what will happen if certain events take place, in order to evaluate all alternative possibilities and propose an optimal policy.

6. Case Study

A bay of a steel reinforcing installation that refers to one of the pilots of FACTLOG project is considered as a case study. The processes performed involve cutting and shaping various diameters of steel reinforcing bar, using various manual or automatic operations. Three types (or families) of final products can be produced: coils, bars and bent bars of different diameters and with different numbers of bents. Coils and bars are produced after a one-stage process, while bent bars need a two-stage process (first cutting and then bending—there is no flexibility between the two stages). For the previously described processes, four machines are available (in parallel) for coil cutting (M_1 – M_4), two for bar cutting (M_5 – M_6) and three machines (M_7 – M_9) for bar bending. Machines performing a type of process are not all identical per stage, as they have different constraints (e.g., maximum size of raw materials that they can process) and production characteristics (e.g., process and speeds).

Each machine performs strictly processes of one stage, and setup is necessary before changing the operation type performed in it. Moving equipment (cranes) is used for transportation loading and unloading of parts and products between machines and buffers, but the respective delays are considered negligible. The possible flows of the described process are presented in Figure 3 as a three-stage process, where stages 1 and 2 are completely

independent, while the parts receiving the bending process (stage 3) must be already cut to the given dimensions (stage 2). Different types of raw materials are initially available and with respect to the specifications of product characteristics concerning, size, geometry, weight, type of raw material used, etc., can be produced. It is obvious that the production system considered refers to parallel machines and for this case, schedule per machine is necessary in order to construct the *PN* model.

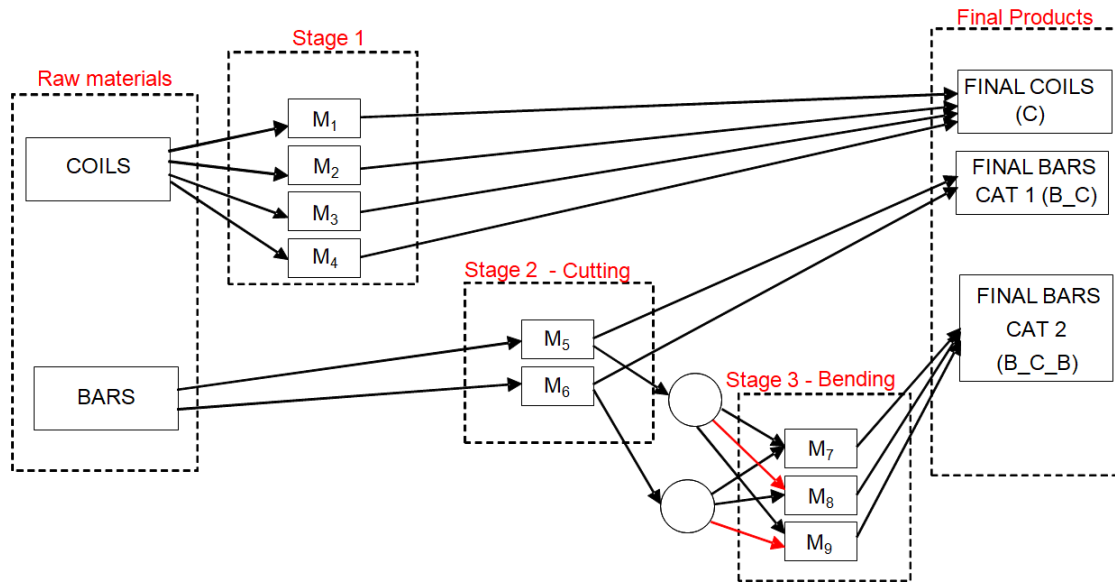


Figure 3. Flowchart of the installation.

The main entities considered are the machines of the three production stages according to the specifications provided. In order to build the respective Petri net models of the machines, the jobs under process must be considered. The orders consist of jobs that refer to batches of products with similar characteristics. The main assumptions regarding process modelling and simulation concern the orders that have to be scheduled, as well as the characteristics (batch sizes, processing times in the alternative machines per stage, setup times, etc.) of all jobs. When a job is scheduled, no preemption–interruption is allowed, and no job can be processed by more than one machine at the same time (i.e., job splitting is not allowed). In addition, machines cannot process multiple jobs at the same time, and if a product is flagged as finished, then it cannot be processed again. Finally, raw material availability in all cases is considered infinite.

The fundamental structural element of the Petri net model of a machine is the model of a process shown in Figure 4. According to this, the model consists of three places and two transitions between them. The initial place (p_1) refers to the available unprocessed parts (resources), (p_2) describes the available parts that are ready for the process after machine setup has finished, while the final one (p_3) refers to products after the performance of the process. First transition refers to the machine setup and the second one to the performance of the batch process.

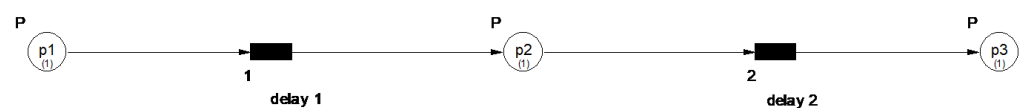


Figure 4. Timed PN model of a process with machine setup.

According to the modelling procedure described, a timed Petri net model for each machine is implemented. In order to do this, the number as well as the sequence of jobs

performed in this machine according to the schedule received must be fed from the optimisation algorithm or scheduling rule used. For each job, the structure of Figure 4 is iterated. In this structure, the sequence in which the different types of processes is performed must be defined. The structure of the Petri net model must enable the implementation of this policy. This concerns two main constraints: first of all, that the sequence specified from optimisation will be followed, and second that the machine will have finished the performance of a job before the setup of the next in the schedule. This is achieved by using a test arc to connect the place representing products after the performance of a process (final place of a process model) to the transition representing the performance of the next process according to the schedule.

In cases of sequential processes, the latter ones are performed after the completion of the initial ones. This means that the output place of a process performed in the first machine will be, in fact, the input place for the setup of the process performed in the second machine. These parts would be the connection points between the individual models of the machines and this is implemented through places fusion (output place of the first and input place of the second model are fused in a common place).

In the case of bar bent processes that comprise the second of two stages (after bar cutting) that occur in different types of machines, the machine setup of machine two begins only when machine two finishes the process of the previous job and at the same time machine one finishes the process of this job (the models of the two machines are interconnected). This is achieved through the use of two input test arcs to the transition representing machine setup for process two in machine two. These two input test arcs to the transition showing machine setup come from the place representing products after the performance of the process in machine one and from the place representing products after the performance of the previous process in machine two.

In cases where related data exist, regarding the distribution of the breakdowns appearances as well as their maintenance connected delays, this can be added to the structure of the Petri net model. The same stands for the case where such data can be fed on the system using sensors. This is done by adding a pair of transitions and places as shown in Figure 5. The transitions refer to the appearance and maintenance of machine breakdowns, while the first place refers to machine being productive and the second one to the machine out of order and under maintenance. The first place (productive machine) is connected through test arcs with all the transition referring to the performance of the different types of machine setups and processes in the machines. The physical meaning of this is that no process can be performed if a breakdown has appeared.

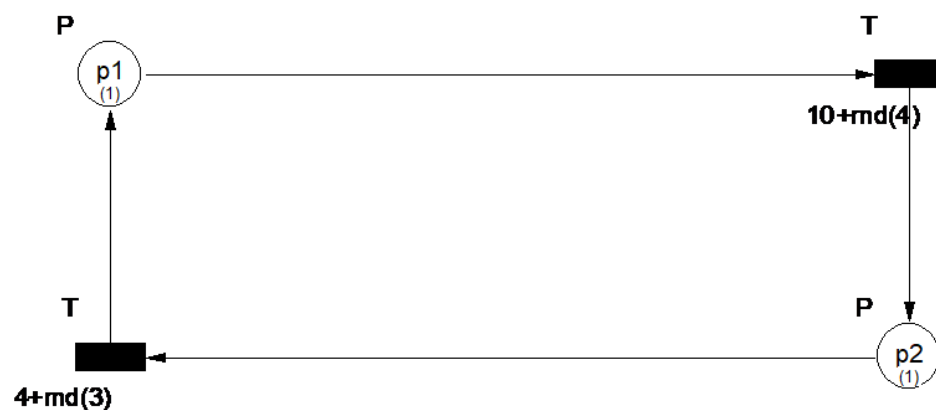


Figure 5. PN structure for machine breakdown and maintenance.

The PN model of a machine performing four different process types on a given sequence (from a schedule) with possible machine breakdowns and maintenance is shown in Figure 6. In this, a test arc is added from the final place of each process (representing the products of the process) to the input transition of the setup for the next process. This

implements in the *PN* structure the sequence in the process performance in the machine, as this is strict and the machine has limited availability and production capabilities.

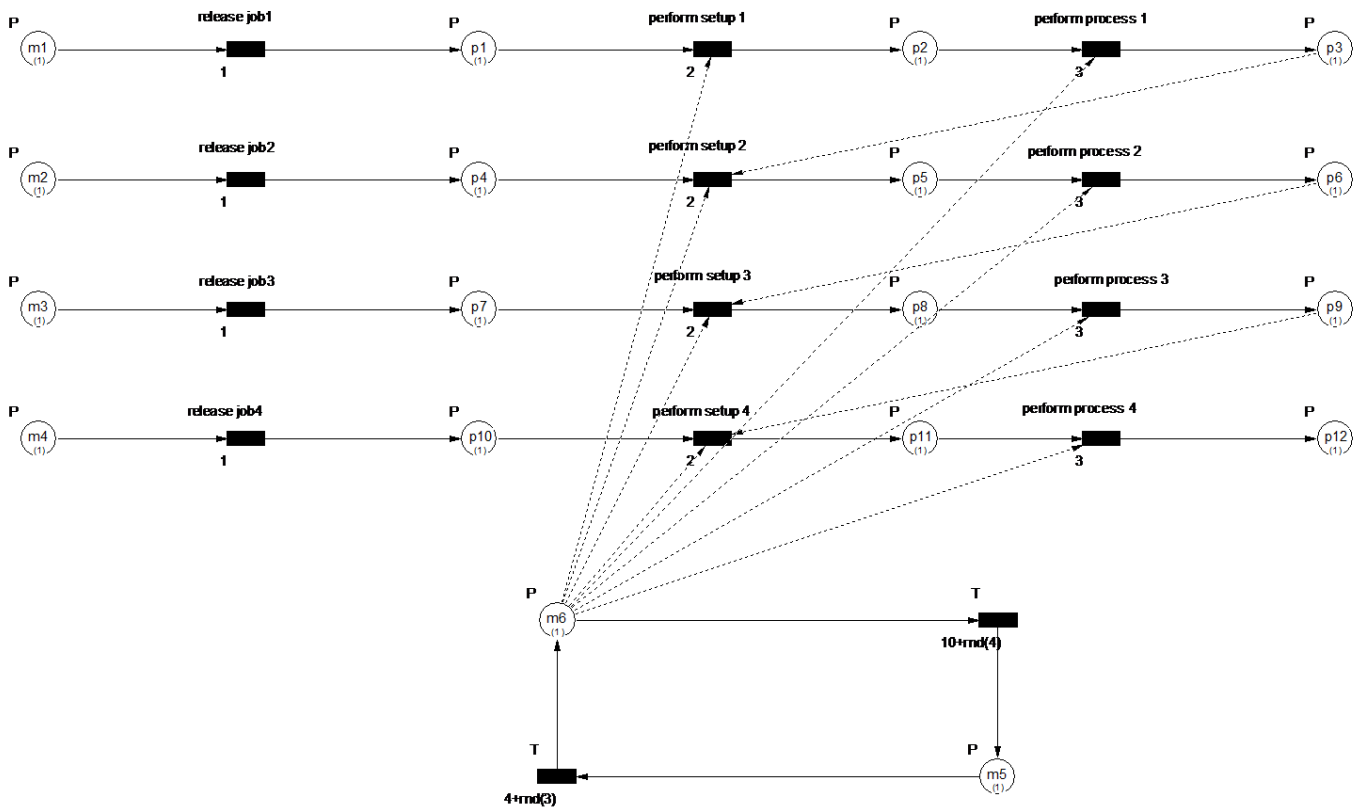


Figure 6. Timed PN model of a machine with breakdowns (unreliable machine) that performs 4 types of process following a defined sequence.

A specific production example is considered. According to this, eight jobs are scheduled and assigned in the subset of machines m_1, m_3, m_5, m_6, m_7 and m_8 in the sequence shown in Table 1. As jobs refer to batches of products, this can be a typical daily schedule in such an installation.

Each machine performs exactly two jobs, while four jobs receive a two-stage process and four jobs, a one-stage process. Six machines are used in total according to the schedule calculated, while three more are not used for many reasons (production characteristics, possible maintenance, etc.) The operation of the machines is considered without breakdowns. Jobs are not split and correspond to the production of batches with given quantitative characteristics. The respective Petri net model of the scenario is presented in Figure 7. It consists of 36 places and 32 transitions and can be used for the production of quantities of eight different types of final parts responding to the orders. The addition of process and setup times can be used for simulating different scenarios. It must be noted that changes to quantities or time parameters do not cause structural changes to the model.

Table 1. Job sequence in each machine.

Jobs Assigned		
Machine	1st Job	2nd Job
machine 1	1	2
machine 3	3	4
machine 5	7	8
machine 6	5	6
machine 7	7	8
machine 8	5	6

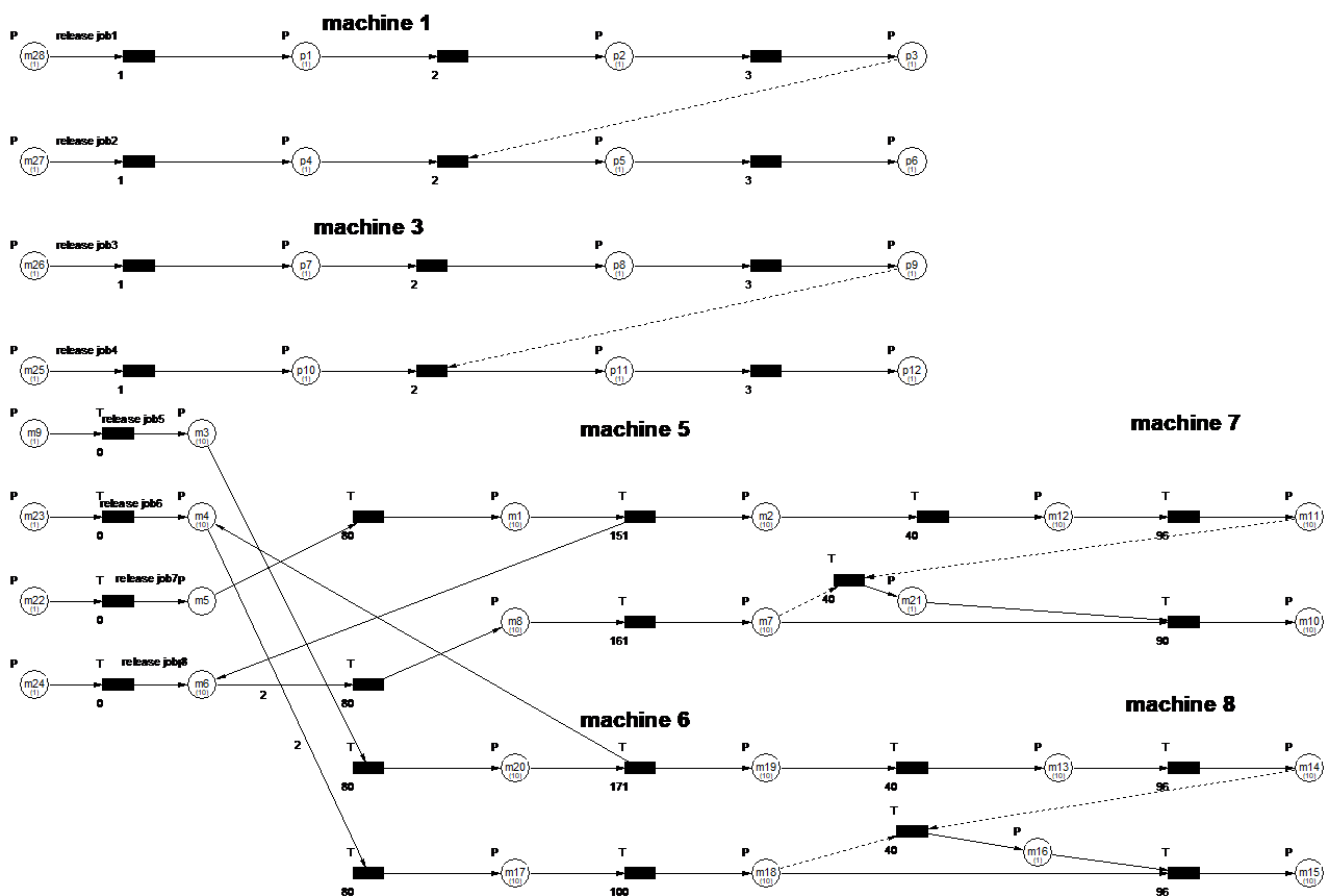


Figure 7. Timed Petri net model of a complicated production scenario in Bay 3.

7. Conclusions

In the current research, a process modelling and simulation framework for discrete industrial systems is proposed. The proposed framework is for general use, is well defined and allows the representation of industrial systems with a variety of different (structural and quantitative) characteristics, following a number of well-defined steps. In these steps, the necessary inputs and outputs are defined as well as the way certain variables are used to implement the models. Initially, the models of the interacting entities of the system are defined and then they are appropriately combined with respect to certain characteristics of the physical system. The proposed method is modular, which means that it can be used to represent systems of increased complexity, using and combining efficiently certain models of the interacting entities of the system.

What makes the proposed methodology innovative and significantly different compared to the previous literature ([38–42]) is that the constructed models, except the typical

use for simulation, performance evaluation and optimisation of industrial systems, can be additionally used in the context of digital twins. In this case, the model mirrors the behaviour of the physical system for monitoring, control, limitation and correction purposes. This, in conjunction with tools and methods from other scientific fields, such as optimisation, analytics and artificial intelligence and the use of sensors and actuators that exploit the potentials arising from Internet of Things, transforms the constructed models in powerful tools with a wide variety of applications.

In the upcoming steps of our research, cases of real systems will be studied in greater detail. This study will concern the comparative evaluation of the behaviour of the system using its digital twin with the current state, the application and evaluation of different scheduling strategies and rules in real time and the calculation of KPIs from a variety of areas. In addition, we plan to perform certain extensions and elucidations of specific features of the proposed method concerning between others, automatic models' construction with reference to certain system inputs (characteristics and parameters), modelling and study of industrial systems with different features from the ones studied here and the use of different levels of detail for modelling and simulation purposes. The technical specifications and considerations for the interaction of process modelling with tools and methods from other areas as analytics and optimisation is also planned. Finally, the extension of the proposed method in order to cover continuous process systems is planned, and the possibility of applying the proposed method in other types of systems with similar characteristics and behaviour with the industrial systems will be examined.

Author Contributions: Conceptualisation, G.T., N.S. and G.A.; methodology: G.T., N.S. and G.A.; software, N.S.; investigation, G.T. and N.S.; resources, G.T., N.S. and G.A.; writing—original draft preparation, G.T. and N.S.; writing—review and editing, G.T., N.S. and G.A.; project administration, G.A.; funding acquisition, G.A. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by European Union's Horizon 2020 program project FACTLOG under grant agreement number H2020-869951.

Data Availability Statement: Publicly not available nor exploitable by other researchers so as to be shared.

Acknowledgments: The authors would like to thank the members of Industrial and Digital Innovations Research Group (INDIGO) of the School of Production Engineering and Management, Technical University of Crete.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

DT	Digital Twin
PN	Petri Net
PLM	Product Lifecycle Management
CPS	Cyber Physical Systems
IoT	Internet of Things
AI	Artificial Intelligence
CPPS	Cyber Physical Production Systems
OME	Observable Manufacturing Elements
ISO	International Standardization Organization
OPN	Ordinary Petri Nets
TPNs	Timed Petri Nets
PSM	Process Simulation Model
NSF	National Science Foundation

References

1. Schuh, G.; Kelzenberg, C.; Wiese, J.; Ochel, T. Data Structure of the Digital Shadow for Systematic Knowledge Management Systems in Single and Small Batch Production. *Procedia CIRP* **2019**, *84*, 1094–1100. [CrossRef]
2. Cameron, I.; Hangos, K.; Perkins, J.; Stephanopoulos, G. *Process Modelling and Model Analysis*; Elsevier Science: Amsterdam, The Netherlands, 2001.
3. Industrial Systems: Capturing Value through Manufacturing. Available online: <https://www.raeng.org.uk/news/news-releases/2012/February/understanding-the-manufacturing-value-chain> (accessed on 29 November 2021).
4. Brooks, R.; Tobias, A. Choosing the best model: Level of detail, complexity, and model performance. *Math. Comput. Model.* **1996**, *24*, 1–14. [CrossRef]
5. Grieves, M. *Digital Twin: Manufacturing Excellence through Virtual Factory Replication*; White Paper; Florida Institute of Technology: Melbourne, FL, USA, 2014; Volume 1, pp. 1–7.
6. Barricelli, B.R.; Casiraghi, E.; Fogli, D. A Survey on Digital Twin: Definitions, Characteristics, Applications, and Design Implications. *IEEE Access* **2019**, *7*, 167653–167671. [CrossRef]
7. Tao, F.; Qi, Q.; Wang, L.; Nee, A. Digital Twins and Cyber-Physical Systems toward Smart Manufacturing and Industry 4.0: Correlation and Comparison. *Engineering* **2019**, *5*, 653–661. [CrossRef]
8. Tay, S.; Te Chuan, L.; Aziati, A.; Ahmad, A.N.A. An Overview of Industry 4.0: Definition, Components, and Government Initiatives. *J. Adv. Res. Dyn. Control Syst.* **2018**, *10*, 14.
9. Büchi, G.; Cugno, M.; Castagnoli, R. Smart factory performance and Industry 4.0. *Technol. Forecast. Soc. Chang.* **2020**, *150*, 119790. [CrossRef]
10. Aoun, A.; Ilinca, A.; Ghandour, M.; Ibrahim, H. A review of Industry 4.0 characteristics and challenges, with potential improvements using blockchain technology. *Comput. Ind. Eng.* **2021**, *162*, 107746. [CrossRef]
11. Maddikunta, P.K.R.; Pham, Q.-V.; B, P.; Deepa, N.; Dev, K.; Gadekallu, T.R.; Ruby, R.; Liyanage, M. Industry 5.0: A survey on enabling technologies and potential applications. *J. Ind. Inf. Integr.* **2021**, 100257. [CrossRef]
12. Tao, F.; Zhang, H.; Liu, A.; Nee, A.Y.C. Digital Twin in Industry: State-of-the-Art. *IEEE Trans. Ind. Inform.* **2019**, *15*, 2405–2415. [CrossRef]
13. Hu, W.; Zhang, T.; Deng, X.; Liu, Z.; Tan, J. Digital twin: A state-of-the-art review of its enabling technologies, applications and challenges. *J. Intell. Manuf. Spec. Equip.* **2021**, *2*, 1–34. [CrossRef]
14. Leng, J.; Wang, D.; Shen, W.; Li, X.; Liu, Q.; Chen, X. Digital twins-based smart manufacturing system design in Industry 4.0: A review. *J. Manuf. Syst.* **2021**, *60*, 119–137. [CrossRef]
15. IPC Releases IPC-2551, International Standard for Digital Twins. Available online: <https://pcb.icconnect007.com/index.php/article/126302/ipc-releases-ipc-2551-international-standard-for-digital-twins/126305/?skin=pcb> (accessed on 29 November 2021).
16. ISO 23247-1:2021—Automation Systems and Integration—Digital Twin Framework for Manufacturing—Part 1: Overview and General Principles. Available online: <https://www.iso.org/standard/75066.html> (accessed on 29 November 2021).
17. Camurri, A.; Minciardi, R.; Paolucci, M. Synthesis procedures for Petri net modelling of a class of manufacturing systems. In Proceedings of the IEEE International Conference on Systems, Man and Cybernetics, San Antonio, TX, USA, 2–5 October 1994; Volume 2, pp. 1989–1994. [CrossRef]
18. Shao, G. *Use Case Scenarios for Digital Twin Implementation Based on ISO 23247*; National Institute of Standards: Gaithersburg, MD, USA, 2021. [CrossRef]
19. NISTIR 8356 (Draft)—Considerations for Digital Twin Technology and Emerging Standards. Available online: <https://csrc.nist.gov/publications/detail/nistir/8356/draft> (accessed on 29 November 2021).
20. Li, H.; Yang, D.; Cao, H.; Ge, W.; Chen, E.; Wen, X.; Li, C. Data-driven hybrid Petri net based energy consumption behaviour modelling for digital twin of energy-efficient manufacturing system. *Energy* **2022**, *239*, 122178. [CrossRef]
21. Friederich, J.; Francis, D.P.; Lazarova-Molnar, S.; Mohamed, N. A framework for data-driven digital twins of smart manufacturing systems. *Comput. Ind.* **2022**, *136*, 103586. [CrossRef]
22. Pommereau, F. SNAKES: A Flexible High-Level Petri Nets Library (Tool Paper). In *Application and Theory of Petri Nets and Concurrency*; Devillers, R., Valmari, A., Eds.; Springer International Publishing: Cham, Switzerland, 2015; pp. 254–265.
23. Ezpeleta, J.; Colom, J. Automatic synthesis of colored Petri nets for the control of FMS. *IEEE Trans. Robot. Autom.* **1997**, *13*, 327–337. [CrossRef]
24. Basile, F.; Caccavale, F.; Chiacchio, P.; Coppola, J.; Marino, A.; Gerbasio, D. Automated synthesis of hybrid Petri net models for robotic cells in the aircraft industry. *Control Eng. Pract.* **2014**, *31*, 35–49. [CrossRef]
25. Letia, T.; Cuiabus, O.; Hulea, M.; Miron, R. Automatic Control Synthesis of Hydro-Power Systems. *IFAC Proc. Vol.* **2013**, *46*, 119–124.
26. Giua, A.; Silva, M. Petri nets and Automatic Control: A historical perspective. *Annu. Rev. Control* **2018**, *45*, 223–239. [CrossRef]
27. Kaid, H.; Dabwan, A.; Al-Ahmari, A. Modeling and Simulation of Queuing Systems Using Stochastic Petri Net and Arena Software: A Case Study. 2018. Available online: <http://www.ieomsociety.org/ieom2018/papers/402.pdf> (accessed on 29 November 2021).
28. Toumodge, S. Applications of Petri Nets in Manufacturing systems; Modeling, Control, and Performance Analysis [Book review]. *IEEE Control Syst. Mag.* **1995**, *15*, 93. [CrossRef]

29. Kuchárik, M.; Balogh, Z. Student Learning Simulation Process with Petri Nets. In *Recent Developments in Intelligent Computing, Communication and Devices*; Patnaik, S., Jain, V., Eds.; Springer: Singapore, 2019; pp. 1115–1124.
30. Zuberek, W. Timed Petri nets definitions, properties, and applications. *Microelectron. Reliab.* **1991**, *31*, 627–644. [[CrossRef](#)]
31. Proth, J.M.; Sauer, N. Scheduling of piecewise constant product flows: A Petri net approach. *Eur. J. Oper. Res.* **1998**, *106*, 45–56. [[CrossRef](#)]
32. Ersal, T.; Fathy, H.K.; Rideout, D.G.; Louca, L.S.; Stein, J.L. A Review of Proper Modeling Techniques. *J. Dyn. Syst. Meas. Control* **2008**, *130*, 061008. Available online: http://xxx.lanl.gov/abs/https://asmedigitalcollection.asme.org/dynamicsystems/article-pdf/130/6/061008/5587121/061008_1.pdf (accessed on 29 November 2021). [[CrossRef](#)]
33. Melesse, T.; Di Pasquale, V.; Riemma, S. Digital Twin models in industrial operations: State-of-the-art and future research directions. *IET Collab. Intell. Manuf.* **2021**, *3*, 37–47. [[CrossRef](#)]
34. Zhao, P.; Liu, J.; Jing, X.; Tang, M.; Sheng, S.; Zhou, H.; Liu, X. The Modeling and Using Strategy for the Digital Twin in Process Planning. *IEEE Access* **2020**, *8*, 41229–41245. [[CrossRef](#)]
35. Thevenon, L.; Marie Flaus, J. Modular Representation of Complex Hybrid Systems: Application to the Simulation of Batch Processes. *Simul. Pract. Theory* **2000**, *8*, 283–306. [[CrossRef](#)]
36. Feldmann, K.; Schnur, C.; Colombo, W. Modularised, distributed real-time control of flexible production cells, using petri nets. *Control Eng. Pract.* **1996**, *4*, 1067–1078. [[CrossRef](#)]
37. Tsiptsias, N.; Tako, A.; Robinson, S. Model validation and testing in simulation: A literature review. In Proceedings of the 5th Student Conference on Operational Research (SCOR 2016), Nottingham, UK, 8–10 April 2016.
38. Khayal, I.S.; Farid, A.M. A Dynamic System Model for Personalized Healthcare Delivery and Managed Individual Health Outcomes. *IEEE Access* **2021**, *9*, 138267–138282. [[CrossRef](#)]
39. Dotoli, M.; Fanti, M.P.; Iacobellis, G.; Mangini, A.M. A First-Order Hybrid Petri Net Model for Supply Chain Management. *IEEE Trans. Autom. Sci. Eng.* **2009**, *6*, 744–758. [[CrossRef](#)]
40. Tang, Y.; Zhou, M.C. A systematic approach to design and operation of disassembly lines. *IEEE Trans. Autom. Sci. Eng.* **2006**, *3*, 324–329. [[CrossRef](#)]
41. Dideban, A.; Zeraatkar, H. Petri Net controller synthesis based on decomposed manufacturing models. *ISA Trans.* **2018**, *77*, 90–99. [[CrossRef](#)]
42. Yu, Z.; Zhou, L.; Ma, Z.; El-Meligy, M.A. Trustworthiness Modeling and Analysis of Cyber-physical Manufacturing Systems. *IEEE Access* **2017**, *5*, 26076–26085. [[CrossRef](#)]