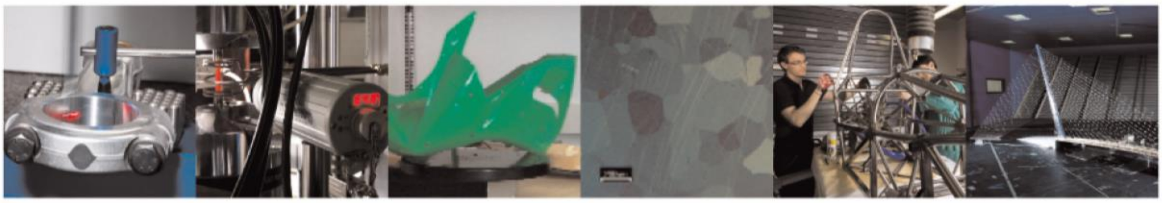


A learning workflow based on an integrated digital toolkit to support education in manufacturing system engineering

Berardinucci F.; Colombo G.; Lorusso M.; Manzini M.; Terkaj W.; Urgo M.

A learning workflow based on an integrated digital toolkit to support education in manufacturing system engineering

Francesco Berardinucci[a,*], Giorgio Colombo[a], Marcello Lorusso[a], Massimo Manzini[c], Walter Terkaj[b], Marcello Urgo[a]

[a] *Politecnico di Milano, Department of Mechanical Engineering, Milan, Italy*
[b] *STIIMA-CNR, Institute of Intelligent Industrial Technologies and Systems for Advanced Manufacturing, National Research Council, Milan, Italy*
[c] *Camelot Biomedical Systems, Genova, Italy*

Abstract

Digital modelling of manufacturing systems is experiencing a fast development, but it still shows significant limitations when considering integration and interoperability of enabling technologies. Indeed, there is still a lack of reference integrated workflows to perform the wide span of tasks, ranging from layout configuration and performance evaluations to 3D representations. Commercial software tools are either too complex or expensive to be approached by non-specialists, therefore it is hard to design effective learning activities in manufacturing system engineering. This paper proposes a structured learning workflow based on an open toolkit that takes advantage of a common ontology-based data model to smoothly integrate digital tools for manufacturing system modelling, performance evaluation, and virtual reality representation. After detailing methodologies and digital tools, the proposed workflow is applied to a pilot case in higher education.

Keywords: Learning workflows, Digital tools, Manufacturing systems engineering

1. Introduction

Designing a manufacturing system is a challenging endeavor involving a wide variety of methodologies and tools addressing different aspects that must be taken into account through a given development workflow [1]. Grounding on this, setting up learning experiences to convey the complexity of this domain and streamline the learning path for engineering students has become a relevant topic within the education community [2, 3]. A significant acceleration in this direction also stems from the ever-changing scenario of new enabling technologies for factories of the future [4] that can be exploited to support innovative learning paradigms based on novel digital technologies [5]. At the same time, the rapid changes occurring in real-world contexts, both in terms of new technologies and business models, impose an update from the standpoint of academic institutions [6].

Currently, even state-of-the-art solutions fail to provide an integrated set of tools that supports basic activities devoted to the design of manufacturing systems in a way that is also suitable for effective learning methodologies. The vast majority of software tools employed in universities enable to effectively tackle specific engineering problems with a very narrow scope, meaning that the level of interoperability across different activities is extremely basic or they are just too difficult and technical to be adopted in a university course. Alternatively, these kinds of functionalities are

[*] Corresponding author.
URL: `francesco.berardinucci@polimi.it` (Francesco Berardinucci)

often included inside large *Product Life-cycle Management* (PLM) suites provided by established commercial ICT companies. Unfortunately, these suites can be typically afforded only by large enterprises, leading to a limited number of potential users, especially among universities and SMEs [7].

This paper describes a framework aimed at designing manufacturing systems that include, within a unified workflow, the fundamental activities to be carried out, together with the intended learning outcomes and digital tools that engineering students can use. The proposed workflow consists of engineering activities that must be performed to address an industrial use case, starting from the formal modelling of a given system and through the evaluation of candidate configurations, and finally to its realistic, interactive representation in a 3D virtual environment. The framework enables to manage information related to products, processes, and production resources within a unified conceptual model that harmonizes knowledge, regardless of the specific tools adopted across the various stages. The framework is supported by an open-source toolkit that provides a consistent advantage in costs and required skills compared to the commercial digital factory tools that are currently available on the market.

The paper is structured as follows. Section 2 provides an overview of currently available studies and research regarding Industrial Engineering Workflows and Learning Factories toolkits. Next, the proposed learning workflow is outlined in Section 3, followed by a methodological description of the activities. Section 4 presents the digital tools related to these activities. A complete use case of an automated assembly line is outlined in Section 5, whereas Section 6 is dedicated to a resume about a collection of feedback from a selected group of Mechanical and Management Engineering students that have been involved in the testing of the workflow.

2. Related Works

The development of an integrated framework for the design of manufacturing systems has a crucial impact on the ongoing digitalization era of the related industry. While this transformation needs to be addressed by the industrial sector, it also poses new challenges in the development of the required digital skills for students in higher education and, specifically, young engineers. The workflow outlined in Section 3 addresses both topics, focusing on the need for a workflow based on integrated digital tools applied to a manufacturing system engineering education environment.

2.1. Industrial engineering workflows

The adoption of new digital tools and the management and processing of data have been increasingly debated since their introduction in the manufacturing industry. The development of frameworks and workflows is needed to integrate engineering data that are related to different aspects of the general problem. This integration can be highly time-consuming, prone to errors, and can raise various technological issues, in particular when dealing with digital tools related to production systems, products, and processes. A possible approach grounds on the concept of Virtual Factory [8, 9] that offers interoperability between heterogeneous digital tools used in an industrial engineering workflow, thus enabling the exchange and reuse of knowledge.

Knowledge-based Engineering (KBE) can typically help to generate automatically a number of candidate solutions for the configuration of a manufacturing system, starting from a set of predefined requirements [10, 11]. Grounding on this, also a 3D environment can be populated and inspected as a VR scene that integrates machine movements and animations.

A crucial aspect of this integration process is the inclusion of knowledge and data formats. While an integrated platform is required, the necessary condition to enable such framework is the definition of a dynamic data model to formalize and exchange engineering knowledge. One possible direct implication of this objective is the possibility to reuse knowledge related to the production system to support dynamic simulations in industrial plants [12].

Cyber-Physical Production Systems (CPPS) are relevant to manufacturing engineering workflows because they support the management of the whole production system lifecycle [13]. The main difficulty is constituted again by the required tools and data exchange formats, thus a wide set of enabling software tool-chains may be needed [14].

An integrated engineering approach can also be beneficial in the analysis and evaluation of system performance, since the generation of alternative configurations can be automated thanks to the digital description of factory entities and processes [15, 16].

2.2. Learning Factories

Learning approaches targeted to manufacturing engineering education are not new in the scientific literature, but the surrounding interest is growing as new digital technologies and tools are made available. Digital environments also have the great advantage of delocalized learning sessions, which is particularly useful when physical meetings or visits are not possible. This is an even more acknowledged issue after the 2020 Covid-19 emergency. A comprehensive review of a wide variety of learning factories approaches was presented by Abele et al. [17], including different kinds of applications for several types of tasks.

There are several notable examples of research studies aimed at generating innovative methodologies and tools for education. ElMaraghy and ElMaraghy [18] proposed a learning factory for the configurations of an assembly system composed of different kinds of machines and stations (iFactory), while taking into account the possible evolution and customization of manufactured products. The learning factory includes interconnected software tools, such as a design innovation studio (iDesign) and a production planning platform (iPlan). .

Learning factory approaches for advanced industrial engineering were proposed to focus the attention on the connection between the digital production planning and the implementation of physical models, while providing the possibility of reconfiguration of the layout through standardized modules [19]. As an example, the IFA learning factory at the University of Hannover [20] tackles the matter of reconfigurability with a particular attention to human-operated processes, meaning ergonomic problems and workplace design, while also providing a virtual representation of the scenario.

Matt et al. [21] proposed the concept of Mini-Factory as a practice-oriented environment to simulate manual or semi-automated assembly processes. On the same topic, Matsas and Vosniakos [22] presented a VR simulation scenario as a serious game addressing advanced devices such as collaborative robots. Salah et al. [23] discussed a learning approach that relies on VR visualization tools for Reconfigurable Manufacturing Systems (RMS), showing higher satisfaction rates from students compared to traditional methodologies. Toivonen et al. [24] proposed the adoption of a methodology based on Digital Twins within Flexible Manufacturing Systems (FMS) to provide students with valuable insights before visiting the real production plant.

Additional examples of a Learning Factory are proposed by Wagner et al. [25], focusing on lean production systems and advanced manufacturing, and by Sivard and Lundholm [26]. Research projects in this area have also been carried out [27] involving both academic and industrial partners, linking together laboratories in universities and the real factories via internet communications.

3. Learning Workflow for Manufacturing Systems Engineering

This section presents the integrated workflow to enhance learning in manufacturing systems engineering. The novel contribution is the seamless integration of the digital tools composing the toolkit that supports the workflow activities. Moreover, these tools are openly available, thus supporting the democratization of the learning approach. The educational dimension is enhanced by tutorials included in an online book[1], where it is possible to find information about the general knowledge base, tools, libraries, and finally a number of examples that implement such methodology, including detailed guidelines to build up use cases such as the one presented in Section 5.

The proposed workflow is represented in Figure 1 as an IDEF0 diagram, focusing on the configuration and reconfiguration of manufacturing systems. After the formalization of manufacturing assets (*Activity A1*, Sect.3.1), candidate configurations are generated (*Activity A2*, Sect.3.2) and their performance is evaluated (*Activity A3*, Sect.3.3). Then, these candidate configurations are compared based on the performance goals (*Activity A4*, Sect.3.4), and the selected ones are further detailed in terms of layout (*Activity A5*, Sect.3.5).

The activities are described in Table 1 together with references to the associated *input*, *output* and *mechanisms*. Input and output data (Table 2) are exchanged between activities, while data are produced or consumed by digital tools that are defined as mechanisms in Table 3. In addition, the proposed workflow explicitly considers the learning objectives associated with each activity in terms of Intended Learning Outcomes (ILOs) [28]. ILOs declare the concepts and skills that a student (trainee) is expected to achieve and be able to demonstrate after the completion of the learning activity [29]. Therefore, each activity in the workflow is associated with specific ILOs, defined as *controls* in Table 4, and related requirements for digital tools.

[1] Virtual Learning Factory Toolkit Gitbook, `https://virtualfactory.gitbook.io/vlft/`

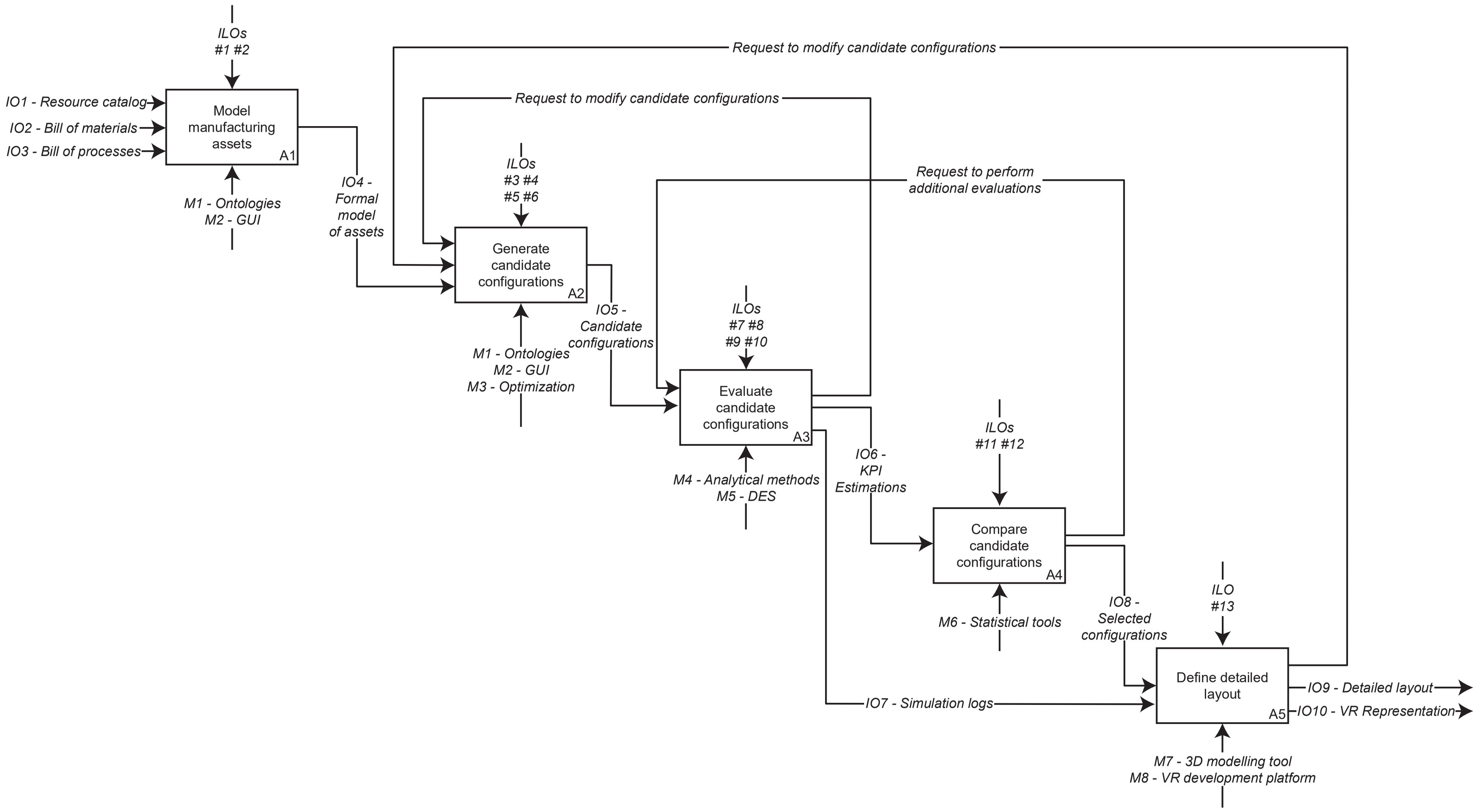


Figure 1: Workflow representation as IDEF0 diagram.

Activity	Input	Output	Mechanism
A1 – Model manufacturing assets The information related to the manufacturing assets (i.e. equipment, fixtures, tools - IO1; products - IO2; processes - IO3) is structured into a formal model (IO4) to support the definition of a manufacturing system configuration. The data formalization takes advantage of an ontology-based factory data model (M1) that supports all the activities in the workflow and ensures interoperability among tools and methodologies. Graphical user interfaces (M2) are exploited to instantiate the factory model.	IO1, IO2, IO3	IO4	M1, M2
A2 – Generate candidate configurations Candidate configurations of a manufacturing system are generated considering the formal model of the manufacturing assets (IO4) obtained from A1. The ontology-based factory data model (M1) supports the formalization of system configurations (IO5) that are generated with Graphical user interfaces (M2) or optimization algorithms (M3) that takes into account the required goals and the constraints related to capabilities, compatibility among pieces of equipment, the feasibility of handling and routing. Multiple configurations can be generated by changing constraints and objective function (M3).	IO4	IO5	M1, M2, M3
A3 – Evaluate candidate configurations The performance of candidate configurations (IO5) is assessed in terms of a set of KPIs (IO6), exploiting performance evaluation approaches, e.g., analytical methods (M4) and discrete event simulation (DES) (M5). The performance evaluation method may also provide detailed output like simulation logs (IO7). A feedback loop towards A2 can trigger modifications of candidate configurations by defining additional constraints to match the target performance.	IO5	IO6, IO7	M4, M5
A4 – Compare candidate configurations The results of performance evaluation activities (A3) are analysed by statistical methods (M6) to assess KPI values (e.g., confidence intervals) (IO6). The performance of alternative configurations is compared to select the best ones (IO8) based on statistical tests (e.g., test for the difference of means, best among *k* solutions, etc.). A feedback loop towards A3 can trigger the request for additional evaluations to increase the statistical significance of the results (e.g., additional simulation replicates).	IO6	IO8	M6
A5 – Define detailed layout The selected candidate configurations (IO8) are further characterised by generating 3D models with CAD tools (M7) and defining a detailed 3D layout (IO9) that can be visualized as a VR representation (IO10) in a VR environment (M8). The dynamic behaviour of the system configurations can be generated from simulation logs (IO7). The system configurations can be interactively refined in the 3D virtual environment (e.g., detailed layout and positioning of assets, the physical dimension of assets, handling and transportation routes). A feedback loop towards A2 can trigger requests for modifications as layout-related constraints (e.g., space constraints, installation requirements).	IO7, IO8	IO9, IO10	M7, M8

Table 1: Description of the activities in the IDEF0 workflow (Figure 1).

The following subsections address the methodologies supporting the activities of the learning workflow and the related ILOs, whereas specific digital tools are presented in Section 4.

3.1. Formal modelling of manufacturing assets

Methodologies and tools supporting the activities in the workflow ask to interoperate through an underlying digital model of the manufacturing system that is continuously updated. Different methodologies and tools take advantage of the underlying factory data model, focusing on a specific view, e.g., logical models for discrete event simulation, physical and geometric models for virtual reality. Semantic interoperability must be enabled by adopting appropriate technologies and

Input/Output	Description
IO1 - Resource catalog	Catalog of production resources (e.g., equipment, fixtures, tools, transporters) that can be selected for the configuration of the manufacturing system.
IO2 - Bill of materials	List of parts, components, assemblies, and sub-assemblies, related to the product type to be manufactured.
IO3 - Bill of processes	List of process steps, composing the manufacturing process plan to be executed, and the associated requirements in terms of equipment, tools, fixture, etc.
IO4 - Formal model of assets	Formal representation of the assets that compose the production system in terms of categories, properties and relations.
IO5 - Candidate configurations	List of alternative system solutions to be evaluated.
IO6 - KPI Estimations	Estimated values of the defined KPIs associated with a candidate configuration.
IO7 - Simulation logs	Logs that are generated during a simulation, e.g. discrete event simulation.
IO8 - Selected configurations	List of selected candidate system solutions.
IO9 - Detailed layout	Detailed layout of the manufacturing system configuration, including geometrical and spatial information.
IO10 - VR Representation	Representation of the manufacturing system configuration in a virtual environment scene.

Table 2: Detailed input and output list in the IDEF0 workflow (Figure 1).

Mechanism	Description
M1 - Ontologies	Formal representation of categories, properties and relations supporting the structured modelling of a manufacturing problem.
M2 - GUI	Graphical user interface supporting the interaction of the users with the digital tools.
M3 - Optimization	Algorithms and methodologies supporting the search for a candidate configuration while satisfying constraints and maximising/minimising an objective function.
M4 - Analytical methods	Performance evaluation approaches for manufacturing systems exploiting an analytical formulation of the dynamic behaviour of the candidate configuration.
M5 - DES	Performance evaluation approaches for manufacturing systems exploiting discrete event simulation (DES) model to mimic the dynamic behaviour of the candidate configuration.
M6 - Statistical tools	Statistical tools supporting the assessment of the values of KPIs grounding on the available results and identifying possible significant performance differences among the candidate configurations.
M7 - 3D modelling tool	Software application to develop 3D models of the assets to be exploited in the VR environment.
M8 - VR development platform	Platform supporting the VR representation of the configuration.

Table 3: Description of the mechanisms in the IDEF0 workflow (Figure 1).

Activity	ILOs	Requirements
A1	• ILO #1: **Understand** the different classes of assets constituting the manufacturing system (products, resources, tools, etc.). • ILO #2: **Understand** the relationships linking the different assets (process steps and precedence constraints, resource requirements and assignments, etc.).	1. Formal representation of knowledge to support conceptual and formal reasoning.
A2	• ILO #3: **Understand** the role of the equipment in the candidate configuration (production resources, transporters, etc.) • ILO #4: **Analyse** the capability provided by the candidate configuration. • ILO #5: **Create** a candidate configuration for the manufacturing system by selecting the pieces of equipment to be installed. • ILO #6: **Create** a structured description of the candidate configuration by defining the relations and roles of the selected pieces of equipment.	1. Easily create a structured formal representation of the candidate configurations. 2. Query structured data (e.g., select the pieces of equipment with a given capability) 3. Reuse/modify formalized pieces of information (e.g., a sub-system). 4. Deal with real-size industrial problems.
A3	• ILO #7: **Identify** relevant KPIs for the candidate system configuration. • ILO #8: **Identify** proper modelling hypothesis for the system under study. • ILO #9: **Identify** the level of modelling details needed for the analysis • ILO #10: **Assess** the selected KPIs through performance evaluation approaches and evaluate the results	1. Take advantage of a clear representation of the candidate configuration to be analysed. 2. Provide a toolbox of alternative performance evaluation methods. 3. Support the matching between modelling hypotheses and the solution.
A4	• ILO #11: **Analyse** the results to derive an estimation of the KPIs. • ILO #12: **Compare** the different candidate configurations considering the estimated KPIs.	1. Link the desired level of details with the formal representation of the manufacturing system. 2. Select of the more efficient modelling strategies and/or the automatic generation of simulation models. 3. Enable the application to real-size industrial cases.
A5	• ILO #13: **Sketch** a detailed layout of the selected system configuration.	1. Define solutions with realistic details. 2. Experience state-of-the-art digital technologies.

Table 4: ILOs for activities A1-A5

technical standards for communication protocols and data exchange formats. Data modelling is a key element to support the interoperability among different digital tools, providing the capability to safely retrieve and store information and make them available for the digital tools, covering and integrating heterogeneous knowledge domains, and guaranteeing extensibility. Semantic Web technologies [30] and in particular ontologies can be employed to meet these requirements [31].

Activity A1 exploits a reference factory data model (*M1*, Sect.4.1) and graphical user interfaces (*M2*, Sect.4.1) to formalize information and knowledge related to:

- production resources (e.g., machine tools, pallets, fixtures, transporters) that can be installed in the production system. These data can be retrieved from catalogues of technology providers (*IO1*);

- part types to be processed together with their characteristics (components, assembly structure, 3d models) and requirements, e.g., production volumes (*IO2*);
- process plans and manufacturing operations, including processing times and resource requirements (*IO3*).

The availability of a structured and comprehensive factory model provides students the opportunity of exploiting formal approaches to deal with complex engineering problems, investigating and understanding the characteristics of assets (*ILO #1*) and inter-relations (*ILO #2*) in a production system.

3.2. Generate candidate configurations

The generation of a system configuration (*Activity A2*) must comply with a given set of requirements (e.g., budget, expected performance, layout constraints, etc.) while exploiting the available and selectable assets defined during the modelling phase (*IO4*).

The factory data model and graphical user interfaces (*M1, M2*, Sect.4.1) are also exploited to formalize system configurations in terms of:

- resources composing the system (e.g., machine tools, buffers, pallets, fixtures, transporters) with corresponding properties (e.g., buffer capacity) and position in the layout;
- relations among production resources and other factory elements, in particular connections (e.g., defining a feasible route between a machine tool and transporters) and aggregations;
- assignments of manufacturing operations to production resources (e.g., a drilling operation assigned to a machining centre);
- control policies governing manufacturing execution [32].

Multiple candidate system configurations can be considered throughout the workflow, according to the range of options and the strategy of the system designer. Differences among configurations can be minor (e.g., capacity of interoperational buffers, selection of a different machine type for a station) or major, in the case of different system architectures (e.g., flexible production line or flexible manufacturing system). Candidate configurations can be generated according to different approaches:

- experience-based design of a single system configuration that is iteratively revised based on feedback provided by end-users and performance evaluation methods (3.3), following a typical trial-error approach;
- extensive search of the solution space, possibly exploiting optimization algorithms and AI techniques (*M3*).

The selection and analysis of the requirements and the equipment for the system configuration enable students to learn how to define relationships between assets, understand their roles (*ILO #3*), and analyse their capability (*ILO #4*), thus building a solid understanding of the system structure. Upon completion of the activity, students are expected to be able to create a configuration of the manufacturing system, selecting the required pieces of equipment (*ILO #5*), detailing relations and roles of the selected assets (*ILO #6*).

3.3. Evaluate candidate configurations

Performance evaluation methods are used to assess the performance (*Activity A3*) of candidate configurations (*IO5*) with respect to a set of relevant KPIs (*IO6*).

Specifically, two types of performance evaluation approaches are considered, i.e., analytical methods and discrete event simulation (DES). Analytical methods (*M4*) ground on a mathematical model of the dynamic behaviour of a manufacturing system. Analytical methods can be extremely fast in terms of computation, but they typically ground on a set of hypotheses that could be constraining. Thus, they are useful to operate a first skimming of candidate configurations. Moreover, they provide students the opportunity to evaluate the relevance and importance of modelling hypotheses with respect to the system under study.

DES approaches (*M5*, Sect.4.2) are used to support a more detailed evaluation of the candidate system configurations while modelling the impact of uncertain factors. DES represents the behaviour of manufacturing systems as a sequence of events generated and managed by a computer to run experiments. In addition, the log of simulated events can be typically exported (*IO7*). By enabling the modelling of a (virtually) unbounded degree of details, DES provides students the opportunity to understand the importance and impact of different levels of detail while matching modelling and design hypotheses to the specific design objectives.

The results of performance evaluation approaches support the estimation of KPIs (*IO6*) related to production performance and operating cost. Common KPIs used in manufacturing systems are throughput (parts/time), lead times, buffer levels, work in progress (WIP), utilisation of resources.

Students are expected to identify relevant KPIs to perform the evaluation (*ILO #7*) and identify proper modelling hypothesis for the system along with the required level of detail of the analysis (*ILOs #8, #9*) and assess the selected KPIs through performance evaluation methods (*ILO #10*).

3.4. Compare candidate configurations

Due to the presence of non-deterministic factors in the modelling of manufacturing systems (e.g., failures, variable processing times, etc.), statistical tools (*M6*, Sect.4.2) are necessary to properly assess the performance of a candidate system configuration (*IO5*) by providing point estimations and confidence intervals for the KPIs (*ILO #11*).

Statistical tools also serve for the selection of the most promising set of configurations (IO8) and quantitative comparison of alternative options (*Activity A4*).

Students will be able to analyse and estimate the KPIs (*ILO #11*) and then compare the candidate configurations based on these KPIs (*ILO #12*).

3.5. Detailed layout definition

The detailed layout definition (*Activity A5*) of candidate configurations (*IO8*) can be supported by a realistic visualization of the manufacturing system that condensates all the design decisions taken in the previous activities. The modelling of a system configuration within a virtual environment (*M8*, Sect.4.3) provides students with the possibility to see the results in a digital factory scenario that can be freely navigated, inspected, and modified with a higher level of detail, supporting a comprehensive validation of the selected solutions.

The generation of a 3D representation of a manufacturing system is typically a demanding task in terms of time and effort, due to the consistent amount of manual activities to be carried out, including the development of 3D models (*M7*, Sect.4.3). An integrated toolkit (Section 4) enables the adoption of advanced virtual reality technologies in a relatively simple workflow. This entails the possibility to use different software tools and rendering engines, by relying on neutral formats to represent assets. Beside basic navigation, advanced interaction features can be developed, e.g.:

- representation of the dynamic behaviour of the manufacturing system in the VR environment based on the results of simulation analyses.
- generation of overlaying additional visual information (i.e., lines showing the routes of moving assets like parts and transporters).
- import and manipulate 3D assets from a library (e.g., modifying the position and orientation) to fine-tune the layout of a candidate configuration.
- show properties and states for specific components (e.g., failure modes for machines).

The VR environment can be exploited as an integrated 3D design environment (*ILO #13*) and support both traditional (monitor, keyboard, mouse) and immersive VR interfaces (e.g.,through head-mounted displays).

4. Digital Tools

We propose to support the workflow presented in Section 3 by means of a Virtual Learning Factory Toolkit (VLFT) that consists of a set of integrated digital tools for engineering learning and teaching purposes [33]. These tools can be integrated thanks to the interoperability provided by the underlying common factory data model (Section 3.1). Given the didactic goal and the need to support an integrated workflow, the following requirements can be identified for the digital tools to be included in the VLFT:

R1 The digital tool offers a way to access and modify (if needed) its internal data structures, by means of exchange files or through an application programming interface (API). Therefore, a customised software component (a.k.a. plugin, connector) can be developed to import/export data considering both the reference factory data model and the specific data model adopted by the digital tool.

R2 The tool must be user-friendly, minimising the time needed to exploit its basic functionalities.

R3 The tool must be free to use to reach the largest set of potential users, thus allowing the democratisation of teaching.

In particular, the following categories of digital tools are needed to support the Configuration and Reconfiguration workflow:

- Modelling of assets and system configurations. Graphical User Interfaces (GUIs) and queries can support the definition and retrieval of production system configurations (i.e. part types, process plans, operations, capabilities of the production resources like machine tools, transporters, storage systems) and related estimates of key performance indicators (KPIs) like throughput, average inventory, average lead time (Section 4.1).
- Performance evaluation via Analytical Methods and Discrete Event Simulation. Commercial off-the-shelf tools and academic tools can be integrated into the workflow by developing plugins that support the automatic generation of simulation models and the automatic retrieval and storage of results to be exploited by other users and applications (Section 4.2).
- 3D visualization and interaction with production systems and production resources by means of Virtual Reality. This tool supports the design, reconfiguration, training and maintenance of production systems and resources (Section 4.3).

4.1. Ontology-based Modelling of System Configurations

The modelling of factory data is based on a Digital Factory approach [34] and relies on a common and extensible data model for the representation of factory assets related to production systems, resources, processes and products. This reference factory data model[2], already presented in previous works [35, 32], has been developed as an OWL ontology [36] since this language provides a way to generate flexible data model integrating different knowledge domains while reusing already existing technical standards (e.g. Industry Foundation Classes [37], W3C SSN/SOSA [38], UML Statechart [39]).

Several digital tools, including commercial ones, can be identified to support the modelling and design of manufacturing system configurations. However, most of them fail to meet the requirements, either because a commercial license must be paid (requirement *R3*) or the management of input/output data exchange is cumbersome (requirement *R1*), or excessively complex to use (requirement *R2*). Therefore, the prototype digital tool OntoGui[3] [40] has been chosen, considering the trade-off among requirements. OntoGui is a digital tool providing a graphical user interface to support the rapid instantiation of ontologies, also providing modules that support the configuration of manufacturing systems. In particular, the following four modules are relevant in the scope of the addressed workflow:

- *System Design* supports the design of a production system and in particular the definition of part types, process plans, and the elements of the system. The process plans can be decomposed into process steps characterised by processing time and precedence relations. Process steps can be assigned to the elements of the production systems acting as resources. The elements in the systems can be characterised by connections and specific properties, e.g. maximum capacity for buffers and failure modes for machine tools.

- *System Control* is dedicated to the definition of control policies and their relations with the manufacturing system in terms of observed variables (e.g. buffer level) and controlled variables (e.g. release of a part to be loaded on a machine tool)[32].

- *Performance Evaluation* supports the definition of a production plan in batches of demanded parts over a time interval by specifying the chosen process plan for each batch, the available manufacturing system, and the selected control policies. The resulting production plan becomes the input of performance evaluation methods to estimate the behaviour of the system in specific conditions.

- *Utilities* provides functionalities to generate reports and exchange files based on the contents of the currently loaded ontology modules. In particular, this module can make elaborations to export the ontology as .json files[4] representing a 3D scene and an associated 3D animation (see Sect.4.3). Similarly, an ontology can be instantiated by importing a .json file that, in turn, can be generated from a spreadsheet table[5].

[2] `https://virtualfactory.gitbook.io/vlft/kb/fdm`

[3] `https://virtualfactory.gitbook.io/vlft/tools/ontogui`

[4] `https://virtualfactory.gitbook.io/vlft/kb/instantiation/assets/json`

[5] `https://virtualfactory.gitbook.io/vlft/kb/instantiation/assets/spreadsheet`

4.2. Performance Evaluation

Several digital tools are available to simulate and evaluate the performance of manufacturing systems. In particular, while many commercial and free DES simulators exist, it is typically difficult to meet requirement *R1*, thus jeopardizing the integration in the workflow. Indeed, an ideal tool should provide an API or an interchangeable format to generate the simulation model in a complete/partial automated way, grounding on the available knowledge on the manufacturing system under study. Furthermore, the simulation results must be easily accessible to assess the associated performance and complete the digital model of the candidate configurations.

Among the available tools, focusing the attention on open source projects (requirement *R3*), the JMT (Java Modelling Tools) suite was selected. JMT [41] offers a comprehensive framework for performance evaluation, including the JSIM module, i.e., a discrete-event simulation tool for the analysis of queuing networks. The underlying model is defined through an XML inter-exchange document and can be generated using an embedded GUI (requirement *R2*). The simulation results, containing performance estimations for the considered system configuration, are also incorporated in the XML document. This feature enables the generation of the model and the collection and analysis of the simulation results through specific digital tools. JSIM was integrated into the workflow by developing jsimIO[6], an additional piece of software based on a Python library able to generate JMT simulation models automatically starting from the available knowledge on the manufacturing system (requirement *R2*). The functionalities provided by JMT also cover statistical tools for the assessment of KPIs from the results of simulation experiments to evaluate and compare the candidate system configurations in terms of throughput of the system and the single stations, response times, average buffer levels and utilization of the stations. Furthermore, JMT enables the definition of virtual *logger* stations that can output the timestamps of the part flows in the system according to the desired requirements and level of details. Therefore, it is possible to reconstruct the flows of parts in the system and use them to feed the generation of 3D animations (Sect.4.3).

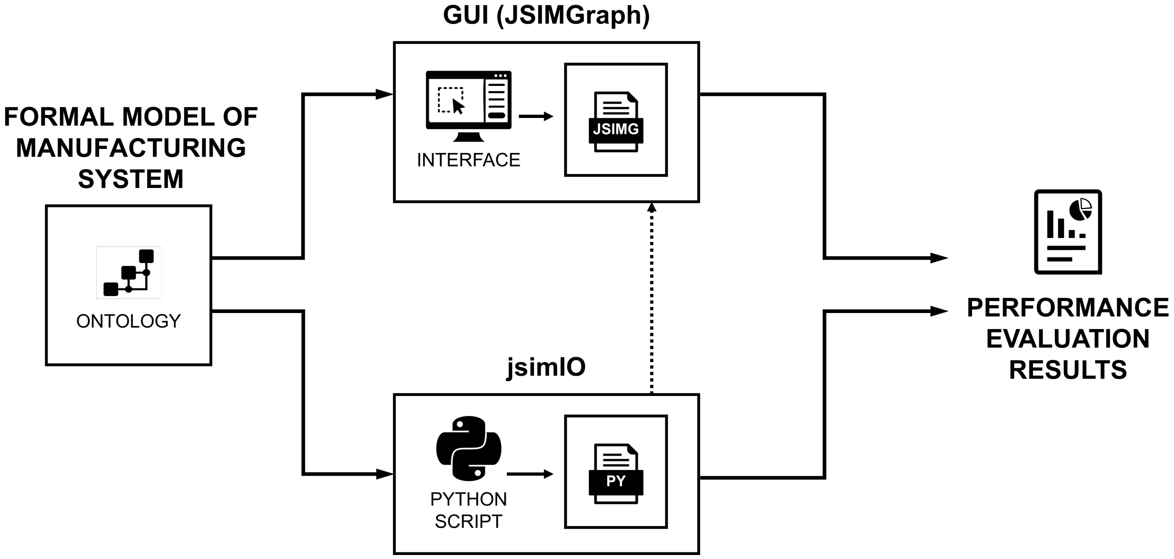

GUI (JSIMGraph)
FORMAL MODEL OF MANUFACTURING SYSTEM
INTERFACE
JSIMG
ONTOLOGY
jsimIO
PYTHON SCRIPT
PY
PERFORMANCE EVALUATION RESULTS

Figure 2: Alternative workflows for the generation and execution of simulations models using JSIM (JMT) and jsimIO.

[6] `https://virtualfactory.gitbook.io/vlft/tools/jsimio`

4.3. 3D Virtual Environment

Many frameworks and libraries are available to develop VR applications (e.g. Unity, UnrealEngine, Amazon Sumerian, Three.js, Babylon.js, etc.). Herein, the prototype web application VEB.js (Virtual Environment based on Babylon.js)[7] was chosen because it offers a reconfigurable model-driven virtual environment based on Babylon.js[8], a complete JavaScript framework and graphics engine for building 3D applications with HTML5 and WebGL (Web Graphics Library). VEB.js provides high-level functionalities that easily enable to load and draw 3D assets, manage these 3D assets in terms of position and orientation, play animations, and generate screenshots (requirement *R2*).

The underlying Babylon.js library gives several advantages because it is free to use, thus compelling didactic scenarios. In addition, a web application can seamlessly work on any platform regardless of the specific operating system as long as it runs on a WebGL compatible web browser (requirement *R3*).

VEB.js can import 3D scenes and animations defined in .json files according to a predefined schema[9] (requirement *R1*). After a system reconfiguration, the updated scene can be exported to a similarly structured .json file. Third-party tools can be exploited to generate 3D scenes and animations as .json files automatically (e.g. OntoGui, see Sect.4.1).

Another relevant input of any VR application is the 3D representation of each asset that populates the virtual environment. Typically, source CAD files (e.g., .step, .iges) are exported in a compatible exchange format (e.g., .obj, .glTF). Herein, the gLTF format was chosen because it is a rapidly spreading standard that enhances the realism of 3D models with visual details in the form of simulated materials and textures. In particular, two options are available:

- *glTF + basic materials*, i.e. basic materials are assigned to asset components directly inside the CAD environment where they have been modelled to reduce the file size.

- *glTF + PBR materials*, i.e. multi-texture material sets are assigned to asset components by using specialized software tools that support the physically based rendering (PBR) pipeline (e.g., Blender[10]).

A simplified outline of workflow to launch VEB.js is represented in Figure 3. The 3D assets (e.g., workstations, conveyors), modeled as .gLTF files, are placed in the layout of the production system according to the definitions in the .json file derived from an ontology. Generic VR environments are not able to handle specific geometric constraints between the assets (i.e., nodes in the scene graph), therefore only positional data in the form of spacial coordinates and parent/child relationships are typically allowed.

Babylon.js provides functionalities to integrate advanced display and interaction techniques to improve the quality of the experience. In this way, a walkthrough around the system can be performed by means of traditional display systems (i.e., a regular PC workstation) or through dedicated devices such as Head-Mounted Displays (HMDs) for a more immersive approach. Aside from basic inspection operations, additional features can be enabled, like interactive manipulations for selected items or animations playback.

[7] `https://virtualfactory.gitbook.io/vlft/tools/vebjs`

[8] `https://www.babylonjs.com/`

[9] `https://virtualfactory.gitbook.io/vlft/tools/vebjs/input-output`

[10] `https://www.blender.org/`

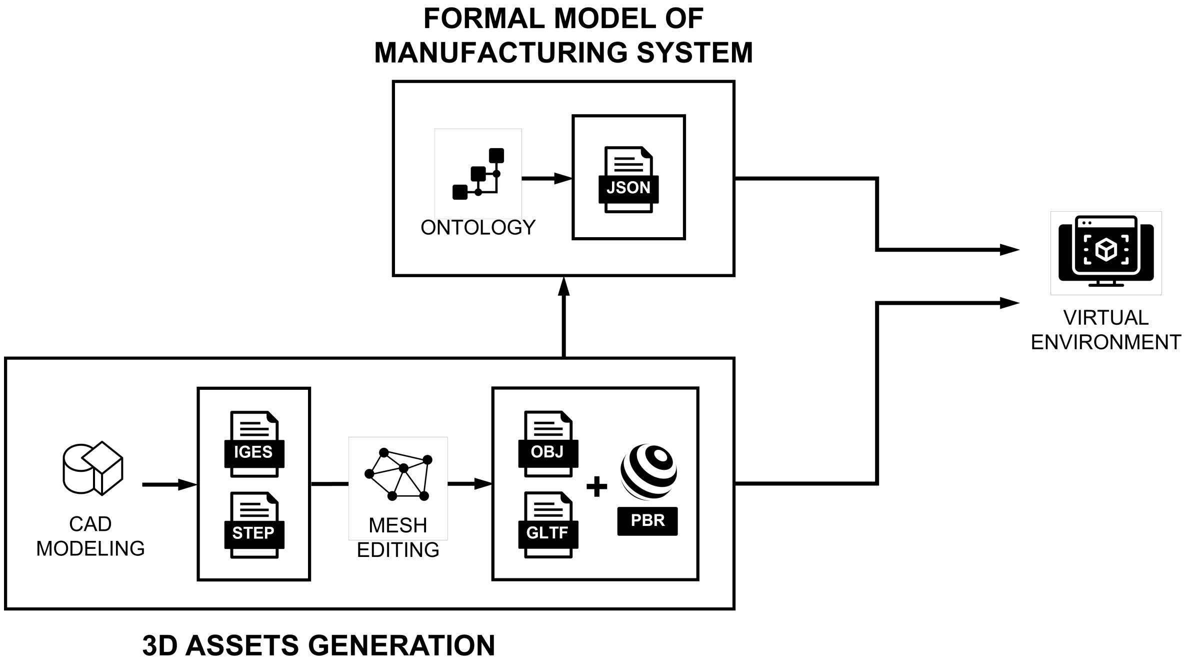


Figure 3: Workflow for the generation of the 3D Virtual Environment.

5. Application of the workflow to support learning activities

This section presents the application of the proposed learning workflow to an industrial case that consists of an assembly line producing self-closing concealed cabinet hinges. After introducing the industrial case (Sect.5.1), this section describes how to instantiate factory assets (Sect.5.2) and the whole assembly line (Sect.5.3), and how to evaluate its performance (Sect.5.4).

5.1. Use-case presentation

The assembly line[11] consists of 19 workstations where specific operations take place (e.g., pick and place, screw tightening, riveting, etc.) to assemble the components of the self-closing concealed cabinet hinge[12]. The components are represented in Figure 4 and listed in Table 5.

Table 6 reports the sequence of operations, specifying which is the input component (cf. Table 5) and the workstation executing the operation. The assembly line can be divided into two sections. The work-in-progress hinges are moved by a rotating table until the workstation *PPH1* (upstream section), whereas the hinges are placed on pallets that are moved along linear conveyors starting from the workstation *RPP1* (downstream section). In addition, four inter-operational buffers are placed in the downstream area.

5.2. Instantiation of Factory Assets

The instantiation of factory assets[13] (both part types and workstations) consists of the characterisation of each asset in terms of geometry, properties, and animations. Taking as a reference

[11] https://virtualfactory.gitbook.io/vlft/use-cases/assembly-line

[12] https://virtualfactory.gitbook.io/vlft/use-cases/factory-assets/assembled-product

[13] https://virtualfactory.gitbook.io/vlft/kb/instantiation

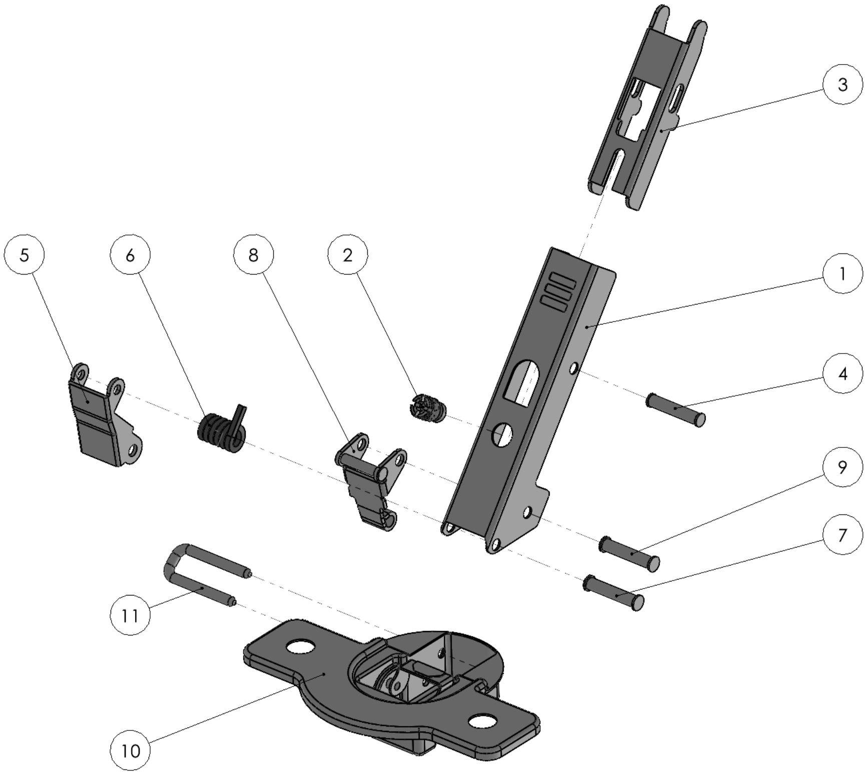


Figure 4: Hinge components with labels defined in Table 5.

Component ID	**Label**
Wing	1
WingScrew	2
Clip	3
Pin1-1	4
Connector1	5
Spring	6
Pin1-2	7
Connector2	8
Pin1-3	9
Box	10
Hook	11

Table 5: Bill of materials of the Hinge.

Operation		Input Component	Station ID	Description
ID	**type**			
1	pick&place	Wing	PPW1	*Wing* is picked from the upstream buffer and placed on the rotating table.
2	tightening	WingScrew	T1	*WingScrew* is aligned to the corresponding hole on the *Wing* and screwed.
3	pick&place		PPH1	The wip hinge is taken from the rotating table and placed on a conveyor.
4	pick&place		RPP1	A robot of workstation *RPP1* picks the wip hinge from a conveyor and places it on a pallet.
5	pick&place	Clip	PP1	*Clip* is assembled on *Wing*, hooking to the *WingScrew*.
6	pin insertion	Pin1-1	PI1	Pin1-1 is inserted to fix Clip on Wing.
7	riveting		R1	*Pin1-1* is riveted.
8	pick&place	Connector1	PP2	*Connector1* is assembled on *Wing*.
9	pick&place	Spring	PP3	Spring is assembled on *Wing*.
10	pin insertion	Pin1-2	PI2	*Pin1-2* is inserted to fix *Connector1* and *Spring* on *Wing*.
11	riveting		R2	*Pin1-2* is riveted.
12	pick&place	Connector2	PP4	*Connector2* is assembled on *Wing*.
13	pin insertion	Pin1-3	PI3	*Pin1-3* is inserted to fix *Connector2* on *Wing*.
14	riveting		R3	*Pin1-3* is riveted.
15	pick&place	Box	PP5	*Box* is assembled on *Wing*.
16	pin insertion	Hook	PI4	*Hook* is inserted to fix Box on *Wing*.
17	riveting		R4	*Hook* is riveted.
18	inspection		C1	The finished hinge is visually inspected by a camera.
19	pick&place		CB1	The finished hinge is picked from the pallet and placed in a box.

Table 6: Operations and workstations.

example a pin insertion station[14] (e.g., workstation *PI1* in Table 6) shown in Figure 5, the following steps were carried out:

- Generation of the 3D model for VR environments, starting from a standard 3D CAD model (Sect.4.3). The result for *PinInsertion_Station* is shown in Figure 6 in the both versions with basic materials and PBR materials.

- The hierarchical structure of the 3D model is analysed and explicitly represented in the digital model (Sect.4.1) till the level of detail that is required for the following elaborations (e.g., animations). Specific properties (e.g., position, dimensions, descriptions, etc.) are defined in compliance with the structure of the data model. Typically, the position of a component is defined as relative to the position of its parent. As an example, the main components of the pin insertion station (*PinInsertion_Station*) are listed in Table 7.

- Animations associated with specific states of the asset are defined at component level. For instance, a trajectory can be specified in terms of discrete steps to be interpolated. As an example, Table 8 shows the animation steps for component *PIs_FloatingX* when *PinInsertion_Station* enters the state of *working*.

[14] https://virtualfactory.gitbook.io/vlft/use-cases/factory-assets/workstation

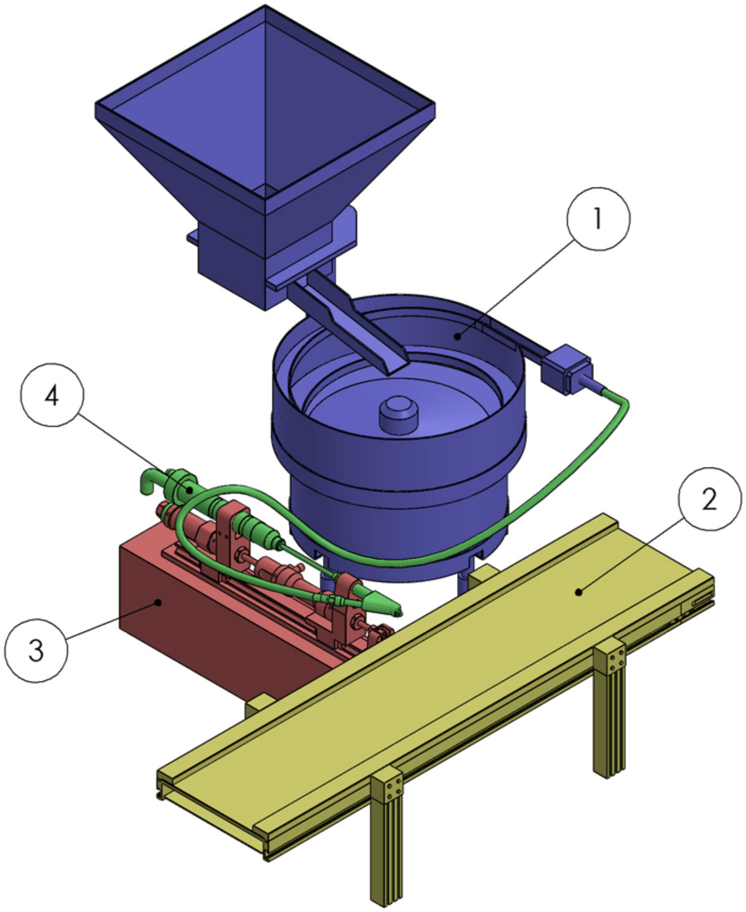


Figure 5: Pin insertion station and main components with labels defined in Table 7.

Component ID	Label	Parent	Relative Position [mm]
PIs_VibratingBowl	1	PinInsertion_Station	[-663, 0, -336]
PIs_Conveyor	2	PinInsertion_Station	[0, 0, 0]
PIs_Static	3	PinInsertion_Station	[418, 0, 0]
PIs_FloatingX	4	PIs_Static	[21, 147, 219]

Table 7: Main components of the pin insertion station.

5.3. Instantiation and Visualization of the Assembly Line

The digital model of the assembly line relies on the ontology factory data model and is based on the factory assets (i.e. hinge, operations, workstation types), as described in Sect. 5.2. The system configuration is generated using the *OntoGui-System Design* module (Figure 7) or filling in spreadsheet tables that are serialized as a .json file that, in turn, is converted into an ontology by using the *OntoGui-Utilities* module.

The resulting assembly line can be visualized and explored using VEB.js, using 3D models with basic materials[15] or PBR[16]. Furthermore, it is possible to modify the system configuration by directly changing the placement of the assets in the scene. The animation panel activates the user interface to manage the animations that can be generated from various sources, e.g. by elaborating

[15] `https://difactory.github.io/DF/scenes/VFLab/glb.html`

[16] `https://difactory.github.io/DF/scenes/VFLab/glbpbr.html`

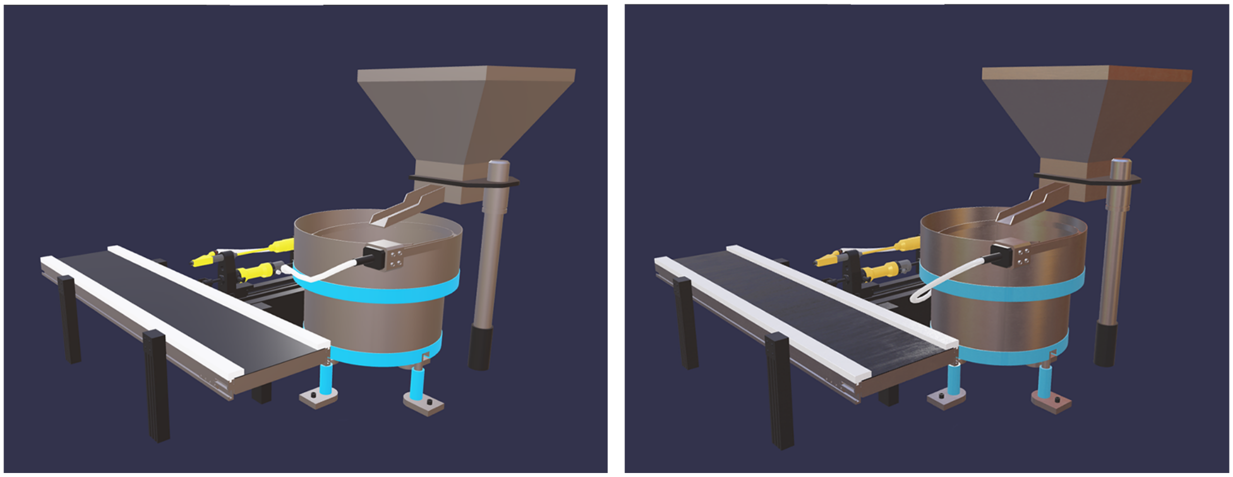


Figure 6: Rendering of .gLTF model of *PinInsertion_Station* using basic materials (left) and advanced PBR (right).

Component ID	**Time [s]**	**Relative Position [mm]**
PIs_FloatingX	0	[21, 147, -219]
PIs_FloatingX	1	[122, 147, -219]
PIs_FloatingX	2.2	[122, 147, -219]
PIs_FloatingX	3	[21, 147, -219]

Table 8: Animation of the pin insertion station associated with the *working* state.

the log of a discrete event simulation (Sect. 5.4).

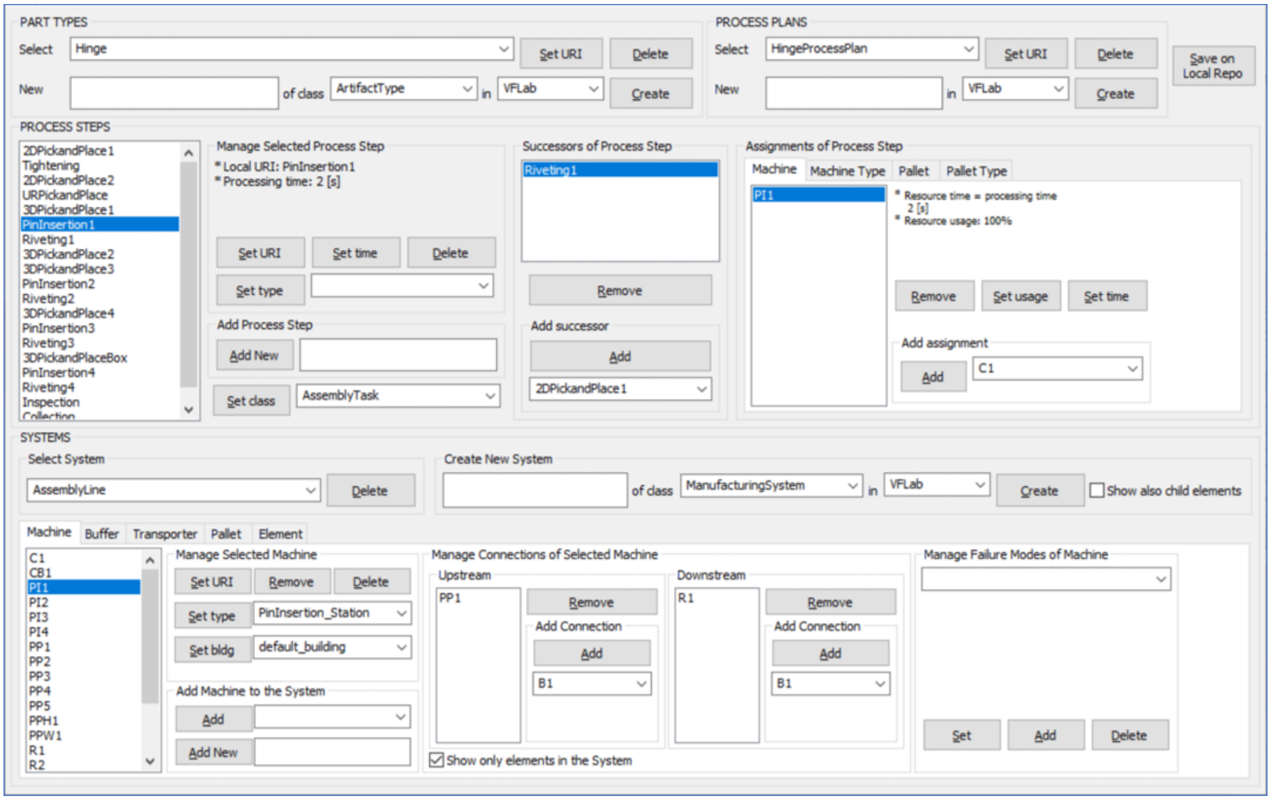
5.4. Performance Evaluation

The Performance Evaluation step has been carried out using the tools presented in Section 4.2 exploiting the automatic generation of the DES model through jsimIO (Figure 9) and executing the simulation experiments in JMT (Figure 2). The results of the experiments have been used to perform a validation of the generated models and assess KPIs measuring the performance of the candidate configurations, i.e., the throughput, the utilization of the stations, flow and queuing times for both the stations and the whole system. These results have also been exploited to select the best candidate configuration.

6. Students Feedback

The proposed learning workflow was tested by a group of 20 university students in the context of the Virtual Learning Factory Toolkit (VLFT) Erasmus+ project[17]. The participants from Politecnico di Milano and TalTech University have been grouped into international teams to use and test the methodologies and tools in the context of a joint learning lab. After completing the learning experience, questionnaires were administered to gain valuable feedback and identify directions for improvement. In particular, students were asked to evaluate the perceived competences concerning the ILOs in Table 4 both before (TO) and after (T1) the joint learning lab. The synthetic results according to a 5-point Likert scale (1: low, 5: high) are reported in Figure 10. Improvements

[17] `https://www.vlft.eu/`



PART TYPES

Select: Hinge

New: of class ArtifactType in VFLab

PROCESS PLANS

Select: HingeProcessPlan

New: in VFLab

PROCESS STEPS

2DPickandPlace1
Tightening
2DPickandPlace2
URPickandPlace
3DPickandPlace1
PinInsertion1
Riveting1
3DPickandPlace2
3DPickandPlace3
PinInsertion2
Riveting2
3DPickandPlace4
PinInsertion3
Riveting3
3DPickandPlaceBox
PinInsertion4
Riveting4
Inspection
Collection

Manage Selected Process Step

- Local URI: PinInsertion1
- Processing time: 2 [s]

Set class: AssemblyTask

Successors of Process Step: Riveting1

Add successor: 2DPickandPlace1

Assignments of Process Step

Machine: PI1

- Resource time = processing time 2 [s]
- Resource usage: 100%

Add assignment: C1

SYSTEMS

Select System: AssemblyLine

Create New System: of class ManufacturingSystem in VFLab

☐ Show also child elements

Machine
C1
CB1
PI1
PI2
PI3
PI4
PP1
PP2
PP3
PP4
PP5
PPH1
PPW1
R1
R2

Manage Selected Machine

Set type: PinInsertion_Station

Set bldg: default_building

Manage Connections of Selected Machine

Upstream: PP1

Add Connection: B1

Downstream: R1

Add Connection: B1

☑ Show only elements in the System

Manage Failure Modes of Machine

Figure 7: Process plan and assembly line definition in OntoGui-System Design.

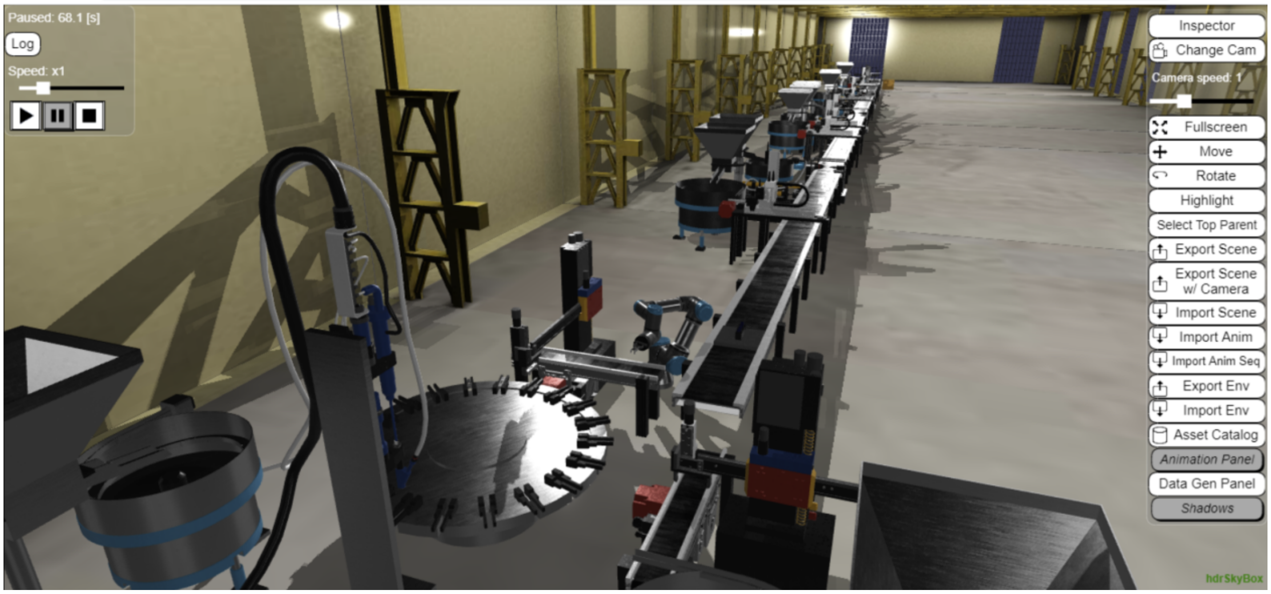


Figure 8: Complete hinge assembly line rendered in VEB.js.

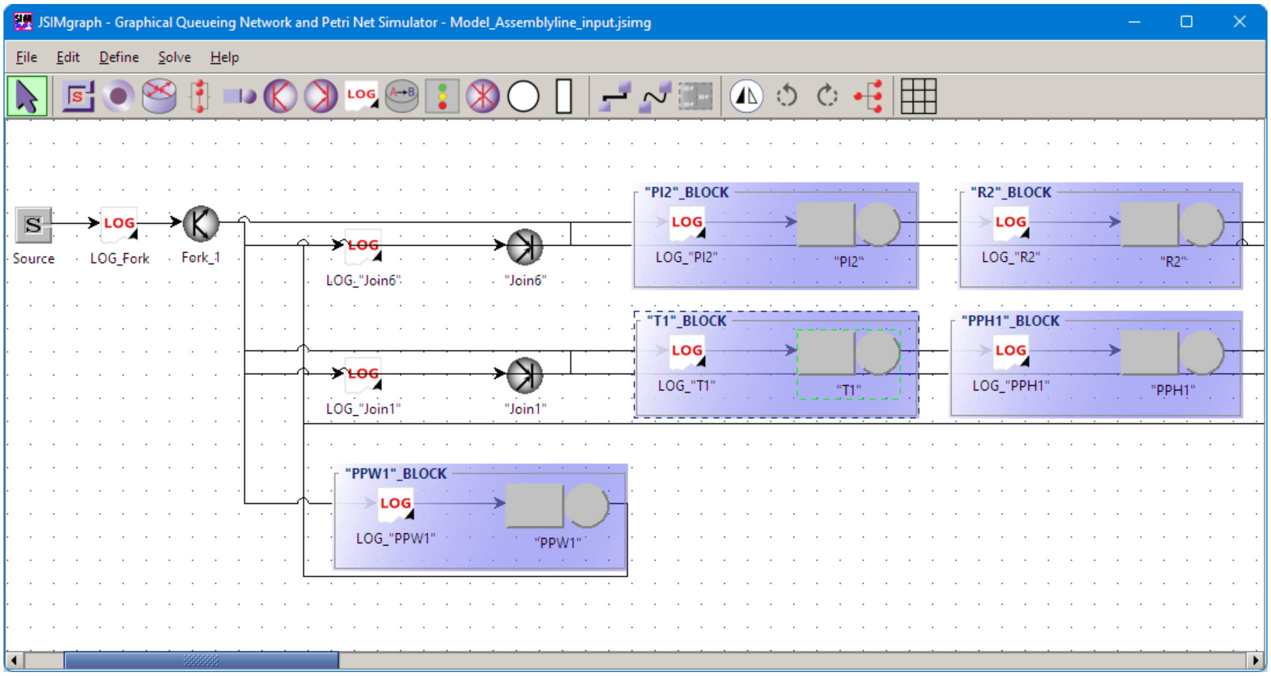


Figure 9: A DES model in JMT-JSIM generated through jsimIO. Each station is located in a finite capacity block and each node is preceded by a logger station to track the part flows. Fork and join nodes are also included to simulate the assembly of the parts.

are generally noticeable across all indicators, while qualitative feedback reported higher levels of involvement and satisfaction than traditional learning methodologies.

Furthermore, the participants also appreciated working in an international environment with different backgrounds and competences.

7. Conclusions

This work addressed the definition of a digital workflow to support learning activities in the area of the design and analysis of manufacturing systems. While available commercial software suites are capable of addressing this engineering area, they are also either too complex or expensive to be used as effective and usable learning tools in higher education. Furthermore, the objective of focusing learning activities on high-level learning objectives, to provide students skills and competencies to master complex engineering workflows, is hindered by the lack of integrated toolkits supporting the use of multiple digital tools and their interoperability.

For these reasons, an integrated learning workflow for the design and analysis of manufacturing systems has been described and integrated with targeted ILOs, engineering methodologies and the related digital tools, ranging from formal modelling, to performance evaluation and VR representation.

The workflow has been implemented and applied to a use case, i.e., an automated assembly system of an industrial product, detailing all the steps to be carried out and the functionalities provided. This implementation of the workflow has been tested by mechanical and management engineering students in the context of the VLFT Erasmus+ project. Qualitative feedback were collected, showing a higher level of involvement and satisfaction in comparison to traditional learning methodologies.

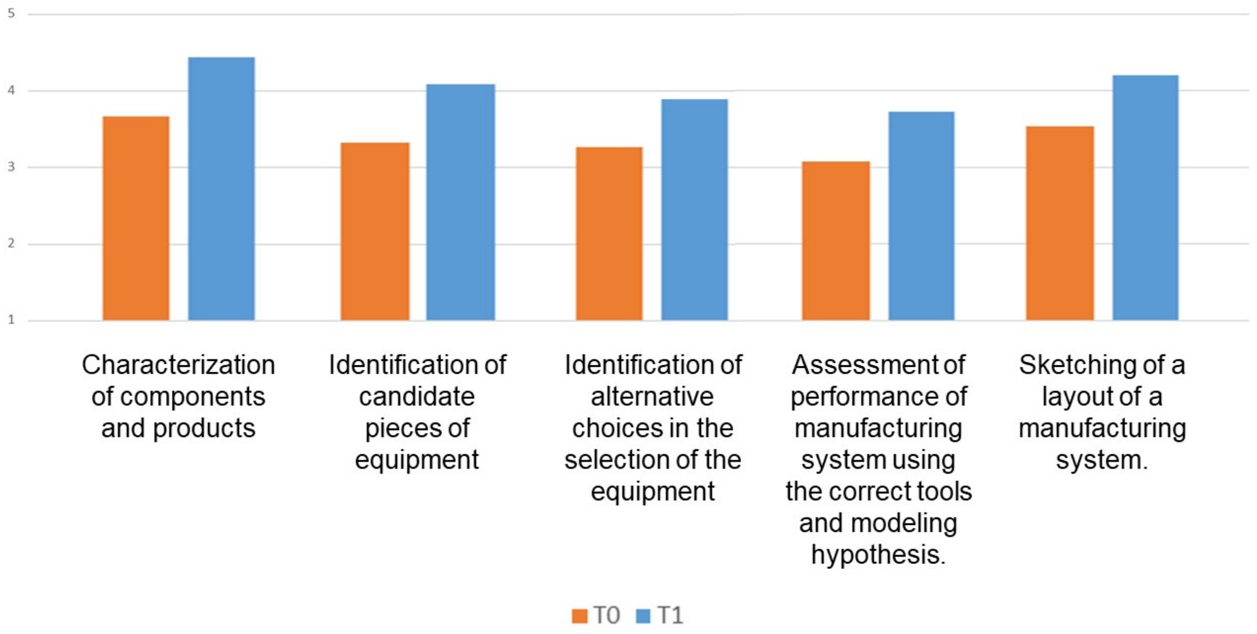


Figure 10: Perceived competences before (T0) and after (T1) the joint learning lab.

The described digital tools and use case, together with the related documentation, have been made available to the academic community under the constraint of non-profit use [42]. Further activities and development will pursue the implementation of additional use cases as well as the enrichment and improvement of the digital tools and their functionalities.

Acknowledgments

This work was partially funded by the Virtual Learning factory Toolkit (VLFT) Erasmus+ project within the call 2018-1-EE01-KA203, Grant Agreement no. 047094.

References

[1] D. Mourtzis, Simulation in the design and operation of manufacturing systems: state of the art and new trends, International Journal of Production Research 58 (2020) 1927–1949.

[2] A. Jerman, M. Bach, A. Bertoncelj, A bibliometric and topic analysis on future competences at smart factories, Machines 6 (2018).

[3] L. Kipper, S. Iepsen, A. Dal Forno, R. Frozza, L. Furstenau, J. Agnes, D. Cossul, Scientific mapping to identify competencies required by industry 4.0, Technology in Society 64 (2021).

[4] W. Terkaj, T. Tolio, The italian flagship project: Factories of the future, in: T. Tolio, G. Copani, W. Terkaj (Eds.), Factories of the Future: The Italian Flagship Initiative, Springer International Publishing, Cham, 2019, pp. 3–35.

[5] F. Baena, A. Guarin, J. Mora, J. Sauza, S. Retat, Learning factory: The path to industry 4.0, Procedia Manufacturing 9 (2017) 73–80.

[6] S. Erol, A. Jäger, P. Hold, K. Ott, W. Sihn, Tangible industry 4.0: A scenario-based approach to learning for the future of production, volume 54, 2016, pp. 13–18.

[7] T. Tolio, M. Sacco, W. Terkaj, M. Urgo, Virtual factory: An integrated framework for manufacturing systems design and analysis, Procedia CIRP 7 (2013) 25–30. Forty Sixth CIRP Conference on Manufacturing Systems 2013.

[8] W. Terkaj, T. Tolio, M. Urgo, A virtual factory approach for in situ simulation to support production and maintenance planning, CIRP Annals 64 (2015) 451 – 454.

[9] X. Yang, R. C. Malak, C. Lauer, C. Weidig, H. Hagen, B. Hamann, J. C. Aurich, O. Kreylos, Manufacturing system design with virtual factory tools, International Journal of Computer Integrated Manufacturing 28 (2015) 25–40.

[10] M. Rossoni, L. Bergonzi, G. Colombo, Integration of virtual reality in a knowledge-based engineering system for preliminary configuration and quotation of assembly lines, Computer-Aided Design and Applications 16 (2019) 329–344.

[11] G. Colombo, F. Furini, M. Rossoni, The Role of Knowledge Based Engineering in Product Configuration, Springer International Publishing, Cham, 2017, pp. 1141–1148.

[12] P. Novák, E. Serral, R. Mordinyi, R. Šindelář, Integrating heterogeneous engineering knowledge and tools for efficient industrial simulation model support, Adv. Eng. Inf. 29 (2015) 575–590.

[13] T. H.-J. Uhlemann, C. Lehmann, R. Steinhilper, The digital twin: Realizing the cyber-physical production system for industry 4.0, Procedia CIRP 61 (2017) 335–340. The 24th CIRP Conference on Life Cycle Engineering.

[14] A. Strahilov, H. Hämmerle, Engineering workflow and software tool chains of automated production systems, in: S. Biffl, A. Lüder, D. Gerhard (Eds.), Multi-Disciplinary Engineering for Cyber-Physical Production Systems: Data Models and Software Solutions for Handling Complex Engineering Projects, Springer International Publishing, Cham, 2017, pp. 207–234.

[15] M. Manzini, M. Urgo, A risk based approach to support the supplying of components in a mto assembly process, Journal of Manufacturing Systems 46 (2018) 67–78.

[16] M. Manzini, J. Unglert, D. Gyulai, M. Colledani, J. M. Jauregui-Becker, L. Monostori, M. Urgo, An integrated framework for design, management and operation of reconfigurable assembly systems, Omega 78 (2018) 69–84.

[17] E. Abele, G. Chryssolouris, W. Sihn, J. Metternich, H. ElMaraghy, G. Seliger, G. Sivard, W. ElMaraghy, V. Hummel, M. Tisch, S. Seifermann, Learning factories for future oriented research and education in manufacturing, CIRP Annals 66 (2017) 803 – 826.

[18] H. ElMaraghy, W. ElMaraghy, Learning integrated product and manufacturing systems, Procedia CIRP 32 (2015) 19–23.

[19] V. Hummel, E. Westkamper, Learning factory for advanced industrial engineering-integrated approach of the digital learning environment and the physical model factory, Production Engineering (2007) 215–227.

[20] M. Goerke, M. Schmidt, J. Busch, P. Nyhuis, Holistic approach of lean thinking in learning factories, Procedia CIRP 32 (2015) 138–143.

[21] D. Matt, E. Rauch, P. Dallasega, Mini-factory – a learning factory concept for students and small and medium sized enterprises, Procedia CIRP 17 (2014) 178–183.

[22] E. Matsas, G.-C. Vosniakos, Design of a virtual reality training system for human–robot collaboration in manufacturing tasks, International Journal on Interactive Design and Manufacturing 11 (2017) 139–153.

[23] B. Salah, M. H. Abidi, S. H. Mian, M. Krid, H. Alkhalefah, A. Abdo, Virtual reality-based engineering education to enhance manufacturing sustainability in industry 4.0, Sustainability 11 (2019).

[24] V. Toivonen, M. Lanz, H. Nylund, H. Nieminen, The fms training center - a versatile learning environment for engineering education, Procedia Manufacturing 23 (2018) 135–140.

[25] U. Wagner, T. AlGeddawy, H. ElMaraghy, E. MŸ̈ller, The state-of-the-art and prospects of learning factories, Procedia CIRP 17 (2012) 109–114.

[26] G. Sivard, T. Lundholm, XPRES—a digital learning factory for adaptive and sustainable manufacturing of future products, Reinhart G and Schnellbach P and Hilgert C and Frank SL (Eds.). 3rd Conference on Learning Factories (2013) 132–154.

[27] KNOW-FACT, KNOW-FACT. a knowledge partnership for the definition and launch of the European Teaching Factory paradigm in manufacturing education, http://www.knowfactproject.eu/, 2013.

[28] J. Biggs, C. Tang, Teaching for Quality Learning at University, McGraw-Hill Education, 2011.

[29] L. Vlăsceanu, L. Grunberg, D. Pârlea, Quality Assurance and Accreditation: A Glossary of Basic Terms and Definitions, Unesco-Cepes, 2005.

[30] P. Hitzler, M. Krötzsch, S. Rudolph, Foundations of Semantic Web technologies, CRC Press, Boca Raton, USA, 2010.

[31] S. Biffl, M. Sabou, Semantic Web Technologies for Intelligent Engineering Applications, Springer, Cham, Switzerland, 2016.

[32] M. Urgo, W. Terkaj, Formal modelling of release control policies as a plug-in for performance evaluation of manufacturing systems, CIRP Annals 69 (2020) 377 – 380.

[33] K. Mahmood, T. Otto, V. Kuts, W. Terkaj, G. E. Modoni, M. Urgo, G. Colombo, G. Haidegger, P. Kovacs, J. Stahre, Advancement in production engineering education through virtual learning factory toolkit concept, Proceedings of the Estonian Academy of Sciences 70 (2021) 374–382.

[34] B. Kádár, W. Terkaj, M. Sacco, Semantic virtual factory supporting interoperable modelling and evaluation of production systems, CIRP Annals 62 (2013) 443 – 446.

[35] W. Terkaj, P. Gaboardi, C. Trevisan, T. Tolio, M. Urgo, A digital factory platform for the design of roll shop plants, CIRP Journal of Manufacturing Science and Technology 26 (2019) 88 – 93.

[36] W3C OWL Working Group, OWL 2 Web Ontology Language, https://www.w3.org/TR/owl2-overview/, 2012. Recommendation, World Wide Web Consortium (W3C), Cambridge, USA.

[37] P. Pauwels, T. Krijnen, W. Terkaj, J. Beetz, Enhancing the ifcowl ontology with an alternative representation for geometric data, Automation in Construction 80 (2017) 77 – 94.

[38] K. Janowicz, A. Haller, S. J. Cox, D. Le Phuoc, M. Lefrançois, Sosa: A lightweight ontology for sensors, observations, samples, and actuators, Journal of Web Semantics 56 (2019) 1 – 10.

[39] P. Dolog, Model-Driven Navigation Design for Semantic Web Applications with the UML-Guide, in: Proc. of ICWE, Munich, Germany, 2004, pp. 75–86.

[40] W. Terkaj, OntoGui: a Graphical User Interface for Rapid Instantiation of OWL Ontologies, in: CEUR Workshop Proceedings, volume 2050, 2017, pp. 1–4.

[41] M. Bertoli, G. Casale, G. Serazzi, JMT: performance engineering tools for system modeling, SIGMETRICS Perform. Eval. Rev. 36 (2009) 10–15.

[42] Guidelines of the virtual learning factory toolkit, https://virtualfactory.gitbook.io/vlft/, 2021. Accessed: 2021-11-10.