Improving the Training Methods for Designers of Flexible Production Cells in Factories of the Future

Ion Mironescu, Daniel-Cristian Crăciunean, Adrian Florea, Ioan Bondrea

Improving the Training Methods for Designers of Flexible Production Cells in Factories of the Future

Ion Mironescu[1,2], Daniel-Cristian Crăciunean[1], Adrian Florea[1,*], Ioan Bondrea[3]

[1] Lucian Blaga University of Sibiu, Computer Science and Electrical Engineering Department,
[2] Lucian Blaga University of Sibiu, Agricultural Science and Food Engineering Department,
[3] Lucian Blaga University of Sibiu, Industrial Engineering and Management Department,

550025, Emil Cioran Street 4, Sibiu, Romania
{ion.mironescu, daniel.craciunean, adrian.florea, ioan.bondrea}@ulbsibiu.ro

Abstract. This work proposes a design method for flexible manufacturing systems (FMS). The method reduces the learning curve by helping employees to solve problems related to the design and optimization of the layout, operation and control of FMS, avoiding the drawbacks of current tools. The approach uses Domain Specific Modeling Languages (DSML) for specification of FMS. The paper presents the definition of the DSML and the implementation of the graphical modeling and simulation tool bringing important contributions to development of the domain through the use of constructions from categories theory for DSML specifications. This mathematical basis allows the definition of constraints to avoid supplementary costs and eventual damages through incorrect or incomplete specification of the solutions. By interconnecting with ADOxx of the DSML and tool developed, facilitates access to other analysis and simulation tools like Bee-up, Petri net, better exploration of the design space and extended support for the design activity.

Keywords: modeling, digitalization, training, theory of categories, DSML.

1 Introduction

The complexity of manufacturing processes is steadily increasing. The market has a high dynamic in volume and requirements, so that complex manufacturing processes should change in order to respond to the market demands. The trend is the transition from dedicated manufacturing lines (DMLs) to flexible manufacturing systems (FMSs) and reconfigurable manufacturing systems (RMSs) [1]. To obtain and maintain the flexibility, the enterprise need employees trained in system (re)engineering able to design, organize and supervise such systems. They should use the appropriate tools for modeling, simulation and process analysis [2].

There are a variety of tools used for the design and optimization of manufacturing system, which can be positioned between two extremes:

[*] Corresponding author

a) Pure modeling and simulation tools like Matlab / Simulink [3], Arena [4], SIMUL8 [5], iGrafx [6] with a higher degree of abstraction of the production processes but more freedom in the choice of simulation methods and in the tuning of simulation parameters. From users, this choice requires more knowledge related to the simulation techniques and the results interpretation.
b) Tools with a more realistically representation of the physical systems modeled and simulated and more integrated with the planning and execution systems of the enterprise like Visual Components[†], ProPlanner [7] and FlexSim [8]. They allow the user to analyze the simulation result in Virtual Reality (VR). FlexSim has complete support for Factory of the Future / Industry 4.0, allowing the implementation of digital twins and the integration of the physical and digital representation through Augmented Reality (AR), co-simulation, virtual PLCs, cloud computing, Big Data Analytics.

The limits between the two extremes are blurring as tools from the first category are starting to be extended with facilities of the second [9], [10].

Although proven by the practice to be very efficient, in specific context all these tools have some limitations:

1) in some cases, the inherent complexity of these programs (a high number of options and libraries with thousands of components) can be overwhelming and counterproductive in a training process;
2) the extensibility of the programs is mostly quantitatively – you can add variation of the same type of component but you cannot define an entirely new type of component;
3) the interoperability with other modeling and simulation tools is limited;
4) there is no possibility to impose constrains at the modeling language level in order to limit the modeling space to feasible/possible solutions.

At least two contexts in which these limitations occur are relevant for our research and teaching activity: the training of manufacturing system design skills and the preliminary design of new manufacturing systems. Therefore, we started the development of a design method and tool that should remove these drawbacks with the trade-off of not having the performances and all the features of the commercial tools.

The paper is structured as follows: Section 2 describes the proposed design method with the associated modeling method for manufacturing processes. In section 3 we present the training method that uses the design method. Section 4 presents the results and the discussion. The final section concludes the paper together with the main contributions and further work.

2 The Proposed Design Method

2.1 Context

As the method is used to design manufacturing system for the Factory of the Future / Industry 4.0 the method will cover some parts of the RAMI 4.0 architecture [11].

[†] https://www.visualcomponents.com

Because of its specifying destination – for training and preliminary design, the method will cover only the *Type* range on the *Life Cycle Value Stream* dimension. Only models of the manufacturing system will be created in the development stage and optimized in the maintenance and usage stage. The models will address concepts from the Asset to the Functional layers. The control scheme will include the Hierarchy levels from field devices to station.

2.2 Overview of the Overall Design Method

The design method should guide and support the user to find a solution to a design problem. A design problem is formulated as an assortment of products characterized by type of material and quantity that must be produced under some time and cost constraints. A solution to this problem has two aspects:

- Structural aspect: the components of the production line, their connections and their spatial configuration;
- Organizational aspect: the order and the timing in which the processed parts are entering each component of the production line – the schedule.

The context in which the design process takes place is: a) all the results are virtual/digital design solution and b) no prior experience exists on the problem's systems either because the user doesn't have it or because it is not existing at all (new case). Taking account of the context, the design activity is divided in three sub-activities presented below (A1÷A3).

A1. Layout design - in which the number, types, connections and placement of the components of the line are established. In this phase, the designer evaluates the general capabilities of the resulted solution, in order to assess the capabilities related to the product assortment(s) that it can produce, the maximal and minimal throughput for the line design proposed as solution. The analysis outlines also the design constraints (i.e. critical path, bottleneck components, critical failure points etc.). The assessment serves as a choice criterion for the selection of the design solutions. As a simplifying assumption, each component has the basic control loops for its parameters incorporated.

A2. Operation planning / Scheduling. At this stage, the designer tries to find an optimal schedule of jobs on the manufacturing line designed in A1. The optimization goal refers mostly to the minimization of the completion time, waiting times and/or of the job delays. The result of this activity are optimal or near optimal schedules. This activity has the role of defining the dynamic requirements for the control system which will be designed in the next activity.

A3. Control system design. At this stage, the designer develops the control system that steers the line to automatically perform the operations corresponding to a given schedule. The system should assure that the jobs are executed in the prescribed order and with the prescribed timing and that the concurrent processes are coordinated so that no undesired events occur. It specifies the number and type of control components, the connections between them and with the manufacturing line components and the control algorithms they implement. Given the assumption made for A1, this refer mainly to the supervisory control system.

When starting a solution from scratch (new production line) the order in which these activities are performed in the first cycle is: A1, A2, A3. Every improvement can imply design iteration on one activity and then design iteration on the other activities. Before the system is instantiated (e.g. the real system is built and installed), A2 should be performed sufficient times in order to cover the space of possible schedules. This process results in the design of a controlled system capable to automatically perform the schedules from the space. One application in which our research group is interested is the design of supervisory control for existing manufacturing systems that doesn't have one – and corresponding bringing them to Industry 4.0 standards. This is current in SMEs where traditionally automation / digitalization is absent or is minimal (like food industry). For such systems, the digital representation of the layout is produced through A1 and the digital representation of the current schedules is realised through A2; on the basis of these digital representations, a digital design of the control system is obtained through A3, which can be subsequently implemented.

2.3 Proposed Digital Workflow

The use of the Domain Specific Modeling Languages (DSML) in developing the digital design tools assures that the resulting applications are highly interoperable (they use the same data model) and support collaborative work (they can access a common model repository). In activity A1, the design tool is used to visually compose the layout of the manufacturing line using the graphical representation of the elements of DSML. The result is a model of the manufacturing line, both in its graphical representation (the layout of the manufacturing line) and its internal data structure representation; the last one is used to perform a structural analysis to assess the fitness of the design solution. For extending the range of performed analysis, the model is transformed in other equivalent models for which a wider range of structural analysis tools exists.

In A2 stage, the model is used in conjunction with the defined job configuration, optimization criteria and constraints, to determine the complexity of the corresponding optimization problem. This information is associated to the model in the model repository so that it can be used subsequently by all users with access to repository. The complexity determines the choice of the solving algorithm. The data structure suitable for the chosen algorithm is then generated from the model. By running the algorithm with this data structure, a schedule is obtained. By associating the schedule with the model, simulation cases are obtained; they can be simulated with the internal simulator of the design tool or with an external simulator. The internal simulator is hardcoded and uses the behavior associated to the language components. In an external simulator, different behaviors can be associated to the components, providing some flexibility.

In A3 phase, the model is completed with control elements. In the design tool, the control elements are visually placed and connected to the manufacturing line components and the control algorithms are defined in the control elements. Flexibility is assured by allowing external programs to interact with the simulation of the model so that more elaborate control strategies can be implemented in this external program. The simulation can interact with cyber physical systems to provide a realistic view.

2.4 Formalizing the Concept of Modeling Method

We implemented our modeling method by following the approach described in [12], [13] for which the ADOxx environment offers full support. We started with the first component of the modeling method, and we defined the modeling language and the modeling procedure. The modeling procedure formalizes the use of the modeling language for the models building. Then, we defined and implemented the second component of the modeling method – the ensemble of the mechanisms & algorithms which can explore and alter the state of the models. The mechanism and algorithms are the basic blocks for building more complex structural analysis and the simulation tools.

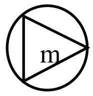
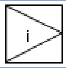
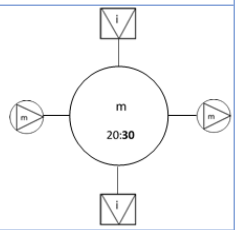
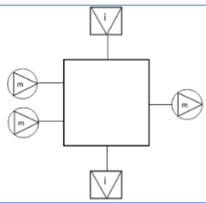
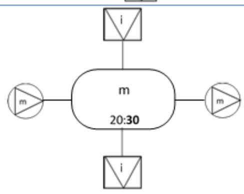
We chose to implement our modeling language as a Domain-Specific Language (DSL) [14], [15] in order to facilitate its use by domain experts. The experts will have the possibility to build models for specific domain problems (i.e. the design of a definite manufacturing system) using familiar concepts. We will use the term Domain-Specific Modeling Language (DSML) to designate the developed language. Based on our own domain expertise, we identified the basic concepts and the connections between them. A model of the manufacturing system is built as a graph with instances of the basic concepts as nodes and edges representing connection between instances. We defined a graphical notation for the DSML by assigning visual symbols to each concept. Consequently, the modeling procedure has a substantial visual programming component which is a “de facto” standard for the modern modeling and simulation method and tools. The process will be detailed in the subsection 2.5.

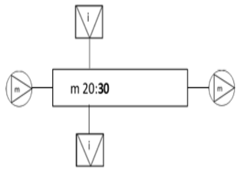
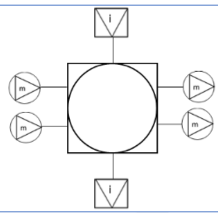
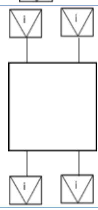
We considered the particularities of the manufacturing systems in the development of mechanisms and algorithms complementing the DSML. At structural level, a manufacturing systems can be viewed as a collection of interconnected subsystems, each with its individual state. The state of the manufacturing system at a certain point in time is the global aggregation of the local states of the subsystems. At behavioral level, each subsystem can evolve (i.e. changes its state) concurrently/in parallel to the others. The subsystems exchange information so that they can influence each other local states. Taking into account the assumptions, we considered both formalisms for representation of concurrent systems, namely Petri net [16] and the algebra of communicating systems [17], as the underlying fundament for the development at this stage. In this way, we can exploit both the graph representation of the models which can easily be transformed in other graphs (i.e. Petri net), and the categorical approach to metamodeling [18], [19].

2.5 DSML definition

The concepts and the design consideration for the proposed DSML named the Modeling Language for Manufacturing Processes (MLMP) are summarized in Table 1. The language is the basis for the categorical modeling method we propose. This is a further development of the work presented in [20]. Our method uses the theory of categories to support the metamodeling activity specific to the modeling and simulation toll development in the ADOxx environment.

Table 1. Modeling language concepts

Concept	**Graphical representation**
Ports The ports are used to realize connection between the other components modeled as concepts of the language. This connection represents way to transfer mater, information or energy between components. Every port has a type and a direction which indicates what is transferred and in which direction. In this approach, we used *M* for materials and *I* for information as such classes.	
Material type port	
Information type port	
Human operator It can be modeled as mobile control components that can attach and detached from information ports and are capable of high level control algorithms.	
Buffer Buffers are components that store material without transform it so they must have all material ports of the same type. Buffers are characterized by their maximal capacity – the number of units of the same type that they can store. The maximal capacity is fixed and can't be extended (constant attribute). Their variable attribute is the current content which can vary between 0 to maxim capacity.	
Machines Are components that transfer material between their input and output ports asa rezult of the commands they receive on the in information port.	
Transport machines Are components that transfer material without transforming it. They only change the position of material from the input buffer to the output buffer.	
Workstation Workstation are components that transform materials from one type to another	
Autonomous guided vehicles (AGV) Are practically mobile buffers having all attributes and behavior of a buffer but can connect and disconnect their ports from the corresponding ports of buffers and can move between preprogramed positions.	

Conveyors, belts, pipes (CBP) They transport only one material so all ports are of the same type and the defining feature is the throughput – number of material units transported in the time unit. This can be variable in some limits, thus the component has a minimal throughput and a maximal throughput as constant attributes and the current throughput as a variable attribute with values in this range.	
Manipulator Is a flexible transporter that can transfer multiple types of materials between different In and Out ports of the same type. The most usual example is an robotic arm that can handle different types of materials moving them between different manufacturing (sub)lines.	
Control component Are components that transfer only information so they have only information ports - not necessarily all of the same information type. They allow the definition of control algorithms that describe what command are transferred to other components on the in ports as the result of the signal received on the in ports.	

2.6 Categorical Specification of MLMP

To give a mathematical foundation to the process of the generating models in the MLMP, firstly each concept corresponding to a component type presented in Table 1 were described using the set theory. On this basis, we then defined the MLMP model as a directed graph with nodes representing the concepts and arcs representing the connections.

We used a categorical sketch as a metamodel for MLMP. The sketch has the property to incorporate beside an exact formal syntax also the semantic of the metamodel. We started from a general sketch [21], [22] (a directed multigraph with loops containing t the MLMP concepts) and added the constraints from above. We defined also the functor that transform a sketch $\boldsymbol{\mathcal{S}}=(\boldsymbol{\mathcal{G}}, \boldsymbol{\mathcal{C}}(\boldsymbol{\mathcal{G}}))$ in a static visual model. We added to the sketch S beside the constraints ($\boldsymbol{\mathcal{G}}$) also types, attributes and behavioral rules allowing the definition of the model's behavior. Double pushout graph transformations [23], [24] were used to specify the behavior of model described using the MLMP model. The rigorous mathematical description of the method is beyond the scope of this article and is provided on line‡. We used the structures and rules to implement the metamodel in the ADOxx environment [15] and we developed a model editor and simulator for MLMP. This was the basic tool used in the training process described in the following section. The ADOxx ecosystem supports the development of collaborative tools based on the same model.

‡ https://digifof.eu/sites/default/files/d3.3_design_method_for_the_factory_of_the_future.pdf

3 Training Method

During the development stages and after the release of the alpha version we have conducted several tests to gather feedback, understand the usability issues and identify how the method is improving the classical approach. The final testing process involved 12 people of different ages, but with the same technical background (engineering students, professors, researchers). The people were grouped in four teams so that each comprised one member which was familiar with the manufacturing process, one familiar with the optimizations methods and one familiar with control problems. The teams took part in Problem Based Learning trainings for manufacturing plant design. The training was divided in four sessions: one preparatory and three following the activities of the design method. In the preparatory session, the team members known each other, the used tools and methodology were presented and used in small didactic examples. In the second session, the teams had to design the layout of manufacturing lines that should produce a given assortment of products with a specified throughput and prove through analysis the capabilities of the designed line. In the third session, they had to find an optimal schedule for the production line designed in the previous session. In the final session, the trainees had to design the control system for the designed plant.

4. Results and Discussions

According to the specialization of the manufacturing engineers involved, two training were performed each with two teams. The use case where provided by the industry. A first step in establishing a collaboration network with the industry based on the design tool was put in place by installing the application at the enterprises witch accepted to offer the cases and connecting it to the shared repository. After a short tutorial, the design engineer where capable of expressing the design problems in the application. The use cases where stored in the repository being so accessible on line to the team participants.

The first training has started with a food industry case – a chocolate manufacturing line. The second training used a case from automotive industry. In each of them, one team used the MLMP modeling as support tool for designing the line layout, the other used only a graphic program (Sketch-up) to draw the layout. The problem settings where exported for them from the collaborative repository. In the next training sessions, the Petri net tool was used for the optimization and controller synthesis. Below we present the use cases:

1. For the first use case, the trainees had to design a manufacturing line that produces chocolate truffles and packages them first in aluminium foil bags and then in cardboard boxes. This case came from a manufacturer of confectionery products which has a high mix low volume manufacturing process with a variable demand in assortment and volume, relatively short production cycle (1 day) and a just-in-time producing policy. The variability is given by both the confectionery assortment and the packaging type.

The position of the raw materials storages, the shape and dimension of the factory floor are given. A design solution expressed in MLMP is presented in figure 1.

- AGV1 transports the buckets with chocolate ganache mass (m1) from the buffer B1 to the buffer B3;
- AGV2 transports the cardboard packaging material (m2) from the buffer B2 to the buffer B7;
- On line 1 (L1), the chocolate ganache mass (m1) is taken from B3 by the workstation WS1; inside WS1 chocolate is melted and chocolate truffles (m3) are formed. The product m3 is discharged in the buffer B4 feeding the transport line with freezing areas B11. The chocolate truffles are loaded in the buffer B5, which is feeding WS2. Two processes occur here: the formation of the aluminum bags (material m7, brought from the buffer B12) and the filling of the aluminum bags with chocolate truffles. The product is m4 – an aluminum bag filled with chocolate truffles – and is stored in the buffer B6;
- On line 2 (L2), cardboard packaging material (m2) is transported by AGV2 to the WS3, where the cardboard packaging boxes (m5) are formed; they are loaded in the buffer B8;
- On the assembly line, the Manipulator M1 transfers m4 and m5 products in the feeding buffers B9 and B10 of the WS4, where the aluminum bags with truffles are introduced in the cardboard boxes. The final products m6 (cardboard boxes having aluminum bags with truffles) are sent in B11 buffer.

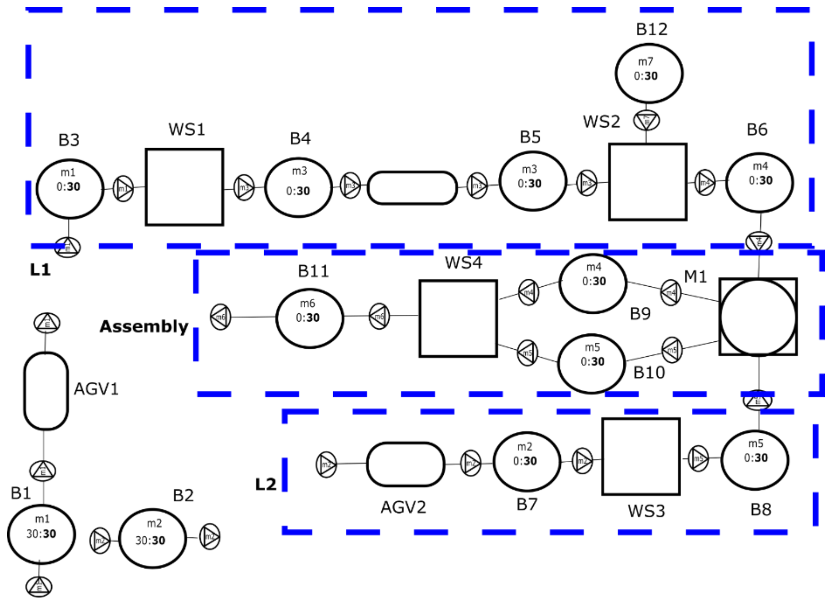


Fig. 1. MLMP model of the chocolate truffle line

For the analysis, the case was transformed to an equivalent timed Petri net. The Petri net was used to check the connectivity of the net, the reachability of the final state (final product in the storage), completion the throughput (number of tokens passing through the net in a given time). In the third training session, the model was asked to optimise a schedule for a number of jobs, each with specific product quantities of products (processing times) and specific deadline. The solution was computed in Matlab and then simulated in the design tool for verification. In the last session, the automatization system was developed by the students through Petri net based controller synthesis. The model of the plant obtained in the session 2 was used to synthesise of a Petri net controller. One particular problem to solve was to force the manipulator M to put alternatively bags and cardboard on the buffers of the WS4 (deadlock danger). The control system was then implemented by that team that used the model in the design tool using the control elements and tested through simulation.

2. The second use case is specific to automotive industry. The analysed case is of a low mix high volume manufacturing line. The business is order based, with an established customers' base of automobile producers with a relatively high and predictable demand.
   The setting, presented in figure 2, is an assembly line for an optoelectronic part of the automatic breaking system. The components of the part (e. g PCB m1, casing m2, lenses m3) are temporarily stored in buffers along the line. On each component, particular operation (adjusting, heating, welding, testing) are performed on the corresponding workstations (WS1-WS8) and after that they are assembled together. In the original setup, the parts are moved from a station to another by human operators (H1, H2, H3), which execute the operations, too. The case starts from an existing layout of the workstations and buffers. The problem raised by the industrial partner submitting the case was to design the paths of operators so that to minimise the completion time for one piece. Thus, in the design session the teams had to draw feasible paths for the human operators, to place supplementary buffers if needed and to evaluate the performances. In the third and fourth sessions, the approach was similar to the other cases.

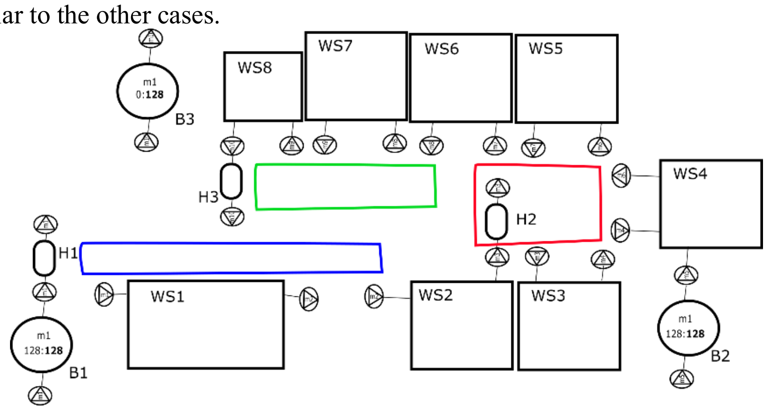


Fig. 2. MLMP model of the automotive assembly line

The comparison between the two teams had revealed the following differences:

- The team that used the design tool has found faster (in shorter time) a functional solution. This had two reasons:

    - The collaboration and communication between the team members was better as the expression of ideas and the comprehension of change proposals was eased by the use of a common graphical language.
    - Although the drawing tool was also collaborative, it distributed only the visual changes of the layout. The team that used the design tool accessed each time the updated model so that the direct analysis was possible.
    - The second team was slower because they were making mistakes which where only late discovered, after the translation to Petri net and structural analysis. The most common mistakes for those that had not used the modeling tools were: forgotten buffers, incorrect connected components and unconnected components. These mistakes were mostly prevented by the modeling tool.
- The translation to the Petri net and back was faster for the team that used the modeling tool because the pattern of transformation were easier to identify for the simplified models as for the real objects.
- The formulation of the optimisation problems was easier as the particularities of the system (number of machines, interdependencies were easier read from model).
- The structure of the real control system was better understood by the users of the modeling tool. They quickly identified patterns for conversion in both directions.

The solutions developed by the team using the collaborative design application were stored in the repository and became immediately accessible to the design engineers from the two enterprises which gave the cases. The engineers took part, virtually, to the assessment of the solutions. They also presented the solution to the management; we received positive feedback. An integration with a collaborative platform for the Knowledge Triangle developed in the frame of the KNOWinFOOD project [25] is in discussion; this integration will allow the solving of tasks given by industry by students and researchers from the high education institutes. Also, the training experience was shared through the repository with the other OMiLAB laboratories.

The training was a real use case of the design method and the associated tool. The use case shoved that both, design method and tool, perform well in the context for which they were developed (meaning the forming of manufacturing system designer skills for employees with absent or less prior experience). The trainees succeeded, with little training, to develop a digital models of two manufacturing lines and to generate the digital artefacts to cover the intended parts of the RAMI 4.0 architecture.

The categorical formulation of the metamodeling language gives a formal definition to the modeling language. This allows: formal analysis of the models; formal specification of constraints that enforce some properties on the models limiting the design space; axiomatic construction of models. The imposing of constraints was used in the application to limit the possibility to generate bad designs with good results. The students produced faster valid designs by exploring only the design space of valid states.

The formal definition of the modeling language is the basis of easy manual or automatic translation to other modeling languages with formal definition, like Petri net, property that was also used by the students. The formal definition allows the axiomatic construction of models. This creates the possibility to extend our tool in the next iteration to support axiomatic design.

Compared to general modeling languages (UML, BPMN, EPC) and similar or derivative languages used in some tools (iGrafx), the DSML we proposed is less abstract. Basic concepts as buffer or transport belts must not be constructed by refining or composing language elements. Compared with the languages of general purpose simulation and modeling tools like Matlab Simulink SIMULA8, our modeling language is better suited for the design of manufacturing system layouts as the geometrical dimensions and position of elements are of relevance and they are neglected in the aforementioned languages. Our language cannot compete in expressive power with the languages of the tools with full-fledged 3D graphical development environments (Arena, AnyLogic, FlexSim) but gains with its ease of use (few components to learn) and support for feasible designs (no incorrect models). Only Flexsim and Anylogic have some shared model repository capabilities due to their web based character. No tool allows the combination of different modeling languages similar to the facilities offered to our developed language by the underlying ADOxx ecosystem.

5 Conclusions and Further Work

This work describes the development of a modeling method and the associated DSML-MLMP to be used as a core element in the design of the digital factory of the future. The mathematical fundamentals of developing the modeling language and methods and arguments sustaining the adequacy of the chosen mathematical instrument (category theory) for this task are presented.

The developed design method and associated DSML and tool where tested in a training for manufacturing system designer skills. The results of the training have shown that they are easy to learn and to use in the context of no or little design experience. Using the OMiLAB ecosystem[§], the tool facilitates the collaboration between the training team members and allows the establishment of collaborative networks on the knowledge triangle [15], [24].

For the future development of the instrument, we plan to introduce the following features:

- Support all activities of the design method with tools integrated through the ADOxx ecosystem. This allows the better interoperability of the tools, reducing the time of switching between the tools and imposing constrains that assures the avoidance of faulty or inconsistent design solution on all levels.
- Formalize the training activities based on further in depth studies with more statistical relevance and build a virtual training environment.
- Use the model to build a digital twin of the factory.

Acknowledgments. This work was partially developed under the ERASMUS+ KA2 project "The FOF-Designer: Digital Design Skills for Factories of the Future", financing contract no. 2018-2553/001-001, project number 601089-EPP-1-2018-1-RO-EPPKA2-KA, web: https://www.digifof.eu/ .

[§] https://www.omilab.org/nodes/nodes.html

References

1. L. Benyoucef, Introduction, in Reconfigurable Manufacturing Systems: From Design to Implementation, L. Benyoucef (ed). Springer, 2018
2. D. Mourtzis, Simulation in the design and operation of manufacturing systems: state of the art and new trends, International Journal of Production Research, DOI: 10.1080/00207543.2019.1636321
3. H. Klee, R. Allen, Simulation of Dynamic Systems with MATLAB® and Simulink® [3rd ed.] Taylor & Francis, 2018.
4. M. D. Rossetti, Simulation Modeling and Arena, [2nd ed.] Wiley, 2015.
5. J. Fousek, M. Kuncová, J. Fábry: Discrete Event Simulation - Production Model in SIMUL8. ECMS 2017: 229-234.
6. A. Kolinski, B. Sliwczynski, P.Golinska-Dawson, The assessment of the economic efficiency of production process - simulation approach, 24th International Conference on Production Research ICPR, (2017).
7. D. Sly, J. Schneider, Collaborative Process Engineering Framework for Ground Vehicle Systems Manufacturing, 2011 NDIA Ground Vehicle Systems Engineering And Technology Symposium, Systems Engineering And Integration (Se)Mini-Symposium August 9-11 Dearborn, Michigan, PE, (2011).
8. P. Pawlewski, P. Hoffa-Dabrowska, P.Golinska-Dawson, K.Werner-Lewandowska, FlexSim in Academe: Teaching and Research (ed), Springer, (2019)
9. N. Khaled, B. Pattel, A. Siddiqui, Digital Twin Development and Deployment on the Cloud: Developing Cloud-Friendly Dynamic Models Using Simulink®/SimscapeTM and Amazon AWS, Academic Press, 2020
10. A. Borshchev, The Big Book of Simulation Modeling Multimethod Modeling with Anylogic 6, AnyLogic North America, 2013.
11. B. Lydon, RAMI 4.0 reference architectural model for Industrie 4.0, InTech, 66(2), 2019.
12. D. Karagiannis, H.Kühn, Metamodelling Platforms. In: Bauknecht K., Tjoa A.M., Quirchmayr G. (eds) E-Commerce and Web Technologies. EC-Web 2002. LNCS, vol. 2455, pp. 182. Springer, Heidelberg (2002).
13. D. Bork, R.A. Buchman, D. Karagiannis, M. Lee, E.T. Miron, An Open Platform for Modeling Method Conceptualization: The OMiLAB Digital Ecosystem, Communications of the Association for Information Systems, 34, 555-579, (2019).
14. M. Fowler, R. Parsons, Domain Specific Languages, 1st ed. Addison-Wesley Longman, Amsterdam, 2010.
15. D. Karagiannis, H.C. Mayr, J. Mylopoulos, Domain-Specific Conceptual Modeling Concepts, Methods and Tools. Springer International Publishing Switzerland (2016).
16. W. Reisig, Understanding Petri nets, Modeling Techniques, Analysis Methods, Case Studies, Springer, 2019
17. W. J. Fokkink. Modelling Distributed Systems. Texts in Theoretical Computer Science, An EATCS Series,. Springer, 2007.
18. D. C. Crăciunean, D. Karagiannis, Categorical Modeling Method of Intelligent Workflow. In: Groza A., Prasath R. (eds) Mining Intelligence and Knowledge Exploration. MIKE Lecture Notes in Computer Science, vol 11308. Springer, Cham (2018).
19. D. C. Crăciunean, Categorical Grammars for Processes Modeling, International Journal of Advanced Computer Science and Applications (IJACSA), 10(1), (2019).
20. I. D. Mironescu, An ADOxx Based Environment for Problem Based Learning in Manufacturing Systems Design, 9th International Conference on Manufacturing Science and Education “Trends in New Industrial Revolution”, MATEC web conf., vol. 290, 2019.
21. U. Wolter, Z. Diskin, The Next Hundred Diagrammatic Specification Techniques, A Gentle Introduction to Generalized Sketches, 02 September 2015.

22. Z. Diskin, T. Maibaum, Category Theory and Model-Driven Engineering: From Formal Semantics to Design Patterns and Beyond, ACCAT 2012.
23. D. Plump, 'Checking graph-transformation systems for confluence', ECEASST, vol. 26, 2010. DOI: 10.14279/tuj.eceasst.26.367.
24. H. Ehrig, C. Ermel, U. Golas, F. Hermann, Graph and Model Transformation General Framework and Applications, Springer-Verlag Heidelberg 2015.
25. C. Georgescu, M. Mironescu, I. D. Mironescu, The "Knowledge triangle for food innovation by harnessing the tradition and assuring sustainability" KNOWinFOOD Project as a tool to educate students to obtain innovative products incorporating bioactive compounds, Advances in Pharmacology and Clinical Trials, 2019, vol 4, no. 1, 5 p, https://medwinpublishers.com/APCT/APCT16000150.pdf
26. D. Karagiannis, R. A. Buchmann, X. Boucher, S. Cavalieri, A. Florea and D. Kiritsis, OMiLAB: a Smart Innovation Environment for Digital Engineers, accepted to PRO-VE 2020, 21st IFIP / SOCOLNET Working Conference on Virtual Enterprises, Valencia, Spain, November 2020.