



Micro Manufacturing Unit and the Corresponding 3D-Model for the Digital Twin

Citation

Lohtander, M., Garcia, E., Lanz, M., Volotinen, J., Ratava, J., & Kaakkunen, J. (2018). Micro Manufacturing Unit and the Corresponding 3D-Model for the Digital Twin. *Procedia Manufacturing*, 17, 55-61.
<https://doi.org/10.1016/j.promfg.2018.10.071>

Year

2018

Version

Publisher's PDF (version of record)

Link to publication

[TUTCRIS Portal \(http://www.tut.fi/tutcris\)](http://www.tut.fi/tutcris)

Published in

Procedia Manufacturing

DOI

[10.1016/j.promfg.2018.10.071](https://doi.org/10.1016/j.promfg.2018.10.071)

Copyright

This is an open access article under the CC BY-NC-ND license (<http://creativecommons.org/licenses/by-nc-nd/3.0/>)

Take down policy

If you believe that this document breaches copyright, please contact cris.tau@tuni.fi, and we will remove access to the work immediately and investigate your claim.



28th International Conference on Flexible Automation and Intelligent Manufacturing
(FAIM2018), June 11-14, 2018, Columbus, OH, USA

Micro Manufacturing Unit – Creating Digital Twin Objects with Common Engineering Software

Mika Lohtander^{a,*}, Elodie Garcia^a, Minna Lanz^b, Jarno Volotinen^a, Juho Ratava^a, Jarno
Kaakkunen^a

^aLappeenranta University of Technology, Skinnarilankatu 34, 53850 Lappeenranta, Finland

^bTampere University of Technology, Korkeakoulunkatu 10, 33720 Tampere, Finland

Abstract

A Micro Manufacturing Unit (MMU) used in this research is meant for digital twin research environment to study how to construct a digital twin from scratch and what kind of information is needed to describe the realistic behavior of the digital model of a micro manufacturing unit. The basis for a digital twin is an actual machine and imitating digital model. The basic model for a digital twin of the MMU will be built using production and part simulation software (FlexSim), permitting the immediate integration of the machine in an industrial context, and allowing to control all the parameters of the production system afterward. This study presents basic logic and software logic how to build an imitative model with engineering tools used commonly in development of production processes. The study shows that common engineering tools are not ready yet to the daily basis usage of virtual reality. Furthermore, the study shows that at the moment advanced coding skills are needed to create an appropriate model for digital twin purposes.

© 2018 The Authors. Published by Elsevier B.V.

This is an open access article under the CC BY-NC-ND license (<http://creativecommons.org/licenses/by-nc-nd/3.0/>)

Peer-review under responsibility of the scientific committee of the 28th Flexible Automation and Intelligent Manufacturing (FAIM2018) Conference.

Keywords: Micro manufacturing, micro manufacturing unit, virtual reality, immerse technology, interactive object.

1. Introduction

The traditional definition of manufacturing states that manufacturing is a process that turns raw materials into physical products. Currently we understand that the physical part of the production process is a part of a much wider

value chain including a raw material process, product and service process and in some cases recycling. Manufacturers are increasingly generating revenue from other activities, many of which are classified services.

The globalization of the economies stays a major challenge to the local industries, and it is pushing the manufacturing sector to the next transformation, which might consist of, at least distributed production, digitally predictive manufacturing and some cases even personal manufacturing. On the other hand, production platforms will play a remarkable role in future. Production platforms are digital based services, where client order products from web service without knowing where physical production is done. The basic logic of production platforms might imitate service like Uber. One important role of a manufacturing platform is how to describe and presents the capabilities of the resources. A digital twin is one possible solution, but also for example direct capability ontologies are possible.

To become more competitive, manufacturers need to embrace emerging advanced analytics and digitally presented systems to improve their efficiency and productivity. In future several stakeholders must work together, to deepen cooperation between different actors is necessary. Networked stakeholder and devices between them via the Internet (IoT) allow wider and deeper cooperation between different parties than previously possible. Individuals, technology providers, communities, machinery manufacturer and other actors could provide information concerning simulation, optimization, and material behavior.

Traditional engineering work like electrical engineering or mechanical engineering design paradigms needs new means of manufacturing design to be used. Therefore, simplified models of a machine are needed to create proper (simplified) calculation models to clarify operations between stakeholders. Simplified models could be for example a vibration model, a clamping model, tolerance model, handling model and so on. The design decisions, originating from either known compromises or approximations, can lead to undesirable and unpredicted emergent behavior are set in place during the phases of creation and production and realized during the operational phase. One of the greatest known design challenges relates to the unexpected or undesired human interaction with the product itself [1]. The need for better understanding and the utilization of a product over the life cycles drives the inclusion of complex system behavior in the models of the digital era. Digital twin concepts aim to allow the observation of the behavior of the whole system and prediction of the system behavior during the utilization.

2. Definition of a Digital Twin

Based on literature research, Digital Twins have been used at least in product design [2] and re-engineering of aircraft [3]. Based on literature research, Grieves might be one of the first authors who present the concept of digital twin in 2005 [4]. In the following years, various definitions for the Digital Twin have been proposed. Hochhalter et al. [5] suggest that a Digital Twin is “a life management and certification paradigm whereby models and simulations consist of as-built a vehicle state, as experienced loads and environments, and other vehicle-specific historical information to enable high-fidelity modeling of individual aerospace vehicles throughout their service lives.” Reifsnider and Majumdar [6] present an idea that the Digital Twin is “an ultra-high fidelity simulation integrating with an onboard health management system, maintenance history, and historical vehicle and fleet data.” Based on slightly more concrete idea Deloitte’s [7] definition is next: “the Digital Twin is a dynamic digital representation of industrial infrastructure. Furthermore, the Digital Twin enables better understanding and the prediction of the performance of production machines, production flow, and cost structure.” The Digital Twin represents all physical devices as they are in a real world, collects data from a real system to represent real-time information from real machinery, and combines information from the real and virtual compositions. Thus, the Digital Twin can be defined as a dynamic digital model of historical and current behavior of a physical process environment helping to optimize business capability.

A generic and maybe the most common definition of the Digital Twin is given by Glaesegen and Stargel in 2012 [8]: “digital twin is an integrated multi-physic, multi scale, probabilistic simulation of a complex product and uses the best available physical models, sensor updates, etc., to mirror the life of its corresponding twin.” Whether it is the definition of a digital twin, it is almost essential that the manufacturing machine characteristics be described and then constructed a model with appropriate properties needed in production.

During last several years, the concept of the Digital Twin has been adopted by the manufacturing industry to represent the physical and digital properties of the current and future status of the processes and equipment [9]. Therefore, we find it fair to state that in the manufacturing environment the Digital Twin is a way to present machinery,

raw material, tools, waste material, process helping aids (for example raw material processing), and other static and dynamic objects by digital means. The digital models can get data from sensors attached to the object, allowing an online model to show the real-time status, shape, condition, speed and other important information in an intelligent and interactive monitoring system. The coupling between physical and virtual makes the Digital Twin a parallel environment, which reflects the process behavior in the real world directly. The Digital Twin allows displaying the systems and processes over time – past, present, and future. With the Digital Twin, it might be possible to create algorithms, analyze unstructured data, record data from the (virtual) system model and analyze the (virtual) system model without disturbing real business processes. The result is an intelligent and scalable virtual environment or digitalized system description, which can be used to detect potential risks within the real system, and pre-emptively propose corrective means before the problem is even encountered in the real world.

Five different fields related to manufacturing science must be considered to create a proper imitative digital model of the manufacturing environment. These fields are [10]

- Materials – raw materials, waste materials, product material (properties and functions),
- Machines (precision and capabilities),
- Methods (efficiency and productivity),
- Measurements (sensing and improvement), and
- Modeling (prediction, optimization, and prevention).

Challenges of utilization of Digital Twins are connected to the overall system complexity [11]. The observed systems are simply enormous from the data collection, model creation, and computational perspective. The other challenge lies in the cross-disciplinary field, where the challenge is manifesting as lack of awareness of models and modeling methodology, and the lack of access to information [12], and resistance to change [13]. Even with the partial solutions created, the scientific world has produced number success stories in a manufacturing field via utilization of digital twins [14],[15].

3. Manufacturing Machines's Model for the Digital Twin

Manufacturing simulation environments have been used developing and improving new and existing manufacturing and production environments. Simulation plays a significant role in teaching and education. Students use simulation software in assignments and other activities. Simulation model and machines inside of a model are the first touch for manufacturing for a large number of students. Based on literature review, discussion with a software distributor and forums seems to be that this research is the first one or one of the firsts attempt, to create interactive and real-time virtual reality-based advanced manufacturing environment, which imitate real-world events at a machine level and are suitable for education purposes. It is important to note that the software tools used in the digital twin development are those, which students are familiar.

The Micro Manufacturing Unit (MMU) presented in this particular study is built directly into research and educational purposes. The MMU consist of a system controller, a network interface, safety functions, a two-layer palletizing system, a manipulator to handle products and two work cells. The mechanical structure of the work cells is identical in this particular Unit, but it is possible to add different kinds of cells to the configuration. The system controller allows up to one hundred new works cells, but mechanical restrictions must be taken into account. The mechanical installation of a new cell is quite easy, only lengthening of the manipulator line is needed after installation of the new work cell. Obviously, power lines must be modified if added more cells. It could define this unit as a Flexible Micro Manufacturing Center (FMMC).

The micro manufacturing unit includes several actuators like doors and hatches. Subassemblies do not include in this stage any of movable items, so all subassemblies are modeled without any movable functions or features. All movable features are created in the main assembly. This kind of model is not realistic, but this decision guarantee that subassemblies stay simple and main assembly stays understandable at this stage. Later when the model is evolving, it is possible to add subassemblies, under the basic structure. Furthermore, the whole design follows bottom-to-top design. For educational purposes, primarily, a simple model might provide more understanding for example first and second-year university students. Figure 1. shows the basic structure of the micro manufacturing unit.

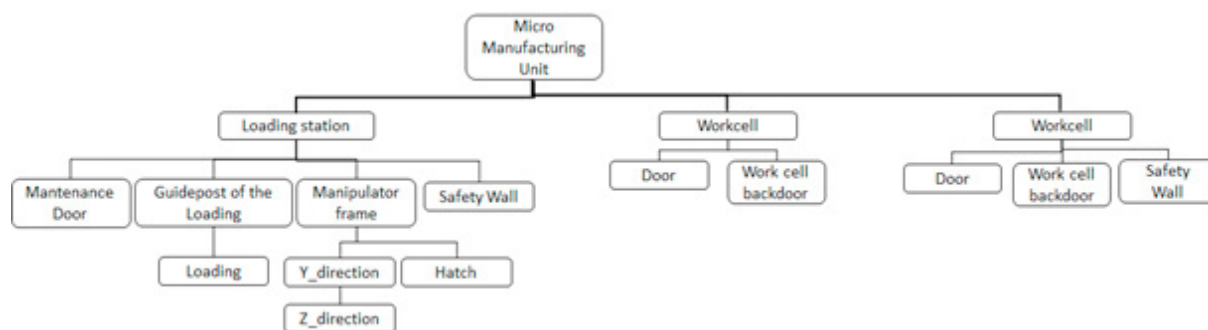


Fig. 1 The main structure of the MMU.

The accurate analysis of function, functionality, and movement included all restrictions must be done to create a proper VR-model to software environment. This work has been shown in earlier studies [15].

4. Building the first interactive object

The first machine model for the digital twin of the MMU will be built using production/part simulation software (FlexSim), this will permit the immediate integration of the machine in an industrial context, allowing to control all the parameters of the production system afterward.

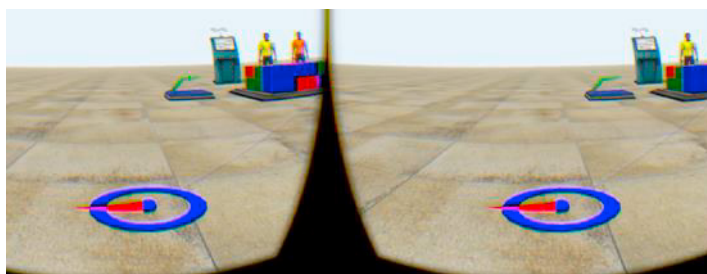


Fig. 2 Virtual reality in manufacturing context gives a completely new dimension.

In one of the demonstration models provided by the developers, many interactive elements are available to test. It is possible to teleport and move around a wide area, push buttons that release conveyor merge lanes, move figurines that change the operator assignments and control a crane using a joystick and pulling a handle. All this implemented in an active working model environment [3]. These new software functionalities make an opening in the possibilities of creating a digital twin that is a high fidelity model of the real machine and gives the opportunity to interact with and operate it in real time.

The main code which handles the touch inputs can be found in the “ModelOnPreDraw” script under the User Command dialog. The programming language used by FlexSim in scripts such as ModelOnPreDraw is called Flexscript, though it is also possible to use C++ or external functions in dynamically linked libraries (DLL). In this demonstration model, there is additional code under the User Command dialog. These scripts are auxiliary functions the main code calls, such as “getDistance,” “getHandCollisionSphere,” “getHandGripPosition,” “ModelOnDraw,” “assignOperatorToDispatcher,” and “onButtonPress.”

The ModelOnPreDraw code first sets the view to fit virtual reality and resets the time. Second, it defines the connection between the Oculus Touch hand controllers and the virtual objects. This contact is called the Collision Sphere; it defines in which area the virtual hands of the user are touching the virtual objects. It is used afterward to set the interaction between the user and the different elements.

The following parts of the code define all the interactive elements, how they behave and what is their impact on the active model. The section about the Figurines explains that the user can move this category of objects all around the virtual environment, but when they are linked to a target they can only be released into one of the targets. Moreover, in this demonstration model two figurines represent two operators, at the beginning of the simulation they are a resting position, they start executing their tasks when the user puts the associated figurine in one of the three targets. Each target is in front of a different processor figurine, one is blue, one is red, and one is green. These colors also correspond to the colors of the boxes that appear continuously in the source. If the operator's figurine is placed in front of the blue processor, the operator will pick up only the blue boxes from the source and put them in the blue processor where they will be driven by a conveyor belt. One of the others elements to be defined in the code is the Buttons. In this model there are three white buttons, each one releases one type of boxes (blue, red or green) into a second conveyor belt. The third and the fourth interactive elements to be described are the Joystick and the Handle. They both control a crane that picks the boxes from the end of the conveyor belt and releases them to another conveyor belt for storage. The movement of the Joystick and the Handle as their impact on the running model are also set on this code.

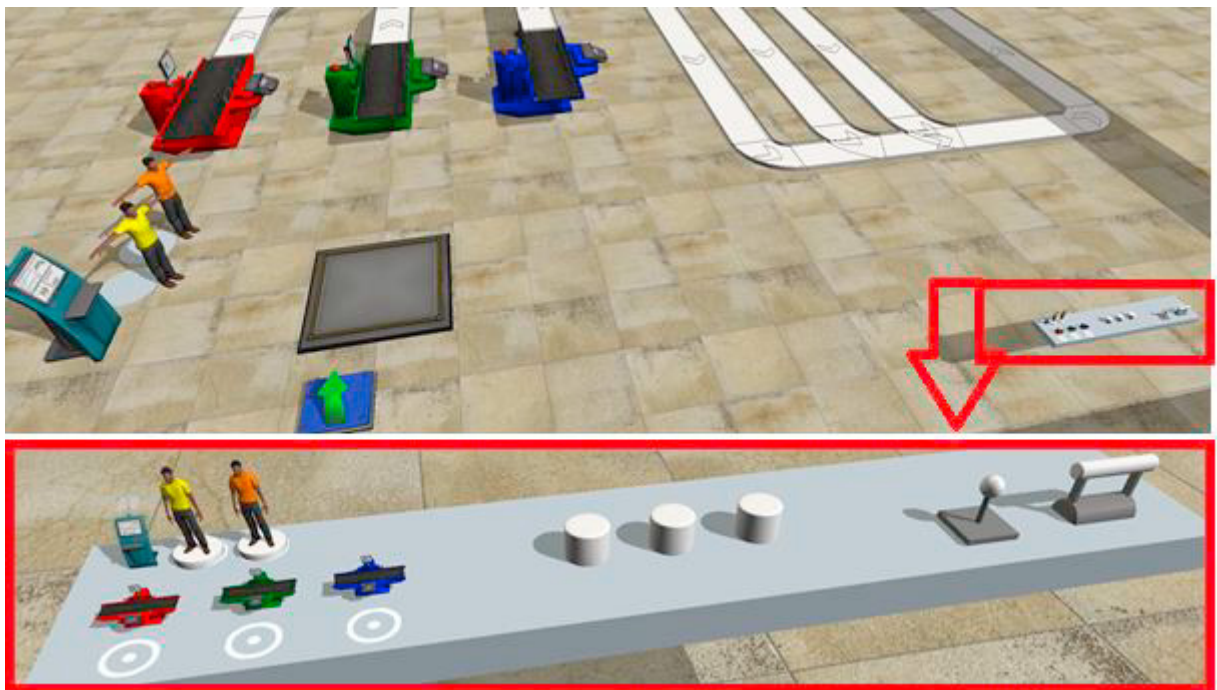


Fig. 3. Virtual interaction inside of production model could include several different possibilities. Handles, joysticks, buttons on/off triggers are possible. The challenge is to implement new functionality! The Model used this picture is a original FlexSim demo environment.

The link between ModelOnPreDraw and the objects is made by attributing variables to the different elements. For example Button 1, Button 2 and Button 3 are under the variable Buttons, whose definition is an array since many elements can behave in this way. It is important to mention that the ModelOnPreDraw code was entirely created for the demonstration model. When opening a new model no code at all is pre-written in the User Command dialog.

To create the MMU starting from a new and empty model seemed challenging at first sight, so the decision was made first to try to create simple objects and movements directly inside the demonstration model. In this way, it is possible to use the ModelOnPreDraw code and hence gain coding time. Once the functioning of the software understood, it is possible to start creating a new MMU model.

It is first necessary to create the shape of a new object to be inserted in a FlexSim model. The design of the model can be done in any computer-aided design software that can save or which output can be converted to a format supported by FlexSim. Once a compatible model is created, it is simple to “insert a Shape from the library” and define

the model to be used in the Shape properties as the custom model created. Later on, Solidworks software and the IGES format will be used to create interactive elements. It is important to know that it is not possible to import assemblies from Solidworks to FlexSim. When attempting to do so, the relations between the different parts (pivot, slide link, embedding) are lost, and as a result, the imported object will not function as intended.

The first step to adding custom object manipulation code is to insert an object, make it movable in the model and virtually grippable. Once these basic functions are understood, it becomes possible to implement more complex ways to manipulate custom objects virtually. It is useful to study the ModelOnPreDraw code and pick an object that already behaves in a similar way that is intended, such as a “Figurine” object which can be relocated in the model. The simplest way of doing the first step appeared to be setting the object created as a Figurine variable without any target association. To stay in the user's virtual hand, it was also necessary to introduce the label `_inHand` to the parameters of the new object. The `_inHand` code can be found in the ModelOnPreDraw in the figurine's section of the code.

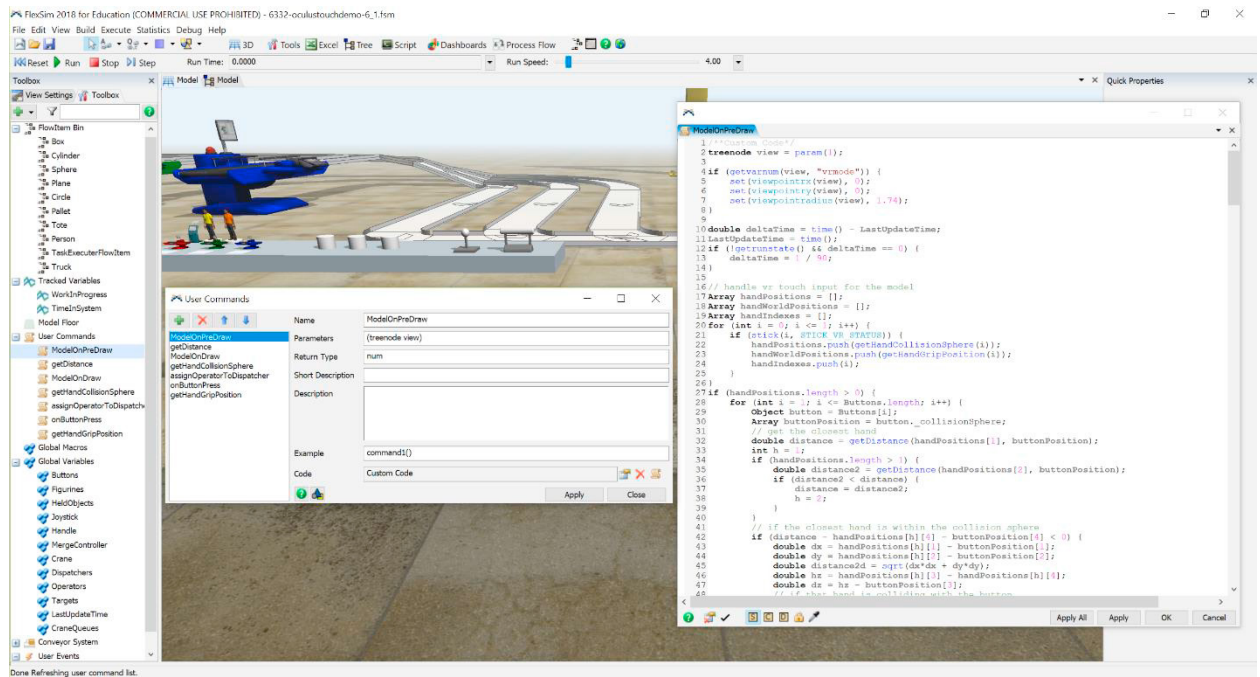


Fig. 4. The link between the ModelOnPreDraw and the objects is made by attributing variables to the different elements.

The second phase of our process is to create a simple object that behaves in a way that is not yet implemented in existing sample code. In the study, a door was selected as a sample object, as it is a necessary part of the MMU and it behaves rather similarly to the Handle object existing in the demo environment. Changed functionality included understanding parts rotating around different axes and having different movement limits. The door should also stay in the position it was released in, and it should be able to grip the door again in that position, whereas the Handle object would reset its position once released. The structure of a door is close to that of the Handle. In the Handle, the user can grab only the grip and not the lever of the object. Similarly, on the Door object, the user should grab the door handle and not the door itself. To create such an object, one must proceed according to the following steps:

- Model the part of the door that can be grabbed (e.g., with Solidworks) and save it in IGES format.
- Model the part(s) of the door that should move, but cannot be grabbed by the user and save it in IGES format.
- Insert the first model to the FlexSim model into the way described previously. As far as the model is concerned, this is the actual model to be actuated, and all custom functionality will be attached to this model.
- Open the object animation in the object parameters.
- In the animation insert the other part(s) and place them as appropriate. These objects will follow the movements of the first model.

- To define the center of rotation of the door, open the object tree and under the spatial parameters set the centroid number to 2 instead of 1. The following three parameters correspond to the X, Y and Z axis. Changing their value displaces the center of rotation.

Now the door element is created. The second step is to add on the ModelOnPreDraw the lines of code that indicate how the objects that are defined as “Door” variables in the model should behave. Parts of the Handle object code was recycled. The most important modification is that the speed of rotation that is changed to make the door follow the hand of the user’s movement and the removal of the part of the code that makes the handle come back to its initial position. The door behaves expectedly. Further improvements would include matching the object rotation limits to the physical model. In this study, the door rotation was changed to 90°.

With these previous explanations and test, the functioning of the software and the coding logic for virtual reality purposes are better understood. Now we can define the necessary steps and the process to build the MMU in FlexSim starting from an empty canvas.

5. Future work

The next step is to model and implement the remaining functions of the MMU in VR model. The implementation will be done using the FlexSim software. The most of the work will be done as study assignments by students in the Bachelor’s and Master’s programs of the university.

On February 12th, 2016, FlexSim announced the release of its first virtual reality compatible version, expected for March 28th, 2016 [16]. The VR compatibility allowed the use of the Oculus Rift and other compatible devices for a more immersive experience inside of used software environment, such as allowing the user to walk through the model and change the viewpoint more naturally. Since 2018, they added new functionality to the virtual reality manufacturing context. Now it is possible to interact with the environment using hand controllers, such as Oculus Rift and Oculus Touch multiple ways [16]. The hand controllers allow the manipulation and virtual operation of the modeled machine in a virtual reality once the functionality for touching and gripping parts is added to the Digital Twin and coupled to gestures and button-presses done by virtual reality controllers.

6. Conclusion

It is first necessary to define the relationship between the Oculus Touch hand controllers and the virtual environment. The simulation software needs to know where the hand controllers are and at which distance the virtual hands can touch the various objects in the simulation. Then, various elements can be created with design software such as SolidWorks and exported in a suitable format for FlexSim. A particular method is recommended to be followed to ease adding new functionality in the simulation. The behavior of the objects can be set by adding user-specified scripts. If multiple objects behave similarly, they should be coupled under the same variable type. The effect of the interactive actions on the active model must also be written about the code. It is not necessarily to add custom functionality for all objects in the simulation. If some objects must be added to the model with the only purpose to make the environment more realistic, it is possible to couple them with the models which do have implemented functionality. However, while the process of adding new functions in the model is clear, it is a challenge to find people with adequate programming skills to finish the implementation.

7. References

- [1] Grieves M., Vickers J. Digital Twin: Mitigating Unpredictable, Undesirable Emergent Behavior in Complex Systems. In: Kahlen J., Flumerfelt S., Alves A. (eds) *Transdisciplinary Perspectives on Complex Systems*. Springer, (2017)
- [2] Ramtilak, A., Joseph, A., Sivakumar, G., and Bhat, S., Digital Twin Spark Ignition for Improved Fuel Economy and Emissions on Four Stroke Engines, SAE Technical Paper 2005-26-008, (2005), doi:10.4271/2005-26-008.
- [3] Tuegel, E.J. , Ingraffea, A.R., Eason, T.G., and Spottswood, S.M., Reengineering Aircraft Structural Life Prediction Using a Digital Twin, *International Journal of Aerospace Engineering*, Volume 2011 (2011), Article ID 154798, 14 p. <http://dx.doi.org/10.1155/2011/154798>
- [4] Grieves, M. "Product Lifecycle Management: the new paradigm for enterprises." *Int. J. Product Development* 2 (2005) (Nos. 1/2): 71-84.

- [5] Hochhalter J, Leser WP, Newman, JA. et. al. Coupling damage-sensing particles to the digital twin concept. (2014) NASA Center for AeroSpace Information.
- [6] Reifsnider K, Majumdar P Multiphysics stimulated simulation digital twin methods for fleetmanagement. (2013) 54th AIAA/ASME ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference 1578.
- [7] Deloitte. Available from:<https://www2.deloitte.com/insights/us/en/focus/industry-4-0/digital-twin-technology-smart-factory.html>
- [8] Glaessgen E, Stargel D. The digital twin paradigm for futur NASA and US Air Force vehicles. (2012) 53rd AIAA/ASME/ASCE AHS/ASC Structures, Structural Dynamics and Materials Conference 20th AIAA/ASME/AHS Adaptive Structures Conference 14th AIAA 1818.
- [9] G.E., Modonia, M., Sacco, W., Terkaj, A Telemetry-driven Approach to Simulate Data-intensive Manufacturing Processes, *Procedia CIRP*, Volume 57, (2016), Pages 281-285.
- [10] Jay Lee, Edzel Lapira, Behrad Bagheri, Hung-an Kao. Recent advances and trends in predictive manufacturing systems in big data environment. *Manufacturing Letters* 1 (2013) 38–41.
- [11] ElMaraghy, H. Flexible and reconfigurable manufacturing systems paradigms. *International Journal of Flexible Manufacturing Systems*, 2005, 17(4), pp. 261-276.
- [12] Vainio, H. Aaltonen, J. Koskinen, K.T. Martinsuo, M. Simulation Models Supporting Multiple Assets Along the Product Life Cycle. In: Koskinen K. et al. (eds) *Proceedings of the 10th World Congress on Engineering Asset Management (WCEAM 2015)*. Lecture Notes in Mechanical Engineering. Springer verlag, (2016), p. 8.
- [13] Nykänen, O. Koskinen, K.T. Ranta, P. Salonen, J. Aaltonen, J. Nurmi, J. Virtuaalinen konelaboratorio ja semanttinen mallinnus konejärjestelmän suunnittelun tukena, Loppuraportti, Semogen II –hanke. Tampere. (2013). Tampere University of Technology. Smart Simulators research group. 46 p. + appendices 4 p.
- [14] Kádár, B. Lengyel, A. Monostori, L. Suginishi, Y. Pfeiffer, A. Nonaka, Y. Enhanced control of complex production structures by tight coupling of the digital and the physical worlds, *CIRP Annals-Manufacturing Technology* 59., 1 (2010), pp. 437-440
- [15] Terkaj, W. Tolio, T. Urgo. M. A virtual factory approach for in situ simulation to support production and maintenance planning. *CIRP Annals* 2017. *Manufacturing Technology*. Volume 64, Issue 1, (2015), Pages 451-454.
- [16] Lohtander, M. et. al. Micro Manufacturing Unit and Corresponding 3D-Model for a Digital Twin. 8th Swedish Production Symposium, SPS 2018, 16-18 May (2018), Stockholm, Sweden.