Engineering a Digital Twin for Manual Assembling

Alexandru Matei(✉), Nicolae-Adrian Ţocu, Constantin-Bălă Zamfirescu(✉), Arpad Gellert, and Mihai Neghină

Lucian Blaga University of Sibiu, Victoriei Blvd. 10, 550024 Sibiu, Romania
{alex.matei,nicolae.tocu,constantin.zamfirescu,arpad.gellert,mihai.neghina}@ulbsibiu.ro

Abstract. The paper synthesizes our preliminary work on developing a digital twin, with learning capabilities, for a system that includes cyber, physical, and social components. The system is an industrial workstation for manual assembly tasks that uses several machine learning models implemented as microservices in a hybrid architecture, a combination between the orchestrated and the event stream approaches. These models have either similar objectives but context-dependent performance, or matching functionalities when the results are fused to support real-life decisions. Some of the models are descriptive but easy to transform in inductive models with extra tuning effort, while others are purely inductive, requiring intrinsic connection with the real world.

Keywords: Manual assembling · Digital twin · Virtual simulation · Machine learning

1 Introduction

Due to their flexibility, human operators are pervasive in many factories where full automation is either unfeasible or too costly. Manual assembling is one of the manufacturing operations where humans are still playing the major role. Therefore, in the last decade there has been an increased interest from both academia and industry to develop intelligent assistance systems to support the assembling process. These systems are complex cyber-physical-social systems, with extended sensing capabilities of a working environment with physical, cyber, and human components. They should be able to recognize the product components and human features and actions, to learn patterns and correlate human operator contexts with the assembly states of a product, to assist in the correct product assembly by recommending the next step or by detecting the wrong ones, to train the human operator and so on.

The main challenge in the engineering of such systems is the complexity of integrating several sensors, with their own control capabilities, in a specific socio-technical context. The straightforward way to integrate these models is a virtual reality (VR) environment, enabling the creation of an artificial world for the manual assembling. The users are immersed in this artificial world with limited behavioral capacity, disconnected

T. Margaria and B. Steffen (Eds.): ISoLA 2020, LNCS 12479, pp. 140–152, 2021.
https://doi.org/10.1007/978-3-030-83723-5_10

from the real system. Over the years, this technology proved to be sufficient to address specific issues in many domains, but we are currently at a point where we need models to control the real systems which are not suitable for the artificial system. Consequently, this integration cannot be achieved completely in a digital way and requires perpetual correlations with the real world.

In contrast to VR, the Digital Twin (DT) concept emphasizes control models for the real system and not for the virtual ones. DT is a natural evolution of the Decision Support System concept which connects the models developed in the design phase with the IoT technology. In this way the problem-solving capabilities become an order of magnitude faster, with the additional capability to synchronize the models with reality. Therefore, the combination of DT and VR is used in many industrial applications, from simple monitoring tasks of an industrial equipment [1] to more complex tasks like fine-tuning the interactions of an operator with a robotic arm [2].

To faithfully reflect the real system along its entire life-cycle, a DT should exhibit some key characteristics [3]: 1) the ability to inspect the system at multiple levels, from system level to system of systems level; 2) the ability to transform, combine and establish equivalence between models; 3) the ability to integrate, add or replace models and the ability to describe the closeness to the physical system. Moreover, a DT needs to integrate the human's data and related context, either to assess the working conditions of humans in a factory [4] or to investigate the task allocation problem in human-robot collaborations [5]. Note that these desiderata are the key concerns in developing multi-model co-simulations as well.

The paper synthesizes the preliminary development of a DT with learning capabilities for a manual assembly workstation. Section 2 introduces the adopted architectural concept to engineer the DT. The underlying technologies and the microservices that are developed to provide the DT functionalities are described in Sect. 3. These services employ several machine learning models which are discussed in Sect. 4. Most of these models are descriptive but can easily be converted into inductive models with extra tuning effort. The last section highlights the current research and conclusions.

2 Assembly Workstation Digital Twin Concept

In the manufacturing industry, the implementation of a DT is following several methods that lack a common understanding, development methods and technological framework. The main concern in designing the assembly workstation is to maximize the reuse of control assets for both systems: physical and virtual. In this case, the DT will be a controller for both the physical and the virtual system that has predictive and adaptive capabilities. By following the guidelines presented in [6] to describe the DT concept, the main features for the manual assembly DT are the following:

- **Physical Entity** is the physical assembly workstation: a table frame with adjustable height for ergonomic use. The tabletop is a large smart tablet where the instructions are given to the user, either visually, audio or a combination of both. In case of heavy pieces that could potentially damage the screen of the tablet, an alternative would be to use a hard tabletop with a separate screen and a speaker.

- **Virtual Entity** is the VR application, which contains the virtual 3D model of the product together with the 3D model of the physical assembly workstation.
- **Physical Environment** is represented by the assembled modules on the assembly workstation and the human operator who is doing the assembly.
- **Virtual Environment** includes different VR rooms for specific needs, such as: simulation, tutorial, operator manual assembly training, product presentation, etc. The virtual environment is designed for a specific need. The link between the physical and virtual entities is made in the DT Scene.
- **State** composed of the variables needed to replicate in the virtual world the states of the following: physical entity (i.e. table height), physical environment (i.e. 3D position and orientation of the objects, assembly status), and human operator (i.e. emotion, eye tracking, skeleton model). In addition, the states of the control unit (i.e. the instructions that are presented) are considered as well. All these variables belong to a certain range that define the required levels of fidelity (e.g. high-fidelity for the physical entity, and medium to high fidelity for the physical environment).
- **Synchronization** presumes the bidirectional connections between the physical and virtual system together with the *twinning rate*. The *physical-to-virtual connection* is realized with various devices to acquire data about the current state of the physical objects (object types and their position with an RGB depth camera) and human operator (facial expressions and body movement with an additional camera, emotion detection with GSR sensors and/or voice recognition, intentional stance and attention with eye tracking glasses). All this data is transmitted to the VR application. The *virtual-to-physical connection* is restricted in the current implementation to the height adjustment for the physical table from the VR application. An assembly process restart or reset command from VR is possible only with the help of an external mediator who will manually rearrange the assembly parts on the physical table for correspondence with the VR scenarios. The *twinning rate* is 1 state update per second. Partial state updates can be made more frequently depending on the update frequency of each sensor.
- **Physical Processes** consist in guiding the human operator to assemble a product. The human operator must follow step by step instructions presented by the system.
- **Virtual Processes** are used for optimization, simulation, supervision, analysis, and improvement of the decision algorithms.

Note that the physical and virtual processes reflect intentions of the decision-makers, such as support human workers either for manual assembly training or real-time operation. They need to be sufficiently connected to the operational reality and complete to allow the execution of either in the physical or in the virtual space. Figure 1 depicts the main modules that are needed to control the physical or virtual assembly workstation. As mentioned, the main concern in designing the DT was to maximize the reuse of control assets for both systems (physical and virtual) while allowing early testing of the manual assembly training processes and algorithm, improving them side by side with the physical entity.

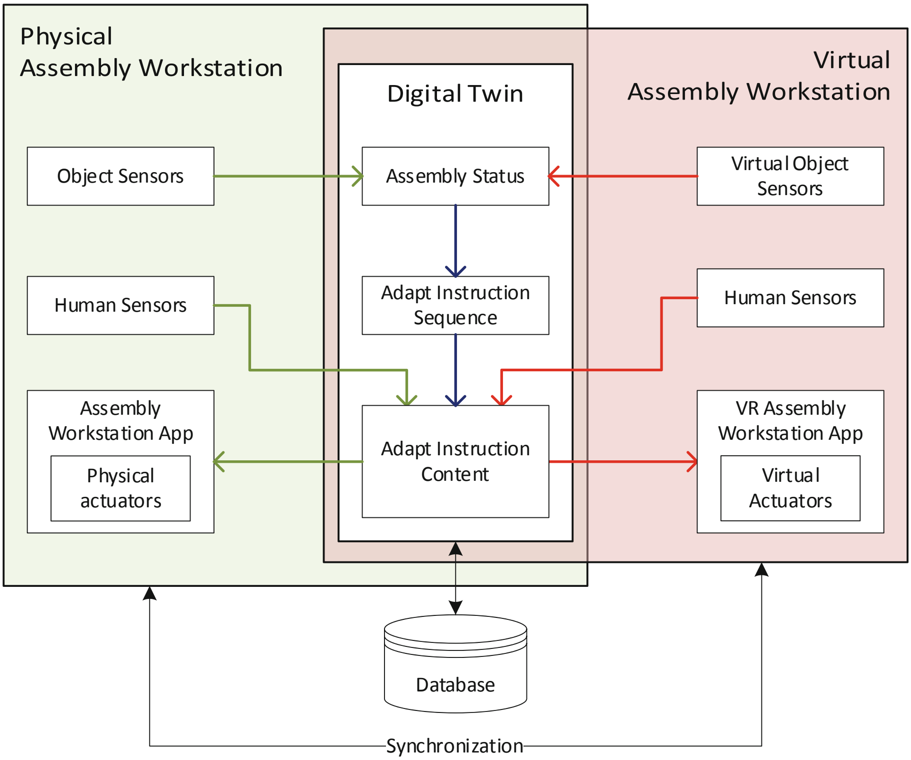

Fig. 1. The DT concept for the assembly workstation

The *Physical Assembly Workstation* has sensors to measure the physical system and environment, and actuators that are used by the Virtual Assembly Workstation to change the physical environment. The *Assembly Workstation Application* module is used to provide audio, video or written instructions to the user, highlight the assembly parts and control the physical actuators of the system. There are two types of sensors:

- *Object Sensors* are aimed at the objects that are assembled on the physical workstation. These sensors can be video cameras, depth cameras, lidars, etc. Another way is to use smart products or product tracing techniques. The data stream of these sensors will be used by the *Assembly Status* module.
- *Human Sensors* are used to detect changes in the user's emotion and intentions, where it is looking, height, and other characteristics. These sensors include video cameras, depth cameras, Galvanic Skin Response sensors, eye tracking sensors, etc.

The control elements shared between the physical and virtual assembly workstations are:

- *Assembly Status* module is responsible for identifying the current state of the assembly using input from the *Object Sensors* module. Based on that state, it must decide if it is a valid or invalid one. In other words, it detects if the user is making mistakes in the assembly process.
- *Adapt Instruction Sequence* module receives inputs from the *Assembly Status* module and recommends the next step that should be done in the assembly process. If the human operator is making mistakes, this module will repeat the current instruction and if not, it will move to the next one. When moving to the next instruction, the module will have to provide the optimal instruction from the available list of feasible ones.
- *Adapt Instruction Content* module decides on how to present the current instruction for a certain human operator. The instruction should be personalized by choosing the communication form (i.e. video, audio, text, or a combination of those). It should also decide the timing and amount of information given to the human operator.

In the case of the *Virtual Assembly Workstation*, all the data that is acquired in the *Physical Assembly Workstation* using *Object Sensors* will be available directly from the VR application through the SDK, without the need of additional sensors or equipment. We will consider this functionality of the SDK as virtual sensors – named *Virtual Object Sensors* in Fig. 1. Some of the user's information is also available using the VR SDK but this requires the VR equipment which will be considered as part of the *Human Sensors* in this case for the *Virtual Assembly Workstation*. There are some exceptions in the case of *Human Sensors* of the *Virtual Assembly Workstation* where some parameters cannot be extracted because the user's face is obstructed by the VR headset. The same sensors from the physical workstation could monitor the human operator and send the data to the VR application. However, these sensors are hard to use from the virtual side. Also, under some circumstances, their usage from the virtual side is unnecessary because the virtual environment is a controlled environment where most of the information is easily available in the software. Next, a correspondence between the sensors used to capture the physical environment and the ones used for the virtual environment is presented in Table 1.

The *database* is used to store the assemblies done on either the physical or the virtual assembly workstation for a later analysis and verification through replay. Using the recorded assembly processes, it is possible to continuously improve the Machine Learning (ML) algorithms of the system, especially the *Adapt Instruction Sequence* and *Adapt Instruction Content* modules.

Using the real time *synchronization* between the *Physical* and *Virtual Assembly Stations*, additional functionalities can be enabled: a trainer from VR can supervise the trainee from real medium using the DT or vice versa, remote operator manual assembly training, real-time manual assembly training analysis, etc. In the end, the *Virtual Assembly Station* is not acting only as a simulator, but also as a product for real-time visualization and analysis or testing new functionalities using real-time sensor input from the physical environment.

Table 1. Sensor correspondence

Behavior/Measurement	Physical station	Virtual station
Determine user's body characteristics: movement, skeleton data, height	Azure Kinect	Software-based using the existing VR equipment: headset, controllers, and trackers
Eye tracking and user intention	Tobii Pro Glasses 2	HTC Vive Pro Eye Series
Tracking the objects that are being assembled	RGB depth camera	Software-based as the position of all parts is always known in the virtual environment
Detection of the user's emotional state	Video camera, Microphone, GSR sensors	Face based is not possible, Microphone, GSR sensors

3 Implementation Issues

The concept of the physical workstation is presented in [7]. The software architecture of the system is based on a hybrid microservices architecture, a combination between orchestrated and event stream approaches. Having a hybrid architecture, allows for a greater flexibility in the development of the microservices and their interaction. The microservices are developed using gRPC[1], an open-source remote procedure call (RPC) framework. For monitoring, control, discovery, and health checking of the microservices we are using Consul[2], an open-source platform. The types of services that are currently available on the assembly workstation:

- *Physical Assembly Station:*

- **Table Height Adjustment** – interface that allows control of the physical table height.

- *Object Data:*

- **Object Detection** – allows identification of known objects in an image. Depending on the assembly scenario, a customized detection algorithm is needed for each object that is assembled. For a basic and fast detection, a bounding box algorithm like YOLO [8] might be enough but for a greater, pixel-level accuracy at the expense of speed, an instance segmentation algorithm like Mask R-CNN [9] should be used. In our implementation we opted for using a YOLO artificial neural network that was trained on demo objects.
- **Object Position** – allows identification of the XYZ position of the detected objects in a depth image based on the output of the Object Detection service.

[1] https://grpc.io/.

[2] https://www.consul.io/.

- **Object Segmentation** – allows the further segmentation of the detected object(s). It is used to extract 2D orientation and pixel-level segmentation of the detected objects. This is achieved using traditional image processing methods like: Otsu binarization, edge detection, contour detection, contour fill, etc.

- *Human Data:*

- **User Characteristics** – used to extract user information like height, age, gender. In [10] is presented the approach for this step of extracting these human characteristics.
- **User emotion detection based on voice** – uses a phase vocoder together with an artificial neural network. The method was trained and validated using the RAVDESS database [11] (Ryerson Audio-Visual Database of Emotional Speech and Song). Details about the approach can be found in [12].
- **User emotion based on facial expression** – is a microservice that is based on an input image with the user's face, identifies face landmarks. Based on the face landmarks and the distance between them, seven possible emotions can be predicted using an artificial neural network. The seven detected emotion are: angry, disgust, fear, happy, sad, surprise and neutral.
- **User Intention** – This microservice is using the eye tracking data to predict what the user wants to assembly next. The video feed together is feed into the Object Detection microservices and based on the gaze location it can be inferred if the user is looking at an object. Currently, the algorithm behind this microservice that will determine user fixation on an object or confusion if the user is looking around is under development.

- *Assembly Instruction Data:*

- **Correct Assembly** – used to determine if user follows the assembly steps correctly. This microservice, based on the object spatial position, orientation, and segmentation from Object Segmentation microservice can determine whether the pieces are assembled correctly or not.
- **Next Assembly Step** – used to provide the next suitable instruction based on the previous instruction. For this microservice, several types of predictors were tested, like two-level prediction table, Markov predictors, prediction by partial matching and long-term short memory artificial neural networks. Details about the implemented predictors can be found in [13, 14] and [15].
- **Adapt Assembly Step** – used to adapt the next instruction based of several factors: user state, mistakes made, assembly state, etc. This microservice is currently under development.

- *Other microservices:*

- **Publish-Subscribe** – allows for a pub-sub communication/event stream-based alternative to direct RPC calls (orchestrated) between microservices.
- **Video Streaming** – this microservice allows viewing the stream of any video source (including screen) connected to the assembly station. To reduce the bandwidth, the video is H264 encoded on the server side using the Windows Media Foundation SDK.

The DT of the physical assembly workstation is an adapted VR simulation. The VR simulator was developed in Unity 3D and it is compatible with Oculus Rift and HTC Vive headsets. The compatibility problem was easily solved using the VRTK Toolkit. This toolkit is a collection of scripts and prefabs made for Unity and VR. Using VRTK and Steam VR we could use the same classes and events to access the controller's apps from both hardware. The VR simulator is further detailed in [16]. In Fig. 2 the physical assembly workstation and its virtual representation in the VR application are shown.

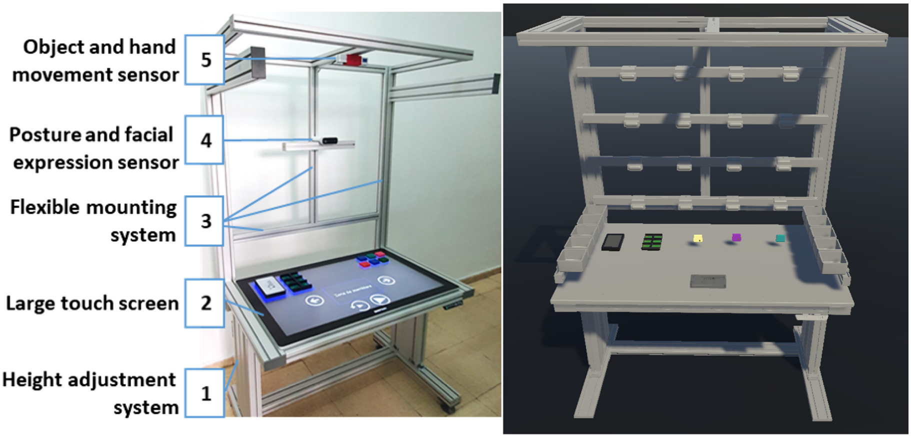

Fig. 2. Physical prototype [15] and its DT

4 Machine Learning Capabilities

As described in the previous section, the DT for assembly workstation combines multiple ML models developed during the design process. There are three broad categories employed in the current developments: 1) human emotion recognition, 2) context predictors for the assembling sequences, and 3) traditional image processing methods for object recognition. In the following sections the first two categories are discussed, whereas the third uses classical image processing algorithm with no need for data to improve their object recognition capability in real-time.

4.1 Human Emotions

Human emotions are known to play a significant role in human behavior and are an essential source of data for improving and adapting human interaction. There are multiple ways of recognizing human emotions, through facial expressions in [17] and [18], speech in [19] and [20], or biometric signals in [21] and [22]. For both the audio (speech) and video (face) channels, the detection of human emotions involves a pre-processing and feature extraction stage and a classification stage. The main features isolated in the audio channel are spectral, the Mel-frequency cepstral coefficients, although other time-based features such as short-term energy, zero-crossing and cross-correlation coefficients for pitch detection are additionally used. The pre-processing of the still images from the video channel revolves around the face detection and the extraction of discriminative features from salient face regions, as well as the history of those features from recent frames. For both the audio and video channel, the classifier chosen is an artificial neural network (ANN). Although there are many approaches, ANNs have been shown to be the most promising of artificial intelligence (AI) techniques, having good results in general pattern recognition tasks.

In the case of the biometric signals, only the galvanic skin response (GSR) is measured, as the heart rate reactions are relatively slow and persist for a long time (considering the general manual assembly training scenario), while EEG sensors are a lot less practical for an assembly workstation. Unlike the audio and image methods, which attempt full emotion identification, the GSR processing consists of adaptive filtering and peak detection, mainly used to identify emotion excitation. However, this approach has the big drawback of not being able to detect the explicit emotion, as the GSR peak can be triggered both by positive (e.g. happy) and negative (e.g. fear) emotions.

Each human emotion investigation channel has its strengths and weaknesses: face caption and GSR are continuous, whereas speech is voluntary; in contrast, the lack of speech inflexions despite emotions is less common than poker-face expressions, while physiological responses to emotions are almost never controlled; audio and video data can be gathered at a distance, whereas GSR, EEG and other biometric measurements require equipping the user. The combination of the methods thus offers a better chance of correctly identifying the true mix of emotions for the observed user and thus provides better data for the workstation to adapt and improve the instructions and feedback provided during the manual assembly training.

ANN methods for detecting human emotions are descriptive models trained on large sets of audio and image (or video) data. However, given the opportunity, they are easily transformed into inductive models through continuous training and refinement of the network weights based on data acquired during the operation. The assembly workstation offers the opportunity of observing the same user during multiple operations, thus enabling both the general refinement and the adaptation to the particularities of the user. In contrast, the GSR peak detection is implicitly an inductive model, adapting its filtering and thresholds to the user to compensate for variability of the physiological responses from human to human.

Although not an emotion, confusion of the user is another important state of mind that, if identified correctly, would greatly improve the performance of the assembling instructions. For the detection of this state, data from all channels may be combined with information from gaze-tracking glasses to provide valuable insights in the human state of mind.

As human emotions are personal, and represent internal biological states, questions about data privacy and ethics arise. Regarding this issue, the ANNs presented are trained on datasets that are freely available in the public domain. Also, the inferred emotional state of the user is used only momentarily by the software application and is not stored anywhere by the system.

4.2 Context-Based Predictors for the Assembly Sequences

In our previous works [13, 14] and [15] we investigated different models for context predictors, such us two-level prediction table, Markov predictors, prediction by partial matching and even long short-term memory artificial neural networks. These models have the same objective of providing adaptive assembly assistance by dynamically adapting the assembling process to the human operator's actual condition, his/her general characteristics, preferences, and behaviors in assembling products. These models are using a context-based predictor to recommend the next assembly step based on the current state of both: the semi-product, and the worker.

Being pre-trained with a set of rules extracted from a dataset of product assembly sequences we can consider them to be descriptive. The two-level context-based predictor from [13] is able to reproduce the assembly step which was last seen after a certain context, whereas the Markov predictors from [14] and [15] can provide multiple choices for the next assembly step, in their descending probability order. All these models can be enhanced with run-time training. Thus, after the pre-training, during the exploitation of the predictors, they can be updated after each assembly step. We expect that with run-time learning, the predictor can cover a higher number of situations.

In the case of the Markov predictors presented in [14], the next state probabilities for a given context are estimated using the state occurrence frequencies in that certain context. Currently, since only a limited pre-training stage is applied, these frequencies are maintained as simple counters. When a prediction is to be made for a given context, the state having the maximum counter can be provided as the most probable next state. If we extend the prediction mechanism with run-time learning, some of these counters can increase a lot in time, reaching very high magnitudes, while others can remain on low levels. Therefore, saturating counters would be more appropriate. The saturating counters, whose magnitude is limited, can adapt faster to changes in the behavior of a certain human operator, and can easily adapt also to different users. Obviously, this fast adaptation could be assured by increasing the saturating counter associated to the correct next state and by decrementing the saturating counters corresponding to wrong states, after each assembly step.

We are currently developing a prediction algorithm based on pre-trained Hidden Markov Models (HMM). That prediction scheme can be easily adapted for run-time learning by periodically adjusting the HMM on a certain window of assembly steps. Another predictor which is currently in evaluation is a Long Short-Term Memory (LSTM). It is pre-trained through a certain number of epochs. We intend to analyze the influence of a possible run-time learning over the LSTM's prediction accuracy. However, since the optimal number of epochs in the pre-training stage is 5000, we expect a slower adaptation capability for the LSTM by run-time learning with respect to the above-mentioned Markov and HMM prediction schemes.

The experiments with these models revealed that:

- If the dataset is strictly restricted to the assembly behavior of the human operator there is no significant difference if the dataset used to pre-train the ML models is generated either from physical or virtual assembly workstation. This finding may significantly speed up the pre-training of ML models.

- There will be always a tradeoff between adaptability and prediction accuracy among alternative ML models. Some works well with a large pre-training dataset, but they have slower adaptability to real-time data coming from the physical system. Therefore, choosing the optimal model depends very much on the contextual use (i.e. products diversity, assembling complexity in terms of number of sequential steps and alternative choices, etc.).
- The ML models developed to predict the assembly behavior trained with data from the virtual workspace has limited applicability. We have found strong correlations between the assembling performance of some processes with the operators' psychomotor capabilities, such as the gender, if he/she is wearing glasses, tiredness, height, etc. While it is clear that the data coming from the real space are limited due to lack of high volume of data and slower generation of data, models developed for the virtual space cannot cope with the entire spectrum of real-life data as a result of limited sensorial capabilities. Consequently, left-over parts that should be considered by the DT in extending the descriptive models into inductive ones will always remain.

5 Conclusions and Future Work

The paper synthetized the preliminary developments of a DT for a manual assembly workstation. This cyber-physical-social system employs several ML models implemented as microservices in a hybrid architecture, a combination between the orchestrated and the event stream approaches. For a fast and intuitive integration of different ML models a VR application have been twinned with the physical workstation. In addition, it was used to generate datasets to pre-train the ML models. These models analyzed in this paper have either similar objectives but context-dependent performance (i.e. context-based predictors for the assembly sequence), or matching functionalities when the results are fused to support real-life decisions (i.e. detection of human emotions). Some of the models are descriptive but easy to transform in inductive models with extra tuning effort, while others are purely inductive requiring intrinsic connection with the real world.

These results will be further exploited on at least two directions. Firstly, the prediction models investigated to suggest the next assembly step can be easily reverted and used to simulate the assembly process of a human operator. The lack of a reliable model for the cognitive behavior of a human operator in assembly tasks was the main concern in not building at design time a co-simulation. The models discussed in the previous section will be further used in a co-simulation to reproduce the user's behavior. This will increase the speed for an extended design-space exploration when new microservices will be added. Moreover, coupled with specific real-time user data (i.e. gender, if he/she is wearing glasses, tiredness, height, etc.) there is the potential to have personalized design-space exploration capabilities for various assembly tasks, limiting substantially the combinatorial complexity arising from the interactions of multiple models. Secondly, all the ML models reported in the paper were pre-trained with datasets obtained from laboratory experiments with students. Real-life experiments with human operators from industry are envisaged. These experiments will provide a better insight on how to employ the alternative models in different contexts.

Acknowledgements. This work is supported through the DiFiCIL project (contract no. 69/08.09.2016, ID P_37_771, web: http://dificil.grants.ulbsibiu.ro), co-funded by ERDF through the Competitiveness Operational Programme 2014–2020.

References

1. Schroeder, G., et al.: Visualising the digital twin using web services and augmented reality. In: IEEE 14th International Conference on Industrial Informatics (INDIN), Poitiers, pp. 522–527 (2016)
2. Havard, V., Jeanne, B., Lacomblez, M., Baudry, D.: Digital twin and virtual reality: a co-simulation environment for design and assessment of industrial workstations. Prod. Manuf. Res. **7**(1), 472–489 (2019)
3. Schleich, B., Anwer, N., Mathieu, L., Wartzack, S.: Shaping the digital twin for design and production engineering. CIRP Ann. Manuf. Technol. **66**(1), 141–144 (2017)
4. Lu, Y., Liu, C., Wang, K.I.-K., Huang, H., Xu, X.: Digital twin-driven smart manufacturing: connotation, reference model, applications and research issues. Robot. Comput. Int. Manuf. **61**, 101837 (2020)
5. Bilberg, A., Malik, A.A.: Digital twin driven human-robot collaborative assembly. CIRP Ann. Manuf. Technol. **68**(1), 499–502 (2019)
6. Jones, D., Snider, C., Nassehi, A., Yon, J., Hicks, B.: Characterising the digital twin a systematic literature review. CIRP J. Manuf. Sci. Technol. **29**(A), 36–52 (2020)
7. Pîrvu, B.C.: Conceptual overview of an anthropocentric training station for manual operations in production. In: Balkan Region Conference on Engineering and Business Education, vol. 1, no. 1, pp. 362–368 (2019)
8. Redmon, J., Divvala, S., Girshick, R., Farhadi, A.: You only look once: unified, real-time object detection. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, pp. 779–788. IEEE (2016)
9. He, K., Gkioxari, G., Dollár, P., Girshick, R.: Mask R-CNN. In: 2017 IEEE International Conference on Computer Vision (ICCV), Venice, Italy, pp. 2980–2988. IEEE (2017)
10. Cruceat, A.M., Matei, A., Pîrvu, B.C., Butean, A.: Extracting human features to enhance the user experience on a training station for manual operations. Int. J. User Syst. Interaction **12**(1), 54–66 (2019)
11. Livingstone, S.R., Russo, F.A.: The Ryerson audio-visual database of emotional speech and song (RAVDESS): a dynamic, multimodal set of facial and vocal expressions in North American English. PLoS ONE **13**(5), e0196391 (2018)
12. Govoreanu, V.C., Neghină, M.: Speech emotion recognition method using time-stretching in the preprocessing phase and artificial neural network classifiers. In: 2020 IEEE 16th International Conference on Intelligent Computer Communication and Processing (ICCP), Cluj-Napoca, Romania, pp. 69–74. IEEE (2020)
13. Gellert, A., Zamfirescu, C.B.: Using two-level context-based predictors for assembly assistance in smart factories. In: 8th International Conference on Computers Communications and Control, Oradea, Romania (2020)
14. Gellert, A., Zamfirescu, C.B.: Assembly support systems using Markov predictors in smart factories. In: 20th Open Conference of the IFIP WG 8.3 on Decision Support, Wrocław, Poland (2020)
15. Gellert, A., Precup, S.A., Pirvu, B.C., Zamfirescu, C.B.: Prediction-based assembly assistance system. In: 25th International Conference on Emerging Technologies and Factory Automation, Vienna, Austria (2020)

16. Țocu, N.A., Gellert, A., Ștefan, I.R., Nițescu, T.M., Luca, G.A.: The impact of virtual reality simulators in manufacturing industry. In: 12th International Conference on Education and New Learning Technologies (2020)
17. Dudul, S.V., Kharat, G.U.: Emotion recognition from facial expression using neural networks. In: 2008 Conference on Human System Interactions, Krakow, pp. 422–427. IEEE (2008)
18. Khanal, S.R., Barroso, J., Lopes, N., Sampaio, J., Filipe, V.: Performance analysis of Microsoft's and Google's emotion recognition API using pose-invariant faces. In: DSAI 2018: Proceedings of the 8th International Conference on Software Development and Technologies for Enhancing Accessibility and Fighting Info-Exclusion, Thessaloniki, Greece, pp. 172–178. ACM (2018)
19. Tóth, S.L., Sztahó, D., Vicsi, K.: Speech emotion perception by human and machine. In: Esposito, A., Bourbakis, N.G., Avouris, N., Hatzilygeroudis, I. (eds.) Verbal and Nonverbal Features of Human-Human and Human-Machine Interaction. LNCS (LNAI), vol. 5042, pp. 213–224. Springer, Heidelberg (2008). https://doi.org/10.1007/978-3-540-70872-8_16
20. Sezgin, M.C., Gunsel, B., Kurt, G.K.: Perceptual audio features for emotion detection. EURASIP J. Audio Speech Music Process. **2012**(1), 1–21 (2012). https://doi.org/10.1186/1687-4722-2012-16
21. Tarnowski, P., Kołodziej, M., Majkowski, A., Rak, R.J.: Combined analysis of GSR and EEG signals for emotion recognition. In: 2018 International Interdisciplinary PhD Workshop (IIPhDW), Swinoujście, pp. 137–141. IEEE (2018)
22. Wu, G., Liu, G., Hao, M.: The analysis of emotion recognition from GSR based on PSO. In: 2010 International Symposium on Intelligence Information Processing and Trusted Computing, Huanggang, pp. 360–363. IEEE (2010)