

11th Conference on Learning Factories, CLF2021

## Learning Factories On Demand

Günter Bitsch<sup>a,\*</sup>, Pascal Senjic<sup>a</sup>

<sup>a</sup>ESB Business School, Reutlingen University, 72762 Reutlingen, Germany

---

### Abstract

Learning Factories are research and learning environments that demonstrate new concepts and technologies for the industry in a practical environment. The interaction between physical and virtual components is a central aspect. The mediation and presentation usually occur directly in the learning factory and are thus limited in time and concerning the user group. A learning factory- on-demand- can be provided by dividing and virtualizing the individual components via containers and microservices. This enables both local operation and operation hybrid cloud or cloud systems. Physical components can be mapped either through standardized interfaces or suitable emulators. Using the example of the Learning Factory at Reutlingen University (Werk150), it will be shown how different use cases can be made available utilizing software-based orchestration, thus promoting broader and more independent teaching.

© 2021 The Authors. This is an open access article.

Peer Review statement: Peer-review under responsibility of the scientific committee of the 11th Conference on Learning Factories 2021.

*Keywords:* virtualization;microservice;containers

---

### 1. Introduction

In recent years, more and more individual learning factories have emerged, which provide a realistic learning and training environment in the context of teaching and research [1, 2]. In this environment, they provide opportunities to use new concepts and technologies for the various application purposes in education, training, and research. In the context of Industry 4.0, these include applications that are developed into smart solutions with the help of cyberphysical systems (CPS) connected via the Internet of Things (IoT) [3]. However, these environments usually require a local configuration [4]. For example, only six of 31 of the best practice learning factories support learning environments with enhanced virtual support, and only one of the learning factories supports a purely virtual learning environment [2].

Due to this local focus, potentials are not used, in that neither the creation of individual learning paths is sufficiently well supported nor the desired scalability can be achieved [1]. Further requirements for an overlapping or cooperative use, which exist in the context of learning factories (see [5]), would therefore not be realizable. However, advancing digitization and the IoT have created the prerequisites for digitizing the technical components involved in learning factories in such a way that these extended requirements can additionally be met in the sense of a hybrid approach for learning factories [6-8]. This is the basis for the learning factory on-demand approach presented here.

This article presents an approach to virtualizing a learning factory environment. This shows how the components can be subdivided into so-called microservices and virtualized using a modular structure. Digitization allows

---

\* Corresponding author. Tel.: +49-7121-271-3079; fax: +49-7121-271-903079

*E-mail address:* guenter.bitsch@reutlingen-university.de

interfaces to be created for the physical components, which can also be virtualized using suitable emulators (for differentiation between emulators and simulators, see [9]).

Through the learning factory on-demand, the local centering of the learning factory is to be dissolved and made available in demonstrations as well as research and teaching scenarios.

## 2. State of the Art

In this section, the learning factory on-demand is classified. Furthermore, exemplary approaches to the virtualization of the learning factory and the modularization towards microservices are described.

### 2.1. Classification of the Learning Factory On-Demand

Learning factories can be distinguished in a broader or narrower sense [1]. Figure 1 illustrates this distinction. Originally, the learning and research environment was placed with a focus on the real-physical environment [10].

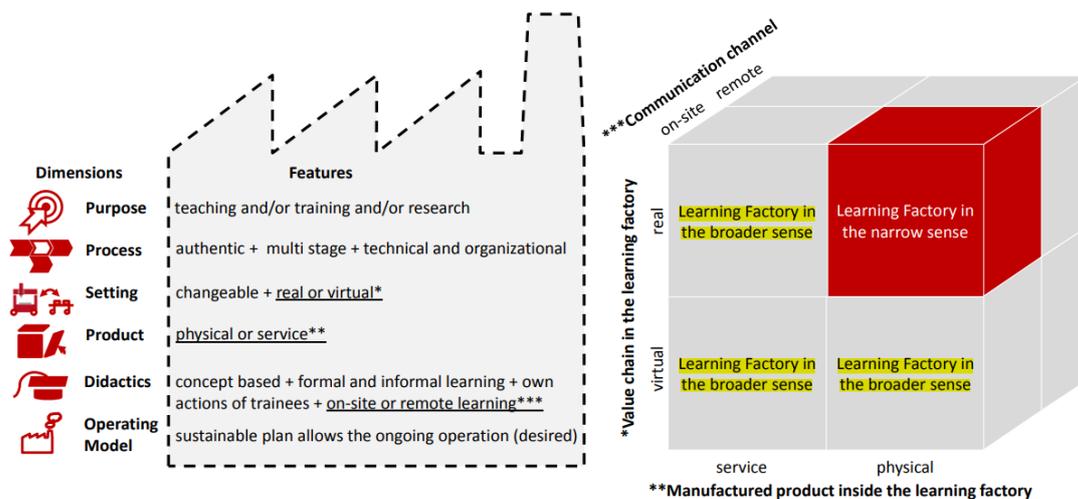


Fig. 1. Features and differentiation of learning factories in the broader and narrower sense [1].

In the further course, the learning factories were extended by the digital factory components, so 29 of the 31 best practices support this integration. The hybrid learning factory's approach is to create a matching virtual representation for physical learning factories based on a digital data set, which allows the user to combine both approaches and thus enables digital planning and virtual validation related to the "physical work execution" [11]. To open up further user groups and support the transfer into operational practice, learning factories can also be operated as teaching factories or remotely accessible learning factories with the two operation modes factory-to-classroom and lab-to-factory [12].

The two extended approaches expand the user groups and offer additional learning fields but are still limited by the local centering on the learning factory. With the learning factory on-demand approach, the infrastructure is to be made available so that it can be operated largely independently of the physical learning factory. Individual learning steps can be carried out both virtually and physically. This makes it possible, for example, to work independently of time and space, analogous to virtual commissioning [13], with subsequent evaluation on the physical infrastructure.

Furthermore, the implementation of a virtual system allows the creation of individual learning paths, which can be specifically addressed to the individual learning progress [14].

### 2.2. Virtualization of the learning factory

Digital and virtual factories are essential components of Industry 4.0 and thus also of modern learning factories. An important enabler for this is the digital twin [15]. However, this requires a strong coupling and interaction between the digital and physical representation. The digital shadow is less strongly coupled. Both concepts, however, are based on a real component. The virtual factory in virtual reality (VR) represents a different approach here [16]. These virtualizations refer to the learning factory's content but not to the configuration or the operational

platform of the learning factory itself. With the learning factory on-demand, self-referential virtualization of the operational process configuration takes place. The goal is the structural introduction of a layer for the operation platform of the learning factory. Through virtualization, a standardized structure can be created by securing and reproducing results, which on the one hand allows for analyzability, e.g., for evaluation, and on the other hand, enables possibilities for building individual learning paths or adaptive processes [6]. After virtualization, the technical realization for this is provided by the orchestration through software containers [17].

### 2.3. *Microservices*

Microservices is an architecture pattern that describes the dissolution of monolithic structures into task-oriented services [18]. Sollfrank et al. [19] evaluate the use of Docker structures in the context of CPS. Docker containers represent a manifestation of microservices development. Omar et al. [20] show how to build and deploy a Fog computing environment using Docker technology. Ciavotta et al. [21] have shown in their work how a microservice architecture via Docker can be built for a digital factory. Merz et al. [22] provide an approach for cloud-based research. Among other things, they show how software components can be developed as microservices and thus easily distributed.

With the approach of the learning factory on-demand, in addition to the existing software components, the physical elements are also virtualized and mapped as microservices. Thus, emulations are available for the physical objects, enabling different operating modes (physical, hybrid, virtual) to be operated.

## 3. **Transfer from local learning factory to learning factory on-demand**

To transform the learning factory into a learning factory on-demand, the following approach was chosen:

1. check prerequisites
2. identify components
3. create necessary emulators for CPS
4. splitting of the components into microservice containers
5. deployment in different runtime scenarios

The goal is to provide a flexible infrastructure for the designed learning factory. The runtime environment can include on-premise, cloud, or hybrid cloud structures. Another goal is to realize the distributability and reproducibility of results within the framework of the learning factory, where the operation platform can be provided and commissioned on demand.

### 3.1. *Prerequisites*

The following preconditions must be fulfilled for the Learning Factory On-Demand:

1. inter-process communication systems (e.g., IoT messaging systems)
2. virtualization platform (e.g., Docker Runtime)

This approach assumes that all components of the learning factory are connected via a uniform communication platform. This represents the idea of Fog Computing, which is already present in Industry 4.0 concepts (see e.g., [23] and [24]).

The learning factory on-demand is based on software containers with the corresponding run-time environment. When orchestrating the containers, the complexity that arises must already be considered structurally at an early stage, see here Hoque et al. [25]. If these prerequisites are met, the next step is to analyze the components of the learning factory.

### 3.2. *Identify components*

The components of the learning factory must be identified and analyzed concerning their interfaces. There are different components within the learning factory: software components and hardware components. Software components include, for example, APS or ERP systems, web stores, databases, indoor localization, and more. Hardware components include robots, sensors, 3D printers, CPS, and others.

A common communication platform was established as a prerequisite for all components. The communication platform must enable the structurally uniform definition of the interfaces and messages to be exchanged. On this basis, the interfaces of the components are analyzed and defined. It is important here that these interfaces are valid across all platforms. An example of this can be found in Madiwalar et al. [26], who defines a "plug and produce" environment for hardware components via OPC UA.

### 3.3. Create necessary emulators

Once the components have been identified, and the component interfaces described, the next step is to provide emulators for all the components to be virtualized. The emulators must reproduce at least the logical behavior of the component. Furthermore, the emulators must be able to be integrated into the learning factory and be exchanged by the real component. For CPS, González-Nalda et al. [27] have described an approach for using Robot Operating Systems (ROS) and Docker to create the environment of a modular architecture for this type of component.

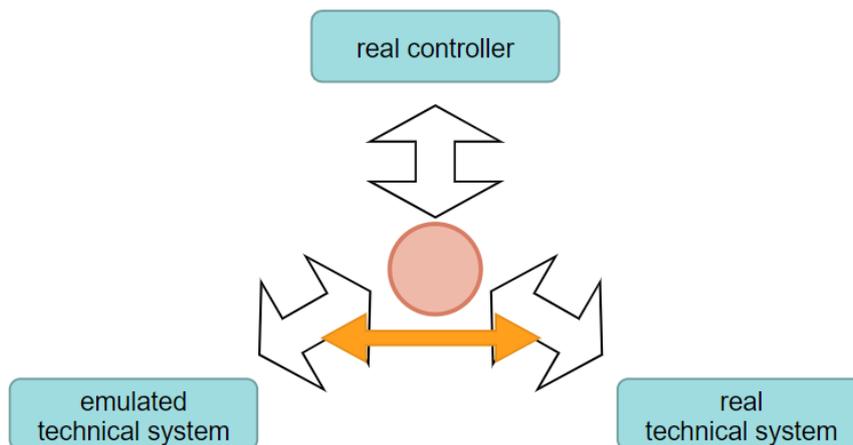


Fig. 2. Integration of the real control system with emulators according to [13].

### 3.4. Splitting of the components into microservice containers

After all of the components are known and the necessary emulators are provided, the next step is to divide the learning factory into small modules. In the process, the components are broken down into small task-oriented packages. These packages are then packed into software containers. Each container contains an individual microservice, which can now be flexibly distributed.

### 3.5. Deployment in different runtime scenarios

By providing the modules of the learning factory as software containers, flexible runtime environments can be realized. As long as the target environment supports the virtualization platform, the containers can be operated in different constellations. Supported scenarios are cloud, hybrid cloud, and on-premise structures.

The container images also enable the versioning of the components. Due to their distributability and reusability, the containers allow an environment in which research results can be evaluated through a consistent infrastructure.

## 4. Use case Werk150

For the Werk150 learning factory, the transformation process to build the emulated and remote environment was carried out in the following steps.

In the first step, the information systems were transferred to the Docker container virtualization platform. The system components (database storage MS-SQLServer, middleware platform Oneiroi) and the application components (webshop nopcommerce, self-execution system components scheduling, prioritizer, logistics, data-collection) were implemented.

In the second step, the emulators for the CPS were built. Specifically, emulators were built for the FLEXLOG conveyor line, the pick-by-light and the UR10 cobots used.

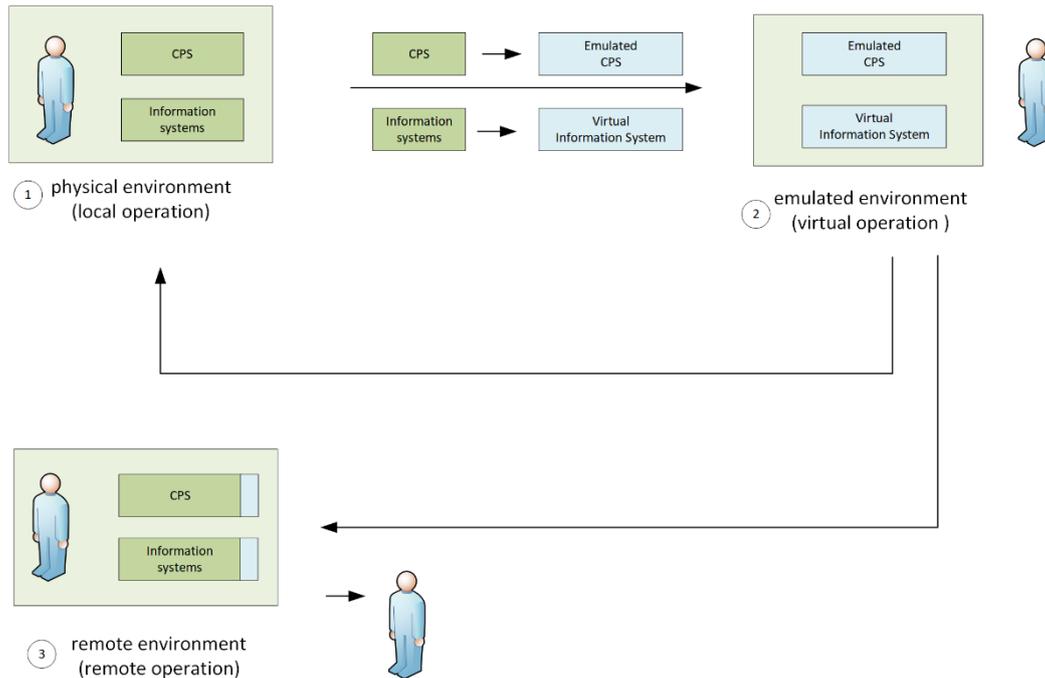


Fig. 3. applied transformation process from physical to emulated/remote environment.

Once fully implemented, the components can now be deployed for operation either on-premise or via the cloud (Microsoft Azure) or in a hybrid mode.

At Werk150, where all components are fully available, the physical environment is deployed on the internal network. For remote training of the used process data collection software (PDC) and the operation of the web store within the scenario, the CPS emulators are used to emulate the physical process.

## 5. Conclusion and outlook

In this article, the concept of the learning factory on-demand was presented. It showed how a locally-focused learning factory could be virtualized with state-of-the-art technologies and achieve increased reusability and scalability. For this purpose, the monolithic structure of the locally real learning factory was dissolved and divided up in a task-oriented manner with microservices' help. This creates an architecture that enables a flexible infrastructure for the learning factory. The flexible runtime environment allows the learning factory to be deployed in different environments as needed. This enables a demonstration environment that can run from the real plant to the laptop. Furthermore, results from the learning factory can be evaluated in the same environment and thus offer enrichment in scientific workflows [17].

With this work, the basis is created to create scenarios for learning factories and orchestrate them for the runtime environment. A basic structure is created through the containers, which can create interaction and tests between different learning factories. A possible container registry for the different learning factories could be the next step. The distribution and deployment of software containers also allow providing scenarios for interested parties easily.

## References

- [1] E. Abele et al., Learning factories for research, education, and training, *Procedia CiRp*, 32 (2015) 1-6.
- [2] E. Abele, J. Metternich, and M. Tisch, *Learning Factories. Concepts, Guidelines, Best-Practice Examples*. Springer, (2019).
- [3] A. Mosler, *Integrierte Unternehmensplanung*. Springer, (2016).
- [4] T. Al-Geddawy, A Digital Twin Creation Method for an Opensource Low-cost Changeable Learning Factory, *Procedia Manufacturing*, 51 (2020) 1799-1805.
- [5] J. Enke et al., Cooperation between Learning Factories: Approach and Example, *Procedia Manufacturing*, 45 (2020) 222-227.
- [6] S. Edwards, The Opportunity for learning factories in the UK. A report to the Gatsby Foundation, (2021).

- [7] A. Ullrich, J. Enke, M. Teichmann, A. Kreß, and N. Gronau, Audit-and then what? A roadmap for digitization of learning factories, *Procedia Manufacturing*, 31 (2019) 162-168.
- [8] G. Schuh, J.-P. Prote, S. Dany, S. Cremer, and M. Molitor, Classification of a hybrid production infrastructure in a learning factory morphology, *Procedia Manufacturing*, 9 (2017) 17-24.
- [9] I. McGregor, The relationship between simulation and emulation, in *Proceedings of the Winter Simulation Conference*, IEEE. 2 (2002), 1683-1688.
- [10] M. Steffen, D. May, and J. Deuse, The Industrial Engineering Laboratory, in *Proceedings of the 2012 IEEE Global Engineering Education Conference (EDUCON)*, 17-20 April 2012 (2012) 1-10, doi: 10.1109/EDUCON.2012.6201098.
- [11] E. Abele et al., Learning factories for future oriented research and education in manufacturing, *CIRP annals*. 66 (2) (2017) 803-826.
- [12] D. Mavrikios, K. Georgoulas, and G. Chryssoulouris, The teaching factory paradigm: Developments and outlook, *Procedia Manufacturing*, 23 (2018) 1-6.
- [13] T. Meyer and U. Grillitsch, Virtuelle Inbetriebnahme mittels Ablaufsimulation in der Automobilindustrie, in *Ablaufsimulation in der Automobilindustrie*, G. Mayer, C. Pöge, S. Spieckermann, and S. Wenzel Eds. Berlin, Heidelberg: Springer Berlin Heidelberg. (2020) 275-287.
- [14] N. Tvenge and K. Martinsen, Integration of digital learning in industry 4.0, *Procedia Manufacturing*. 23 (2018) 261-266, doi: <https://doi.org/10.1016/j.promfg.2018.04.027>.
- [15] N. Tvenge, O. Ogorodnyk, N. P. Østbø, and K. Martinsen, Added value of a virtual approach to simulation-based learning in a manufacturing learning factory, *Procedia CIRP*. 88 (2020) 36-41.
- [16] T. Riemann, A. Kreß, L. Roth, S. Klipfel, J. Metternich, and P. Grell, Agile Implementation of Virtual Reality in Learning Factories, *Procedia Manufacturing*. 45 (2020) 1-6.
- [17] W. Gerlach, W. Tang, A. Wilke, D. Olson, and F. Meyer, Container Orchestration for Scientific Workflows, in *2015 IEEE International Conference on Cloud Engineering*. 9-13 March 2015 (2015) 377-378, doi: 10.1109/IC2E.2015.87.
- [18] T. Hunter II, *Advanced Microservices : A Hands-on Approach to Microservice Infrastructure and Tooling*. Berkeley, CA: Apress. (2017).
- [19] M. Sollfrank, F. Loch, S. Denteneer, and B. Vogel-Heuser, Evaluating Docker for Lightweight Virtualization of Distributed and Time-Sensitive Applications in Industrial Automation, *IEEE Transactions on Industrial Informatics*. 17 (5) (2021) 3566-3576, doi: 10.1109/TII.2020.3022843.
- [20] A. Omar, B. Imen, S. M'hammed, B. Bouziane, and B. David, Deployment of Fog Computing Platform for Cyber Physical Production System Based on Docker Technology, in *2019 International Conference on Applied Automation and Industrial Diagnostics (ICAAID)*, 25-27 Sept. 2019, 1 (2019) 1-6, doi: 10.1109/ICAAID.2019.8934949.
- [21] M. Ciavotta, M. Alge, S. Menato, D. Rovere, and P. Pedrazzoli, A Microservice-based Middleware for the Digital Factory, *Procedia Manufacturing*, 11 (2017) 931-938, doi: <https://doi.org/10.1016/j.promfg.2017.07.197>.
- [22] R. Merz, R. Hoch, and D. Drexel, A Cloud-Based Research and Learning Factory for Industrial Production, *Procedia Manufacturing*, 45 (2020) 215-221, doi: <https://doi.org/10.1016/j.promfg.2020.04.097>.
- [23] M. Ghazi Vakili, C. Demartini, M. Guerrero, and B. Montrucchio, Open Source Fog Architecture for Industrial IoT Automation Based on Industrial Protocols, in *2019 IEEE 43rd Annual Computer Software and Applications Conference (COMPSAC)*, 15-19 Jul 2019, 1 (2019) 570-578, doi: 10.1109/COMPSAC.2019.00088.
- [24] I. Bedhief, L. Foschini, P. Bellavista, M. Kassar, and T. Aguilu, Toward Self-Adaptive Software Defined Fog Networking Architecture for IIoT and Industry 4.0, in *2019 IEEE 24th International Workshop on Computer Aided Modeling and Design of Communication Links and Networks (CAMAD)*. 11-13 Sept. 2019 (2019) 1-5, doi: 10.1109/CAMAD.2019.8858499.
- [25] S. Hoque, M. S. De Brito, A. Willner, O. Keil, and T. Magedanz, Towards Container Orchestration in Fog Computing Infrastructures, in *2017 IEEE 41st Annual Computer Software and Applications Conference (COMPSAC)*, 4-8 July 2017, 2 (2017) 294-299, doi: 10.1109/COMPSAC.2017.248.
- [26] B. Madiwalar, B. Schneider, and S. Profanter, Plug and Produce for Industry 4.0 using Software-defined Networking and OPC UA, in *2019 24th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*. 10-13 Sept. 2019 (2019) 126-133, doi: 10.1109/ETFA.2019.8869525.
- [27] P. González-Nalda, I. Etxeberria-Agiriano, I. Calvo, and M. C. Otero, A modular CPS architecture design based on ROS and Docker, *International Journal on Interactive Design and Manufacturing (IJDeM)*, 11 (4) (2017) 949-955, doi: 10.1007/s12008-016-0313-8.