

---

# Configuration of a Web-Based Digital Twin Using a Modular and Flexible Simulation Chain

**Christian Plesker, Vladimir Kutscher, David Bassauer,  
and Reiner Anderl**

Department of Integrated Design, Technical University of Darmstadt Otto-Berndt-Str. 2,  
D-64287 Darmstadt, Germany

## ABSTRACT

The versatile concept of the digital twin is being applied in various economic sectors. To exploit its full potential, the digital twin has to be adaptable to changing requirements. In this paper we address the configuration of digital twins with a focus on a flexible and modular simulation chain. Thereby, we utilize the Functional Mock-Up Interface to enable variable configurations based on basic simulation components. We validate the concept by a prototypical implementation.

**Keywords:** Digital twin, Web-based, Simulation chain, Configuration, Industry 4.0

## INTRODUCTION

The digital twin marks an important milestone on the road to digitization. Various implementations already exist, often based on manufacturer-dependent, proprietary systems with specialized interfaces. This limits the end-to-end use throughout the product lifecycle. Instead, open, standardized and cross-domain technologies can support continuous and flexible usage. At the same time a digital twin cannot fulfill contrary requirements like level of detail and speed of execution. Therefore, the user should be able to adjust and modify to keep up with changing requirements. The presented paper addresses the research gap of how the digital twin can be reconfigured in the usage phase by flexibly changing the composition of individual components.

## BACKGROUND AND RELATED WORK

In the following, we present background information on the state of the art in the addressed research field and the utilized technologies, as well as relevant research work. In the context of this work, we define a digital twin as a digital representation of a physical object based on (Anderl et al., 2018). The focus is on technical products with extensive functionality like Cyber-Physical Systems whose digital twins can be planned and designed according to particular characteristics (Jones et al., 2020). (Stark & Damerau, 2020) propose an 8-dimension model to assess the desired level in each relevant dimension in order to pre-define the functionality of the digital twin. During

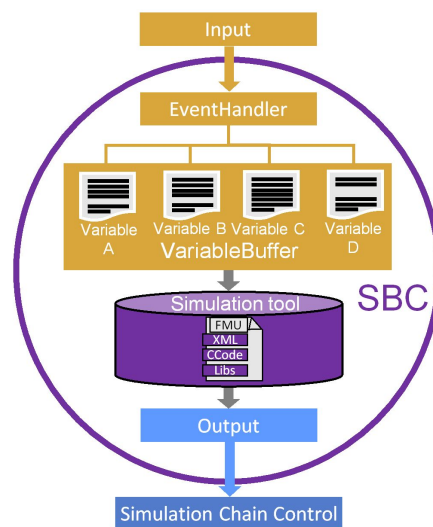
the generation of the digital twin, models from product development can be used (Stark et al., 2020). Thereby simulations form a significant part of digital twins (Boschert & Rosen, 2016). On the one hand, the simulations allow an interactive experience of the digital twin by a user (Schluse & Rossmann, 2016) and on the other hand, enable the prediction of the system behavior during operation in order to mitigate or completely avoid undesirable effects (Grieves & Vickers, 2017). Simulations from different domains converge within the digital twin and form co-simulations, which offer many new opportunities, but also pose challenges (Gomes et al., 2017). One approach to realize non-proprietary co-simulations is to use simulation models in neutral formats as basis for a digital twin. The Functional Mock-Up Interface (FMI) standard (Modelica Association Project FMI, 2019) provides a technology to accomplish this. With the FMI standard, simulations from proprietary simulation environments can be exported in the form of Functional Mock-Up Units (FMUs). A FMU serves as a container with solvers represented in C code and configurations described in an Extensible Markup Language (XML) file. On this basis, (Kutscher et al., 2021) already developed a web-based digital twin utilizing FMU simulations which served as the starting point for the present work. Schroeder et al. described a layered architecture for the management of digital twin data based on web services (Schroeder et al., 2016). Konstantinov et al. presented a concept for the development of a web-based digital twin for industrial cyber-physical systems where the authors also used the Open Platform Communications Unified Architecture (OPC UA) technology to connect the control software of the physical twin to geometry models of the digital twin. (S. Konstantinov et al., 2021). Liu et al. developed a systematic framework for modeling, configuring, and executing cyber-physical systems by using web technologies (Liu et al., 2020). The approach presented in the following goes one step further and addresses the research gap of flexible configuration of simulation elements of a web-based digital twin.

## APPROACH

In the following, we present a concept for the configuration of a digital twin by means of a modular and flexible simulation chain. The focus lies on representing the desired aspects of the physical twin by linking so-called simulation base components. The digital twin should provide a tool to support configuration without specific knowledge of the internal structure. Each component has input and output variables that are linked to other components in order to represent complex functionalities and extended behaviour. This interconnection can then be simulated in the digital twin. The prerequisite for the concept is the web-based digital twin developed in previous work (Kutscher et al., 2021), which creates a platform-independent basis. To allow the integration of different simulation types, web-based digital twin uses the FMI-standard and allows a standardized import of simulations as FMUs. An imported and stored simulation unit forms the *simulation base component* (SBC) and can be exchanged individually within the simulation chain.

### Simulation Base Component (SBC)

The SBC represents the smallest modular unit. The SBC has an FMU at its core and can be simulated by FMI-supporting tools. Depending on the use case and the configuration, these tools can run on the local system or utilize external services like web-based simulation servers, cloud technologies or databases. This enables the systematic distribution of computing power and also supports innovative business models such as a marketplace for simulations. For the interconnection within the simulation chain, the SBC has an *Input* and an *Output* (Figure 1). The Input is managed by an *EventHandler*, which reacts to incoming changes and starts the simulation. For each input variable of the simulation a buffer is required to compensate different frequencies and arrival times of input values. Input events are triggered when an input value is received. The EventHandler stores the values in the corresponding buffer and checks the requirements for the start or the next iteration of the simulation. If all conditions are fulfilled, the simulation is started. The results of each simulation iteration are then formatted and forwarded by the Output to the next SBC in the simulation chain via the *Simulation Chain Control*.

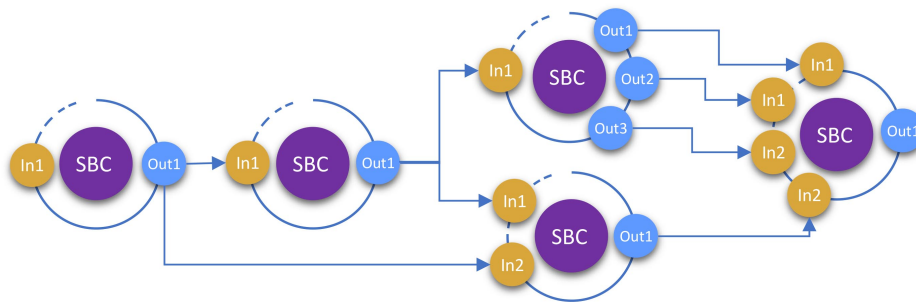


**Figure 1:** Simulation base component.

### Modular and Flexible Simulation Chain

The simulation chain consists of individual SBCs to form higher-level simulation structures. This composition enables the user to represent any existing or new functionality of the physical twin and thus to specifically configure the based digital twin. The structure is not only limited to the physical twin, but can also represent the combination of several machines into one system of systems. In this way, entire production lines and factories can be configured and simulated.

An SBC in a simulation chain is either a start-, end- or intermediate element, which together build one iteration of the simulation. Each SBC has

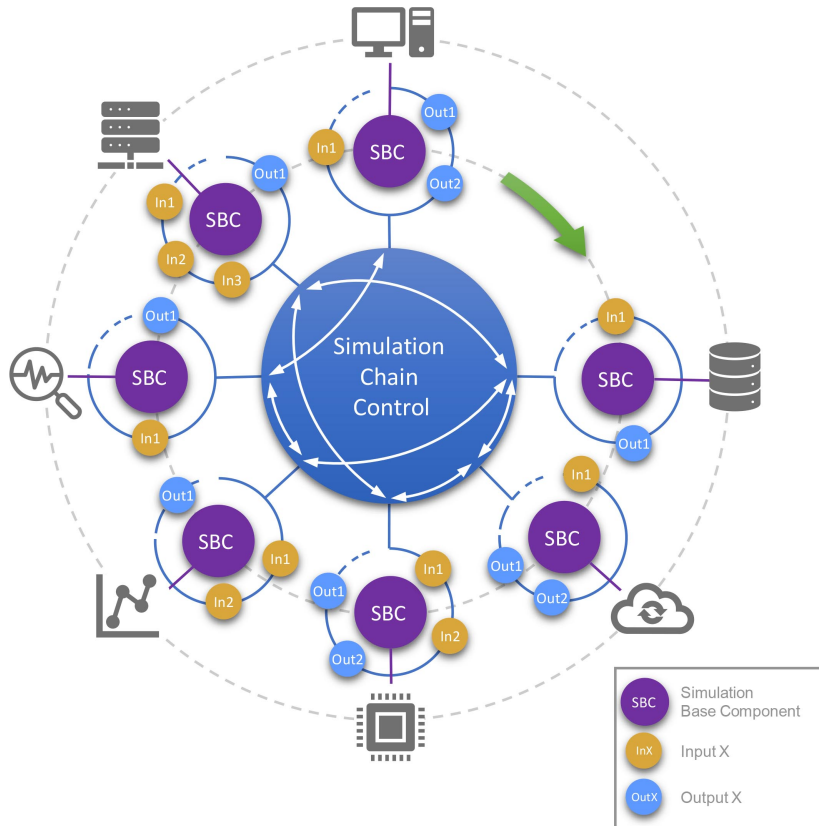


**Figure 2:** Simulation chain.

unique input and output variables (Figure 2) which depends on the FMU in its core. For linking a simulation chain, the web-based digital twin must be extended by an interface. This should have a visual component for an intuitive, no-code usage. The configuration interface should allow the placement and arrangement of the individual SBCs in any order. In the representation of the individual SBCs, the individual inputs and outputs can be shown to the user and linked individually. When linking, the user can be supported by e.g. checking that the connected out- and inputs have the same type (unit) and the same order of magnitude. In many cases, it is necessary to combine the outputs of multiple SBCs to one down-stream SBC. For this purpose, so-called branches can be created. For the logic of the simulation chain, it is important that no cycles are created by the branches, as it is a directed simulation chain between a start and end point. Every branch must complete one simulation increment before the simulation continues with the cumulative input. Due to the different simulation- and waiting-times between the individual components of the chain the intermediate results are marked with a timestamp. This indicates which time step of the simulation was simulated in order to ensure the correct timing in the individual SBC buffers. This is particularly important for simulations with input variables from different SBCs. A correctly linked simulation chain can then be simulated via the web-based digital twin. The *Simulation Chain Control (SCC)* is responsible for the correct execution of the simulation chain.

### Simulation Chain Control (SCC)

The SCC is the overall instance that ensures the instantiation of the SBCs, the correct forwarding of messages and the management of the simulation chain (Figure 3). When the simulation chain is started, the SCC is called. The structure of the simulation chain and the links of the input and output variables are required to initialize the process. In the first step, the SCC checks whether the required SBCs are currently running or connected. The SBC is a part of the digital twin, but as already mentioned, it can utilize external resources and services for simulation. After all components are ready, the simulation controller starts the first SBC simulation. The results are then forwarded as defined in the simulation chain to be used as input for the next SBCs. Through the *EventHandler* of the individual SBC, only the values belonging to the



**Figure 3:** Simulation chain control.

variable are sent to the input of the SBC. SCC thus takes over the broker function between the individual simulation chain links. Through the individual simulation of SBC, the computation of the chain can be parallelised. By interacting with the simulation chain control, all intermediate steps of the simulation chain can be recorded and saved in SCC without having to make changes to the simulation chain. For example, the intermediate values can be saved directly and made usable via the OPC UA interface of the web-based digital twin. It is now possible to configure the digital twin in such a way that it can reproduce the physical twin's behaviour and functionality.

## IMPLEMENTATION

The concept was validated by adding a *Simulation Board* to the existing implementation of the web-based digital twin of an CNC (computerized numerical control) laser. The Simulation Board is a user interface that allows the user to build up simulation chains. The new function supports the user in importing SBCs into the board and in particular in linking them individually. After successful configuration, the system can simulate advanced and new behavior profiles. The individual simulations are executed in defined increments and the partial results are forwarded within the digital twin and then visualized.

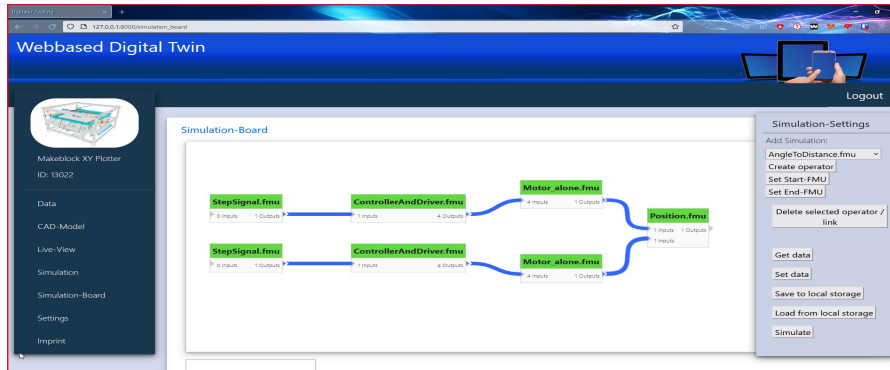


Figure 4: Simulation board.

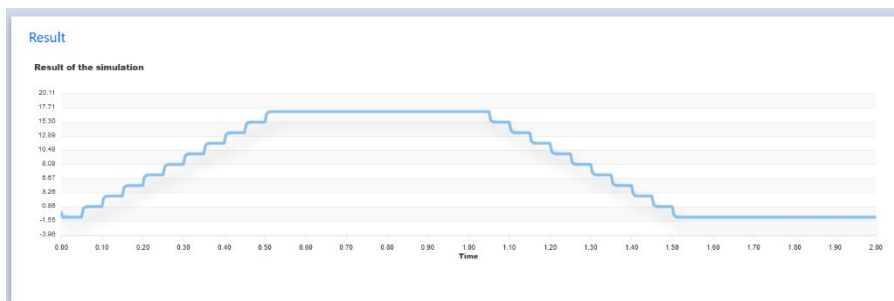


Figure 5: Possible visualization of simulation results.

A new website was created for the Simulation Board in the digital twin. The main component of the page is an HTML canvas and an associated control panel as shown in Figure 4. The Simulation Board can be controlled via the control panel on the right. All FMU-simulations that are already registered in the web-based digital twin can be selected as an SBC from a drop-down list. The “Create Operator” button inserts an operator into the canvas. The SBC is represented by the operator and has a title as well as the inputs and outputs corresponding to the simulation variables. By clicking between an output and an input, a connection visualized with a blue line can be created between the operators. After all required connections have been set, a start and an end operator must be selected. These represent the beginning and the end of the simulation chain. The simulation chain can be read out in JSON (JavaScript Object Notation) format using the *Get Data* function. In addition, the simulation chain can be saved locally. The simulation chain is started via the *Simulate* button. The functionality of the canvas board was developed with flowchart.js library (Drouyer, 2020). In order to enable the functionalities as described above, adjustments were made to the framework. The simulations of the SBCs were carried out on web servers to show the integration of external services. During execution the connection and availability of the external resources is checked. If this fails or an SBC of this type is missing, a corresponding error message is displayed. After successful simulation of the entire chain, the final result can be visualized in suitable diagrams (Figure 5).

## CONCLUSION

The research work presented makes it possible to constantly reconfigure the web-based digital twin on different levels of granularity. The configuration is intuitive and does not require in-depth knowledge of the individual modules. The concept is designed to involve external resources in the simulation process and allows parallelization. Thus, the presented concept supports the long-term use and distribution of a flexible web-based digital twin to easily adjust to new production environments. This helps to manage and implement existing as well as new behaviors in the production chain. The actual user can reconfigure digital twin at any time and use it as a basic tool for digitization of the future.

However, the evaluation also revealed constraints. We are bound to the use of the FMI standard for import and execution of simulations, which is not supported by all simulation tools. Furthermore, the decomposition of the simulation into separate elements increases the overhead of the execution, which is caused by communication, buffering and management. For this reason, a sensible tradeoff must always be made between modular decomposition and integration into fixed simulation units.

## OUTLOOK

The described concept offers additional potential, which will be realized in the future. The next step will be to investigate how a network of several simulation units can be flexibly merged into one composite to reduce the overhead when necessary. This approach can then be extended to the entire digital twin by forming a cluster of several digital twins into a higher-level system. Furthermore, complex interactions in the simulations need to be analyzed to improve reliability and efficiency. In particular, the interaction of simulations from different domains is an important aspect of the digital twin.

## ACKNOWLEDGMENT

This work was funded by the Hessian LOEWE initiative within the Software-Factory 4.0 project.

## REFERENCES

- Anderl, R., Haag, S., Schützer, K. & Zancul, E. (2018) “Digital twin technology – An approach for Industrie 4.0 vertical and horizontal lifecycle integration”, *it - Information Technology*, Vol. 60, No. 3, S. 125–132.
- Boschert, S. & Rosen, R. (2016) “Digital Twin—The Simulation Aspect”, in *Mechatronic Futures*, Springer.
- Drouyer, S. (2020) JQuery.flowchart [Online]. Available at <http://sebastien.drouyer.com/jquery.flowchart-demo/> (accessed 12.02.2022).
- Gomes, C., Thule, C., Broman, D., Larsen, P. G. & Vangheluwe, H. (2017) “Co-simulation: State of the Art”, in *Systems and Control*.
- Grieves, M. & Vickers, J. (2017) “Digital Twin: Mitigating Unpredictable, Undesirable Emergent Behavior in Complex Systems”, in *Transdisciplinary Perspectives on Complex Systems*, Springer.

- Jones, D., Snider, C., Nassehi, A., Yon, J. & Hicks, B. (2020) “Characterising the Digital Twin: A systematic literature review”, *CIRP J. of Manuf. Sci. and Tech.*.
- Kutscher, V., Olbort, J., Röhm, B., Plesker, C. & Anderl, R. (2021) “Web-Based Digital Twin“, *IIC Journal of Innovation*, S. 1–18.
- Liu, C., Jiang, P. & Jiang, W. (2020) “Web-based digital twin modeling and remote control of cyber-physical production systems”, *Robotics and Computer-Integrated Manufacturing*, Vol. 64.
- FMI - Modelica Association Project (2019) “Functional Mock-up Interface for Model Exchange and Co-Simulation: Version 2.0.1“.
- Konstantinov, S., Assad, F., Azam, W., Vera, D., Ahmad, B. & Harrison R., (2021) “Developing web-based digital twin for industrial cyber-physical systems“.
- Schluse, M. & Rossmann, J. (2016) “From simulation to experimentable digital twins: Simulation-based development and operation of complex technical systems”, *2016 IEEE-ISSE*. Edinburgh, United Kingdom.
- Schroeder, G., Steinmetz, C., Pereira, C. E., Muller, I., Garcia, N., Espindola, D. & Rodrigues, R. (2016) “Visualising the digital twin using web services and augmented reality”, *Proceedings, 2016 IEEE-INDIN*: Poitiers, France.
- Stark, R., Anderl, R., Thoben, K.-D. & Wartzack, S. (2020) “WiGeP-Positionspapier: “Digitaler Zwilling”, *ZWF* Vol. 115, S. 47–50.
- Stark, R. & Damerou, T. (2020) “Digital Twin”, in *CIRP Encyclopedia of Production Engineering*, Springer Berlin Heidelberg, S. 1–8.