Web-Based Digital Twin

Authors:

Vladimir Kutscher
Technical University of Darmstadt
kutscher@dik.tu-darmstadt.de

Johannes Olbort
Technical University of Darmstadt,
olbort@dik.tu-darmstadt.de

Benjamin Röhm
Technical University of Darmstadt
roehm@dik.tu-darmstadt.de

Christian Plesker
Technical University of Darmstadt
christian.plesker@stud.tu-darmstadt.de

Reiner Anderl
Technical University of Darmstadt
sekretariat@dik.tu-darmstadt.de

INTRODUCTION

The digital twin is gaining acceptance in a variety of industries. It is a cornerstone that enables the next step of extensive digitization. Satellites, airplanes, automobiles [1], production machines, energy systems, chemical processes [2], sports shoes [3] and humans themselves [4] receive a digital twin, opening new business models and use cases.

In many cases, the digital twin or parts of it are implemented using proprietary and specialized technologies. As a result, it is manufacturer-dependent and requires specialist knowledge to use. This hinders the interaction of different disciplines and makes it difficult to use the digital twin throughout the entire lifecycle. At the same time, open, standardized and widely used technologies also offer many opportunities for implementing a digital twin. Web technologies have been ubiquitous for a long time, connecting both people and technical systems. It is largely platform-independent, has many users, offers strong security and a high degree of maturity. Computers that have a web interface can use basic functions on the Internet immediately, usually without supplementary software. In addition, many technologies such as cloud computing, big data and Internet of Things are already available within the web technologies and are relevant for digital twins. We see this as the prerequisite for the digital twin to become established on a broad scale in practice.

In the following, we would like to present the solutions to some problems in current practice with the approach of a web-based digital twin. Thereby, we focus on industrial systems by choosing a computer-controlled laser plotter as demonstrator for a prototypical implementation. From this context, we set requirements for the concept and implementation and use them for validation. Fig. 1 shows the physical laser plotter, which is connected to its web-based digital twin bidirectionally.

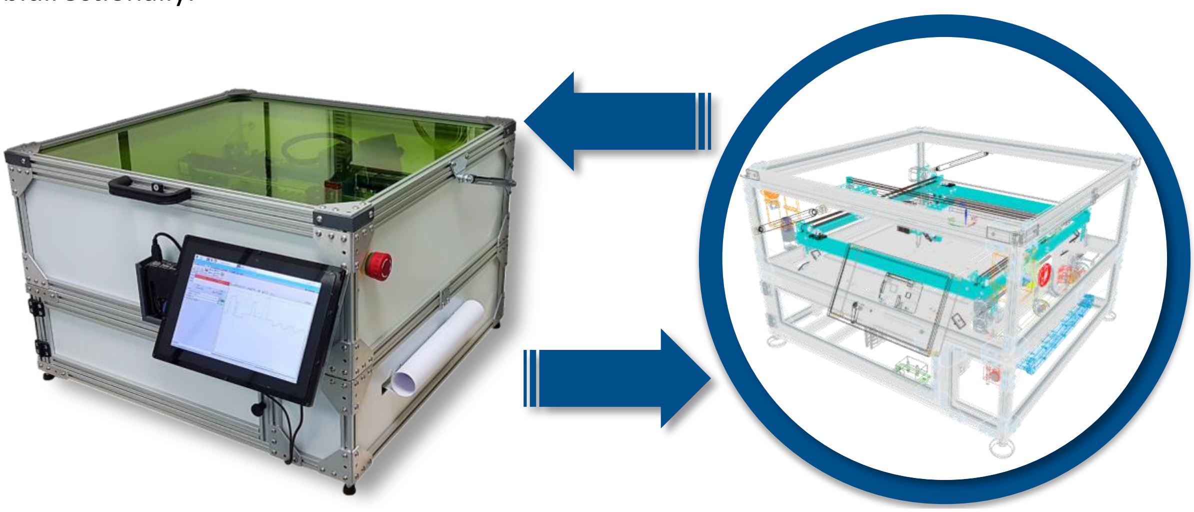


Fig. 1: Computer-controlled laser plotter as physical asset and the symbol of its web-based digital twin.

STATE OF THE ART

Digital Twin

The digital twin permeates diverse sectors and found its way into almost all areas of industry [5]. As a result, the counterpart of a digital twin is no longer just the physical twin, but rather an asset in general, including a system or a process. By means of a bidirectional connection, both are in contact and the digital twin can represent the current, future and historical state of its counterpart. Via an interface, the user can interact with the digital twin and thus influence the asset.

The *Industrial Internet Consortium* (*IIC*) is working on an overarching and concise definition of the digital twin [2] and is cooperating with the German *Plattform Industrie 4.0 [6]*, since the digital twin is a key concept for *Industrie 4.0* (*eng.: Industry 4.0*) [7]. Progressive digitization and networking can create intelligent and individual products that meet current trends and customer requirements.

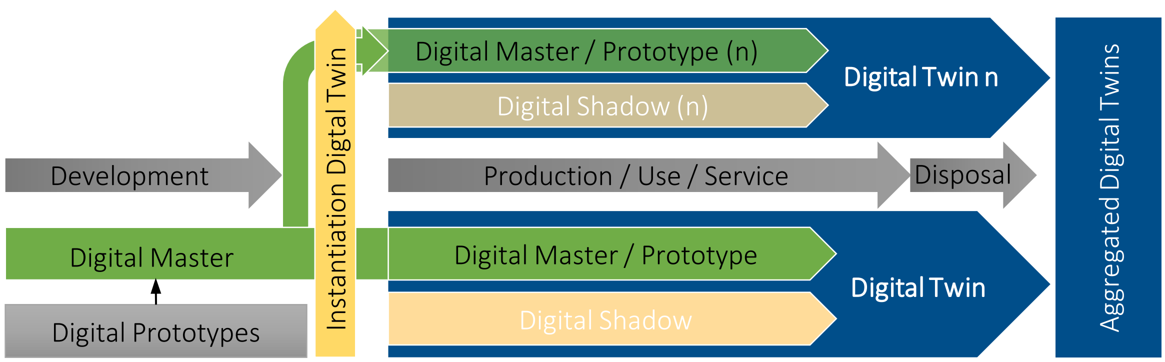


Fig. 2: Instantiation of digital twins following [9].

The German *Scientific Society for Product Development,* called *WiGeP,* provides a possible description of the life cycle of a digital twin of a product [8]. Fig. 2 shows an adapted illustration. In the presented scenario, the digital twins are instantiated from so-called digital masters, which result from product development and serve as a blueprint for the digital twin [8]. The digital master includes the digital prototypes from product development and continues to exist in the digital twin in the form of individualized models.

The entire data from the life cycle of a physical product form a so-called digital shadow. The instantiated digital twins exist throughout the product lifecycle [1] and provide aggregated data that can be reused as a knowledge pool for example to optimize the next product generation. For existing assets (brownfield), there is also the possibility of retrofitting a digital twin. The existing environment provides constraints that need to be analyzed and incorporated into the necessary retrofit measures [9]. In any case, the product-specific digital twin is linked to the asset [10].

Digital twins can be discrete or composite during their existence [2]. The discrete digital twin is an entity that generates added value on its own. In a composite digital twin, multiple elements serve a common purpose. Multiple digital twins can further act in a system of digital twins and serve an overarching purpose. In the digital twin, the internal networking of individual components and the external interaction with other twins is thus an important aspect that addressed by suitable communication technologies and interfaces.

In addition to communication, the digital twin consists of further key aspects like data, models and computing. As Fig. 3 already indicates, the aspects are not clearly separable and overlap with each other, all of which influence the digital twin. The data of a digital twin is of different nature and may cover all phases of the product lifecycle. The types of data considered in the following are operational data and simulation data, since these are relevant in the presented context.

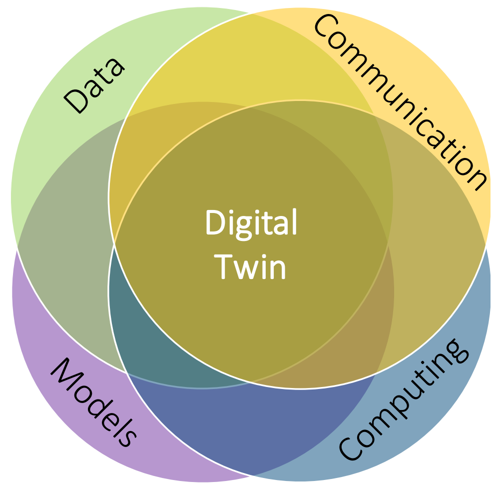


Fig. 3: Key aspects of the digital twin.

Data Modelling and OPC UA

For the operational data of an asset, the *Open Platform Communications Unified Architecture* (OPC UA) [11] attains a special position. It is a platform-independent, service-oriented architecture that is being standardized for a wide range of industries. The standard builds on two basic pillars: the transport mechanism and the data modeling. Various communication technologies are available as transports, which differ depending on the approach (client-server/publish-subscribe).

But the distinctive feature is the data modeling, which focuses on layer-wise information models. The information models have sub-models for devices in general, as well as extensibility for specific industries and vendor-specific information [12]. The industry-specific models are the result of collaborations between various companies and organizations and manifest in so-called Companion Specifications. The OPC UA information models represent the state data of the asset. All information and functions are thus available via standardized interfaces, to which the digital twin can also connect [10].

Simulation and FMI

Another key aspect in the digital twin is simulation [13] and integration various simulations into a multi-domain simulation chain. The simulation techniques and tools that exist today are mostly designed to be used during product development. On the one hand, these proprietary tools are not suitable for use in the digital twin due to frequent incompatibility with simulations from other domains and, on the other hand, due to the required licenses and expertise. The problem of compatibility is addressed in corresponding standardization organizations [14] and scientific publications [15, 16]. Vendor and domain independence can be achieved through open cross-domain modeling languages such as *Modelica* [17] and exchange formats like the *Functional Mock-Up Interface* (FMI). FMI is an open standard that defines a container and interface for exchanging dynamic models [18] and is particularly relevant in the presented context, since simulations can be supplied in the digital twin and integrated into a web-based environment. The goal is to provide a uniform interface to simplify the creation, storage, exchange, and reuse of simulations from different simulation environments. FMI can be divided into two process models: *FMI for model exchange* and *FMI for co-simulation.*

The dynamic models are packaged in a *Functional Mock-Up Unit* (*FMU*). An *Extensible Markup Language* (*XML*) file configures the FMU and C-code files describe the function. An FMU for model exchange only contains the model of the simulation. Accordingly, the tool for using an FMU must have its own solver for equation-solving. In an FMU for co-simulation, the model is compiled and exported with its own solver. This allows the tool to run the FMU independently with the solver provided and operate it without knowledge and ownership of solvers [18].

With the application of sensor data-based simulation models, the need for a dedicated *Simulation Data Management for Digital Twins* (SDM-DT) also arises. Derived from classical simulation data management systems from product development, the SDM-DT provides, manages and archives sensor-based simulation data [19]. Consequently, sensor-based simulation data can be managed in the digital twin to describe predictive and prescriptive scenarios. The performed simulations are virtually backed up and decision traceability is ensured [20].

Web Technologies

Both FMI and OPC UA already utilize web technologies such as various transfer protocols or offer corresponding interfaces. The multitude of protocols involved in the network connection between two end points can be divided into groups, which are represented, for example, in the OSI (*Open Systems Interconnection*) layer model [21]. Within individual groups, the protocols are interchangeable, resulting in many possible combinations. Widely used protocols include the *Hypertext Transfer Protocol* (*HTTP*), the *Transmission Control Protocol* (*TCP*) and the *Internet Protocol* (*IP*). Internet technologies are being further developed especially for industrial applications. Under the collective term *Industrial Internet of Things* (*IIoT*), the IIC offers concepts and architectures that make Internet technologies ready for industry and corresponds with activities within the *Plattform Industrie 4.0* [22].

A variety of technologies exists not only for communication via the Internet, but also for the development of web resources. These can be classified into the three basic areas of frontend, backend and database. While the frontend is close to the user and is processed on his computer, the backend and database is executed on the server. For the development of web applications, web frameworks specialize either on development of the backend, frontend or both (full-stack). These libraries provide pre-programmed functions and simplify the development of new applications. Some of the most popular web frameworks include *jQuery*, *React*, and *Angular* [23]. These support the developer in the creation of a web application, which for the frontend mostly contain the programming languages *Hypertext Markup Language* (*HTML*), Cascading Style Sheets (*CSS*) and *JavaScript*. HTML determines the structure, CSS the presentation and JavaScript the logic of web applications. The programming language chosen to implement the backend depends on the web application. Languages used here include C, C++, Java, Python, PHP and others.

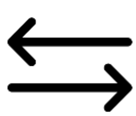
A considerable number of scientific papers already use web technologies for the conceptualization and implementation of the digital twin. Schroeder et. al. [24] employ web services in combination with augmented reality for visualizing the data of the digital twin. Souza et. al. [25] present an architecture for a digital twin based on Industrial Internet of Things technologies. Liu et. al. [26] utilize the digital twin for modeling and web-based remote control of cyber-physical production systems. In contrary to approaches that use web technologies only partially, we would like to present a completely web-based digital twin in the following.

Expected Features

What features do we expect from a web-based digital twin? In the following, we would like to list some essential characteristics we determined for our approach condensed from scientific literature [24, 27–29].

Representation of data: The digital twin must represent consistently all relevant data of the specific asset. This includes the characteristics and information about the asset, data collected by sensors and any asset-related data from the entire lifecycle from development to operation and maintenance. The data can either be located on the digital twin itself or accessed by reference to external sources.

Representation of behavior: The digital twin must be able to represent the behavior of the asset. The behavior of an asset can usually be represented by mathematical equations that describe fundamental physical, chemical, and biological phenomena. It can be analytical models that run on a computer. Or it may be software in general, which describes the logic of a system and dictates it by electronics. All these models and computation determine the changes of an asset over time and thus its dynamic behavior.



Controlled bidirectional connection: The digital twin must be able to establish a bidirectional connection with the asset. This connection does not necessarily have to exist all the time, as tests may be performed on the digital twin without directly influencing the asset, for example. For this reason, we do not consider the bidirectional connection to be present all the time but controlled, to also enable independence of the systems. This is also beneficial during failure of a system.

Standardized external interfaces: Digital twins must be able to provide their data and functions to each other and to other entities. To enable the exchange, a network connection must be provided and communication must be based on vendor-independent interface standards.

Standardized internal interfaces: Internal communication within a composite digital twin must also be based on standardized interfaces. Since the individual elements come from different domains, their need a unifying interface.

Quality of services: The digital twin does not have any benefit if it cannot provide sufficient quality of service. Especially in the industrial environment, the reliability of the services, for example in terms of real-time requirements, stability under load, or accuracy, are crucial factors that the digital twin must guarantee.

WEB-BASED DIGITAL TWIN

In the following, we want to discuss the concept of a web-based digital twin. For the generation of the web-based digital twin we used a methodology with four steps: requirements analysis, system design, implementation, and verification. The requirements analysis is used to determine the requirements on a digital twin in general and a web-based digital twin in particular. We then used these requirements in the system design to determine suitable technologies for implementing a web-based digital twin. In the third step, the prototypical implementation serves to test the viability of the final model. The verification of the requirements in the last step verifies the suitability of the generated web-based digital twin.

Concept of a Web-Based Digital Twin

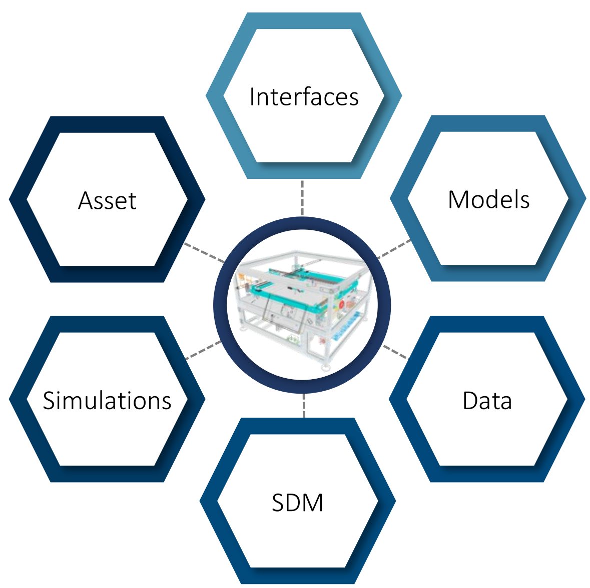


Fig. 4: Components of a web-based digital twin.

Using the described requirements and the methodological approach, we have developed a distributed web-based digital twin. Fig. 4 represents the components relevant during our design. The web-based digital twin represents an asset in its current and past state throughout its lifecycle. The state is recorded in the form of data and stored in databases. To obtain an estimate of a possible state in the future, the digital twin also represents behavior using models and simulations. Since on the one hand the simulations should run mainly autonomously in the background and on the other hand several simulation types interact with each other to represent the entire system, a simulation data management (SDM) is needed. The digital twin makes its data and functions available to external entities via interfaces and accesses their resources. These entities include, for example, other digital twins, assets, and humans.

Fig. 5 shows the structure of the digital twin. In the following, we would like to discuss the individual aspects. As a web-based application, the digital twin must ensure access to resources via HTTP-interface. All information and functions provided must be able to be called, changed, saved or controlled in this way. The web server makes it possible to provide resources, for example, by means of web pages via HTTP.

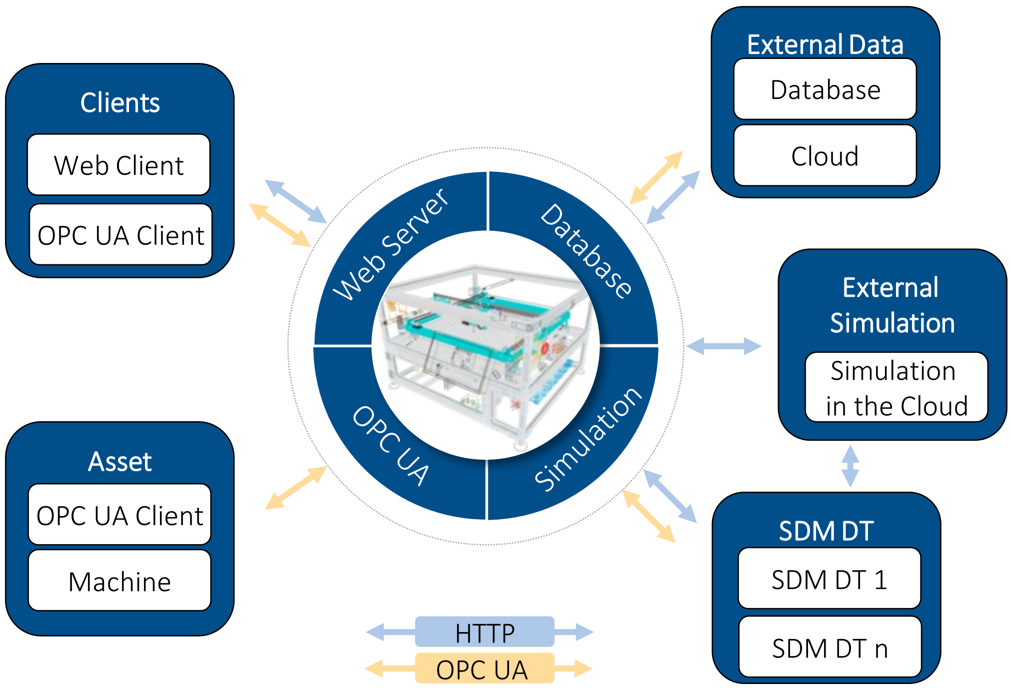


Fig. 5: Web-based digital twin.

The web server stores the data of the web server itself, as well as of the digital twin, in databases. This can be, for example, user login data, data from product development or the operating data of the asset. If the size of the data load exceeds the available memory, additional external databases are used. The web server has access to the asset's data and presents the information by means of web pages. Simple texts, images, graphics or videos can be integrated directly.

The visualization of special file formats such as CAD (*Computer Aided Design*) models is enabled by libraries of the JavaScript scripting language, which although does not support all file formats like most native ones. Neutral or standardized file formats, such as *STEP* (*STandard for the Exchange of Product model data*), *IGES* (*Initial Graphics Exchange Specification*), *STL* (*Standard Triangulation Language*) or *VRML* (*Virtual Reality Modeling Language*) differ in their information content. While STEP transfer most of the information from a native format, STL as a 3D format describes only the surface. With the possibility to visualize the data of the linked asset, the web-based digital twin can now digitally represent the state.

The bidirectional connection between the digital twin and the asset is established on the basis of OPC UA information models, which are integrated in an OPC UA server [10]. Due to the vertical integration of OPC UA and the renunciation of the classic automation pyramid, the OPC UA server can be located anywhere. The OPC UA server contains the status information and provides them to any OPC UA clients. In addition to the simple display of the information units, the state of the asset can be visualized with the help of geometric models.

Another characteristic of a digital twin is the representation of behavior. Representing behavior digitally means replicating how it works through simulations. Performing simulations within the web-based digital twin can be divided into three categories:

- Simulations on the client side
- Simulations on the server side
- Simulations on external servers

The three categories differ in the use of computing resources to perform the simulations. Simulations on the client side use the resources of the requesting client for this purpose. To run the simulations, the desired solution algorithms must be developed in JavaScript. The shift to the client side has advantages if the client has large computing resources and drawbacks if the client does not, e.g. a smartphone. The opposite approach is simulation on the server side. The simulation accesses the same resources as the web server of the web-based digital twin for computation. This also reverses the pros and cons compared to frontend simulation. A third option is to move the computation to external hardware. In this case, the simulation runs on an external server and the results return to the web server. The external server can be self-managed or provided by cloud computing solutions.

The integration of simulations into the web-based digital twin requires the selection of a suitable interface. The multitude of possible interfaces from different simulation environments is a challenge that is addressed by the FMI standard. Currently, more than 100 tools are available for exporting and importing FMUs. With the increasing number of FMUs in the digital twin, the management of the simulation models becomes more important and should be addressed by SDM-DT.

From the corresponding metadata of the models, the respective open simulation parameters with their functional values as well as the output parameters can be read out. The different available simulation models can put together into simulation networks, knowing the input- and output parameters. The use of simulation networks enables time-efficient high-resolution simulations. Crucial for the calculation of the simulations, is the linking of the open simulation parameters with the corresponding sensor values of the asset. Furthermore, discrete simulation data must be generated from the continuous sensor values, which are calculated together with the models.

Depending on the sampling rate of the sensors, different measured values from different sensors and different time steps are combined to a coherent data set. The fusion with data that may not be captured by the asset, such as environmental values, must also be ensured. The use of a simulation data management system in the digital twin meets these requirements. Only through the central interface between models, sensor data and solver, an efficient sensor data-based simulation can be realized.

The current FMI technology is suitable for simulation on the server side and on external servers. Client-side simulation cannot be performed with the current state of the art, since no tools for integrating FMI technology are available for JavaScript.

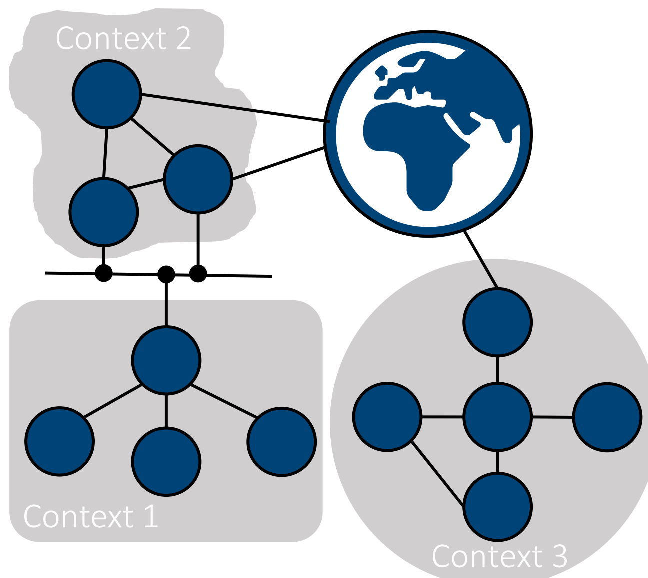


Fig. 6: Possibilities of networking digital twins.

For internal and external communication, the developed web-based digital twin basically offers two interfaces. As shown in Fig. 5, the digital twin uses both HTTP and OPC UA. For the bidirectional connection between the digital twin and the asset, the web-based digital twin utilizes an OPC UA server, which enables a uniform, cross-industry interface. All accessible information and functions are available via this server and can be requested by a client.

Networking between digital twins, which can have different relationships to each other, also plays a key role. Fig. 6 shows some possibilities for the networking. First, the digital twins are arranged in their environments or contexts. In the hierarchical structure of context 1, a digital twin connects across the boundaries of the context by means of an internal network to several digital twins of a second context. The fully networked digital twins of the second context are partially connected to the Web and can interact with a third context with its own structure. This flexible networking is fundamental for a digital twin and can be implemented in the web-based digital twin by means of the web technologies.

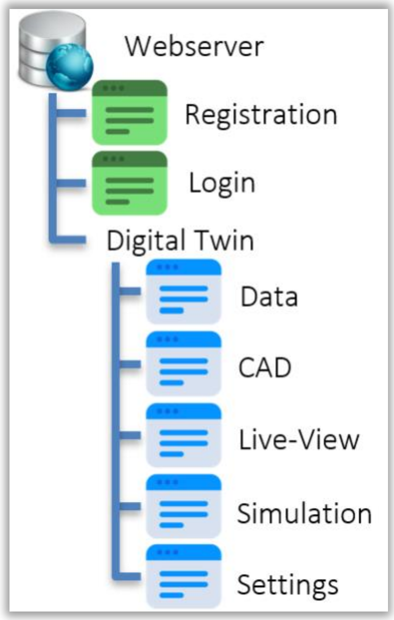


Fig. 7: Structure of the web-based digital twin.

PROTOTYPE OF A WEB-BASED DIGITAL TWIN

We implemented a prototype to validate the concept. For the technical implementation, we used the Python programming language and numerous libraries.

The web server as the core of a web-based digital twin was developed using the open-source full-stack framework *Django*. The individual functional areas of the web-based digital twin are implemented as apps in Django and thus enable the modularization of the digital twin. Fig. 7 shows the structure of the web-based digital twin. In addition to the public pages Login and Registration, six additional pages are added for the digital twin. The Data page enables access to asset data coming from multiple origins like databases and cloud services. The CAD view contains 3D models of the asset. Live View displays the current state of the asset. The Simulation tab provides access to simulations, their execution and their results. The Settings view provides the preferences of the web-based digital twin.

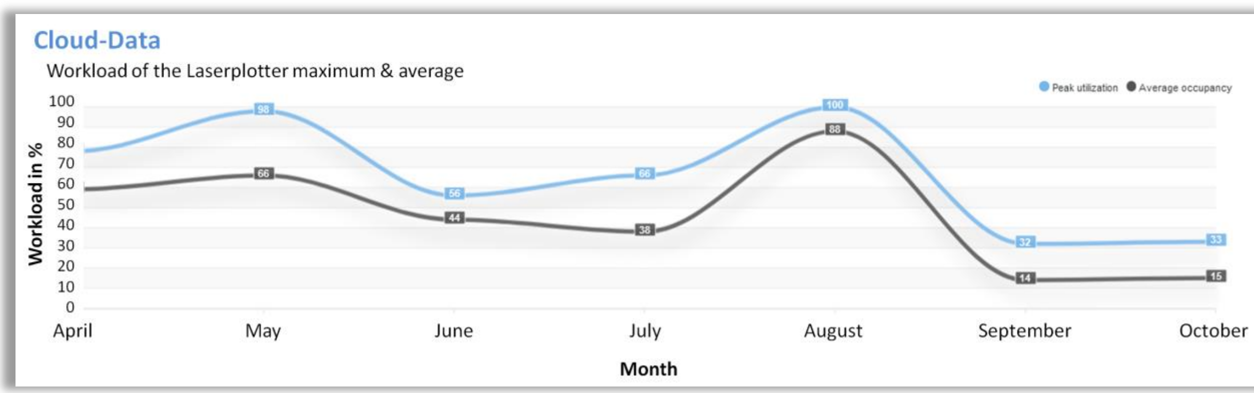


Fig. 8: Exemplary operation history of the laser plotter using ApexCharts library (https://apexcharts.com/).

The data for the Data View originates from different sources. On the one hand the digital twin has its own database system, which can be one of the databases the utilized framework Django supports, for example: PostgreSQL, MariaDB, MySQL, Oracle or SQLite. On the other hand, external databases, also from the cloud, can be integrated via HTTP. Fig. 8 shows exemplary data on the operation history of the laser plotter.

Furthermore, the Data View links the OPC UA information model of the laser plotter to the web service. Fig. 9 shows a section of the data node structure created using the *FreeOpcUa*[1] library. The JavaScript library *jsTree*[2] was used to display the data [10] of the OPC UA Server. In addition to the system information structured according to use cases, the control variables of the laser plotter are also represented.

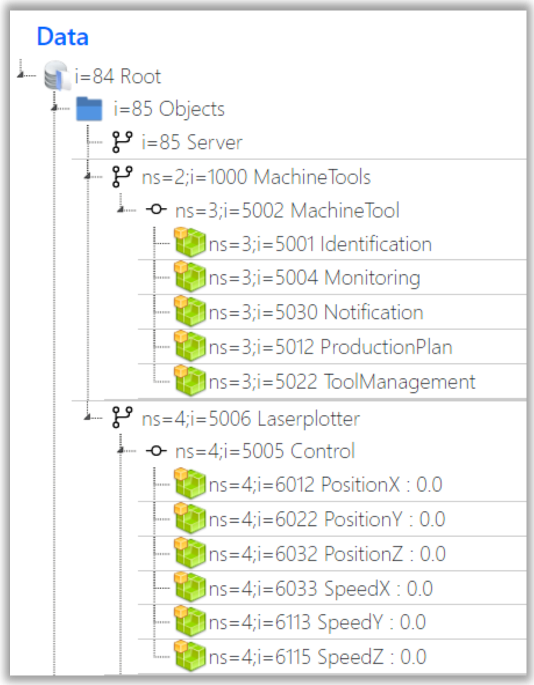


Fig. 9: OPC UA data structure of the laser plotter integrated into the web-based digital twin (ns – namespace, i – Identifier).

For the basic representation of geometric models there are some JavaScript libraries. However, the import of standardized CAD exchange formats such as STEP are not supported as the preferred option. Babylon.js was identified as an alternative which supports the file formats .glTF, .glb, .obj and .stl. The CAD models are represented by vector graphics on the web pages and can be viewed in the Internet browser from different perspectives. The calculations are performed on the computing resources of the client and thus determine the performance. Fig. 10 shows an example of the implemented CAD user interface using the plot and laser unit.

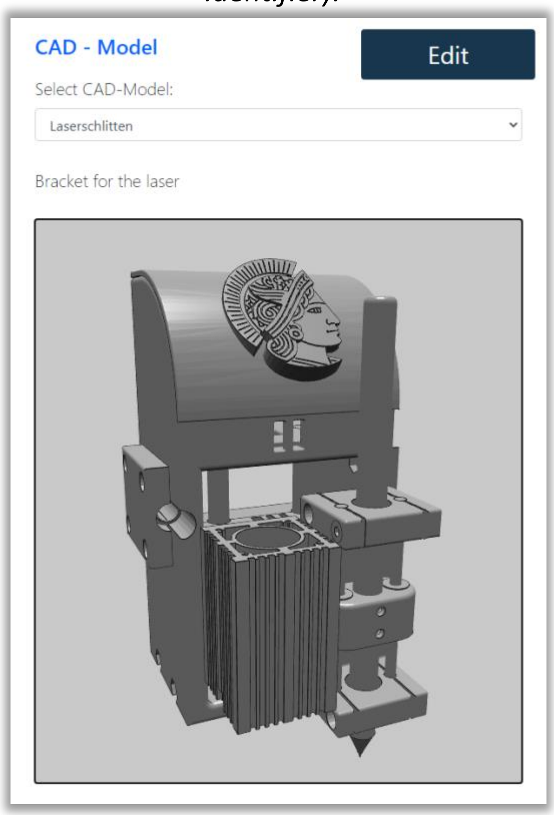


Fig. 10: Visualization of 3D geometry of a laser plotter component.

The live view combines the up-to-date data and the 3D geometry models. The representation of the

[1] https://github.com/FreeOpcUa/freeopcua

[2] https://www.jstree.com/

models is based on the continuously updated variables from the OPC UA server within the bidirectional connection to the physical asset and thus reflects the current, spatial state of the system. Fig. 11 shows the geometry model of the laser plotter mechanics, whose position of the cartesian axes is animated according to the position variables from the OPC UA server.

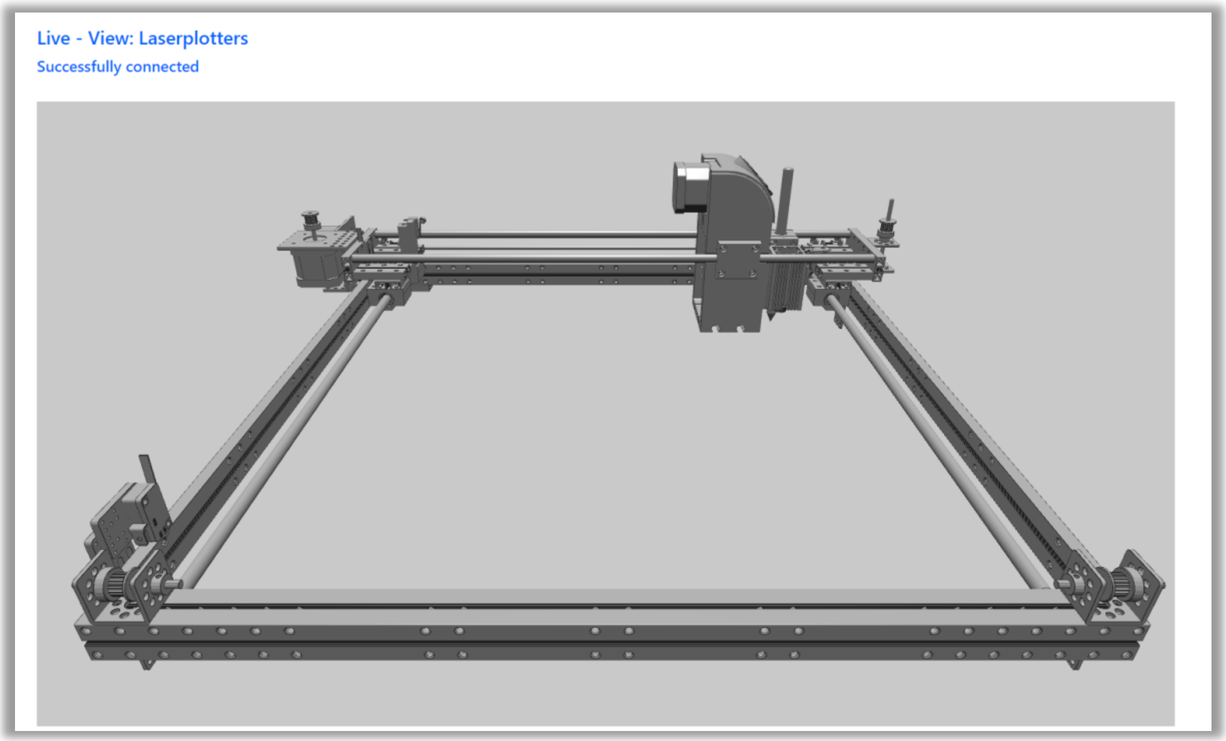


Fig. 11: Live view of the up-to-date state of the asset. The animation is moving accordingly to the current position variables in the OPC UA server. The model is stripped down to the essentials to speed up the animation.

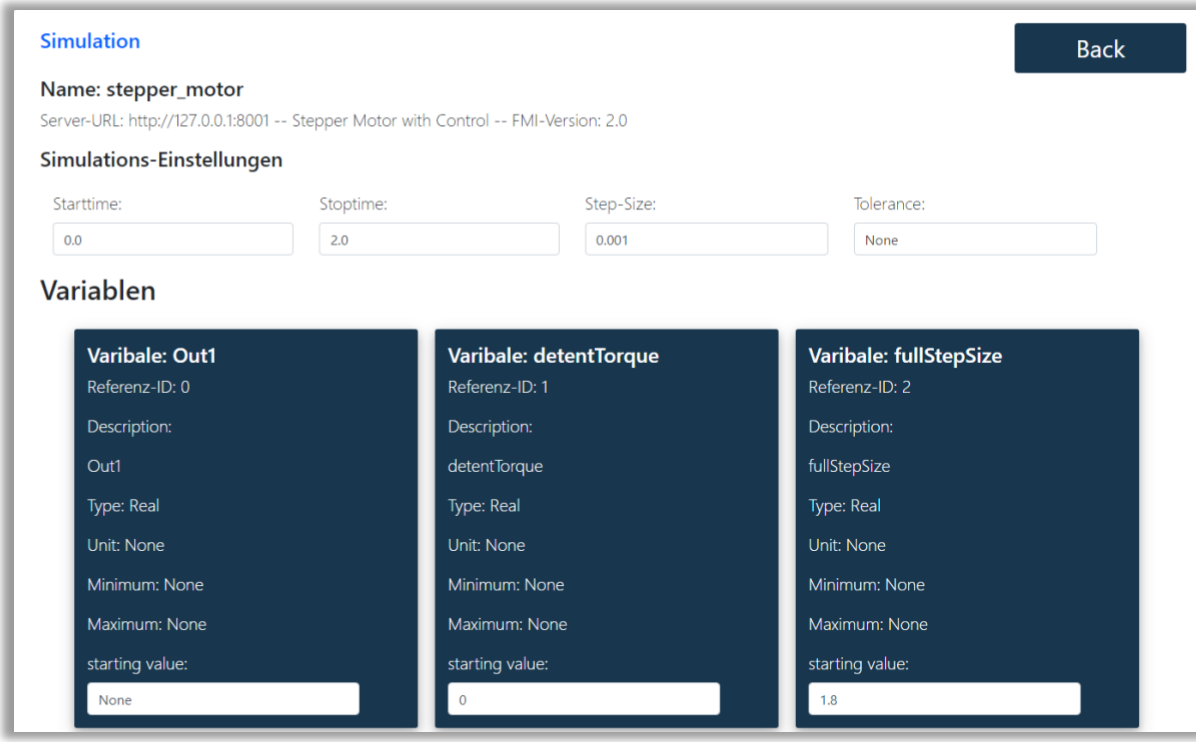


Fig. 12: Web-based user interface for the FMU simulation of a stepper motor.

The web-based digital twin can also include different types of simulations using FMI. For the laser plotter, for example, the behavior of the stepper motors used is relevant. The modelling and simulation tool MATLAB/Simulink was used to create a suitable simulation. Subsequently, the simulation was exported as a stand-alone FMU with solver. Variables were defined, which serve the user as input parameters for the simulation. The simulation runs on an external server and is integrated into the web-based digital twin using HTTP. After successfully adding a simulation, the web-based digital twin creates a simulation interface automatically, as shown in Fig. 12. An input field displays the default start value, which can be adjusted by the user. By pressing the start button the digital twin sends the corresponding data to a simulation server via an HTTP request to call the corresponding simulation function. Fig. 13 shows the simulated steps of the stepper motor over time. The simulations can thus be used, for example, to analyze the behavior of individual components under specific conditions.

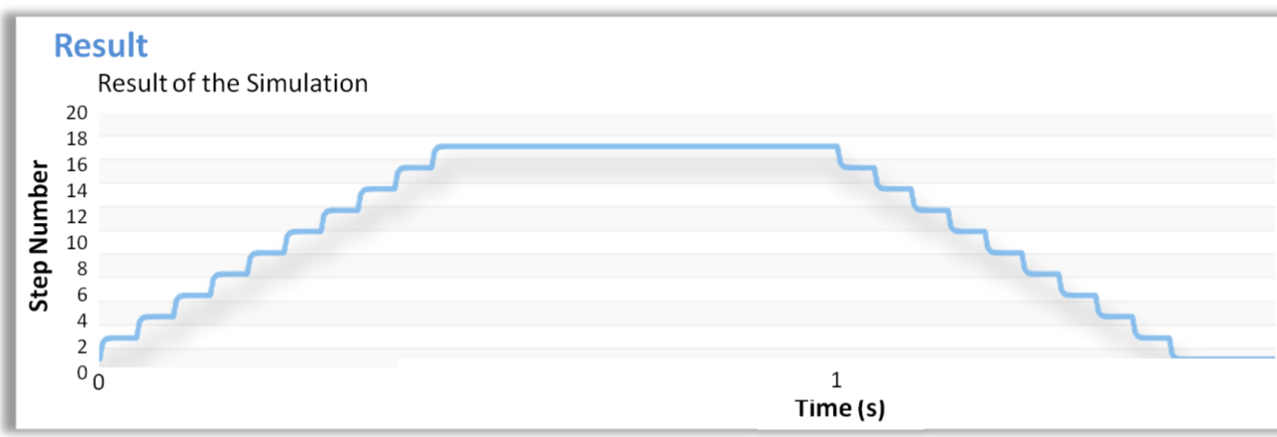


Fig. 13: Simulation results of an FMU stepper motor simulation on an external simulation server.

Discussion of the Results

In the following, we would like to discuss the presented concept and the implementation regarding the requirements for a web-based digital twin and the challenges and research gaps. Table 1 lists our assessment of the requirements.

Table 1: Assessment of the elaborated requirements on a web-based digital twin.

Requirements	
Representation of data	◐
Representation of behavior	◐
Controlled bidirectional connection	●
Standardized external interfaces	●
Standardized internal interfaces	●
Quality of Services	◐

As we have shown in the concept and implementation, different types of data can be represented on the web to describe the asset. However, the integration of CAD models is limited at the current time. Exchange formats such as STL can transfer the geometry - but information about the material or information about product manufacturing is lost. Extensive formats such as STEP cannot be imported with the libraries examined.

After the system is represented in the web-based digital twin, the current status and behavior can be mapped by coupling the status information. The FMI standard can be used to bridge the incompatibilities between different simulation types when interconnecting them to build a simulation chain. The support of the FMI import and export is different in the proprietary simulation tools and not consistently possible. In addition, the interfaces between simulations of different domains are not standardized and the communicated data must be adapted manually. There is still a great need for research in the use of simulation data management in connection with digital twins. The conceptually elaborated idea could not be tested in the prototype, since no supporting development tools could be identified in this area.

The web-based digital twin, which is similar in state and behavior, can be coupled bidirectionally with the asset. However, due to the conceptual consideration, the directional connection was not implemented rigidly, but can also be unidirectional or completely disconnected depending on the use case.

By using standardized web interfaces when providing data and functions, the web-based digital twin can interact with other entities. The web server provides all functions via HTTP using all HTTP standards.

The digital twin must offer sufficient service quality depending on the use case in order to be used reliably. The quality of service depends on many factors, which must be analyzed depending on the implementation. Despite the advanced technologies, it will not be possible to ensure the guaranteed and very short response time required in some industrial applications. The communication between the individual components of the digital twin, as well as the processing of individual programs, and the execution of simulations reaches a composition of the digital twin that cannot be used for real-time applications at the current state. Further research needs to be done in this area and the technologies used need to be made more capable for this type of application.

Overall, we believe that the web-based digital twin offers an interesting opportunity to implement new use cases and business models. In this context, some suitable web technologies are already available, while others need research and development first.

References

[1] M. Grieves and J. Vickers, “Digital Twin: Mitigating Unpredictable, Undesirable Emergent Behavior in Complex Systems” in F.-J. Kahlen et al., Eds.: Springer International Publishing, 2017.

[2] Industrial Internet Consortium, “Digital Twins for Industrial Applications: Definitions, Business Values, Design Aspects, Standards and Use Cases”, IIC, Feb. 2020.

[3] U. Weber and H. Grosser, “Digital Twins - Leading the Way to Tomorrow’s Ecosystems: Detecon Study”, Detecon, 2019.

[4] B. R. Barricelli, E. Casiraghi, J. Gliozzo, A. Petrini, and S. Valtolina, “Human Digital Twin for Fitness Management”, *IEEE Access*, 2020.

[5] W. Kritzinger, M. Karner, G. Traar, J. Henjes, and W. Sihn, “Digital Twin in manufacturing: A categorical literature review and classification”, 2018.

[6] P. I. 4.0. IIC, “Digital Twin and Asset Administration Shell Concepts and Application in the Industrial Internet and Industrie 4.0: An Industrial Internet Consortium and Plattform Industrie 4.0 Joint Whitepaper”, 2020.

[7] R. Anderl, S. Haag, K. Schützer, and E. Zancul, “Digital twin technology – An approach for Industrie 4.0 vertical and horizontal lifecycle integration”, *Information Technology*, 2018.

[8] R. Stark, R. Anderl, K.-D. Thoben, and S. Wartzack, “WiGeP-Positionspapier: Digitaler Zwilling”, *ZWF*, 2020.

[9] V. Kutscher, J. Olbort, O. Anokhin, L. Bambach, and R. Anderl, “Upgrading of legacy systems to cyber-physical systems”, *Proceedings of TMCE 2020*, 2020.

[10] V. Kutscher, J. Olbort, C. Steinhauer, and R. Anderl, “Model-Based Interconnection of Digital and Physical Twins Using OPC UA” in *Advances in Intelligent Systems and Computing*, Springer International Publishing, 2020.

[11] DIN, “DIN 62541-1 OPC UA: Teil 1: Übersicht und Konzepte” 2011.

[12] W. Mahnke, S.-H. Leitner, and M. Damm, *OPC Unified Architecture,* Springer Vieweg, 2009.

[13] S. Boschert and R. Rosen, "Digital Twin—The Simulation Aspect", in P. Hehenberger and D. Bradley, Eds.: Springer International Publishing, 2016.

[14] Simulation Interoperability Standards Organization. *Official Website* [Online]. Available: https://www.sisostds.org/ (accessed: Feb. 27 2020).

[15] C. Gomes, C. Thule, D. Broman, P. G. Larsen, and H. Vangheluwe, "Co-simulation: State of the Art" in *Systems and Control*, 2017.

[16] D. Broman, L. Greenberg, E. A. Lee, M. Masin, S. Tripakis, and M. Wetter, "Requirements for hybrid cosimulation standards" in *Proceedings of the 18th International Conference on Hybrid Systems Computation and Control*, 2015.

[17] Modelica Association, "Modelica - A Unified Object-Oriented Language for Systems Modeling: Language Specification", 2017.

[18] Modelica Association Project FMI, "Functional Mock-up Interface for Model Exchange and Co-Simulation: Version 2.0.1" 2019.

[19] Arbeitskreis PLM, "Simulation Data Management: Integration of Simulation and Computation in a PDMEnvironment (SimPDM)" 2008.

[20] F. Rieg and R. Steinhilper, *Handbuch Konstruktion*: Hanser, 2012.

[21] *Information technology - Open Systems Interconnection: Basic Reference*, 7498-1, EN ISO/IEC, 1995.

[22] P. I. 4.0. IIC, "Architecture Alignment and Interoperability: An Industrial Internet Consortium and Plattform Industrie 4.0 Joint Whitepaper" 2017.

[23] Stack Overflow, *Developer Survey 2020.* [Online]. Available: https://insights.stackoverflow.com/survey/2020 (accessed: Jan. 25 2021).

[24] G. Schroeder *et al.,* "Visualising the digital twin using web services and augmented reality" in *Proceedings of the International Conference on Industrial Informatics (INDIN)*, 2016.

[25] V. Souza, R. Cruz, W. Silva, S. Lins, and V. Lucena, "A Digital Twin Architecture Based on the Industrial Internet of Things Technologies" in *2019 IEEE International Conference on Consumer Electronics (ICCE)*, 2019.

[26] C. Liu, P. Jiang, and W. Jiang, "Web-based digital twin modeling and remote control of cyber-physical production systems" *Robotics and Computer-Integrated Manufacturing*, 2020.

[27] L. F. C. S. Durão, S. Haag, R. Anderl, K. Schützer, and E. Zancul, "Digital Twin Requirements in the Context of Industry 4.0" *Proceedings of the 15th IFIP International Conference on Product Lifecycle Management*, 2018.

[28] B. Ashtari Talkhestani *et al.,* “An architecture of an Intelligent Digital Twin in a Cyber-Physical Production System” *at - Automatisierungstechnik*, 2019.

[29] D. Jones, C. Snider, A. Nassehi, J. Yon, and B. Hicks, “Characterising the Digital Twin: A systematic literature review” *CIRP Journal of Manufacturing Science and Technology*, 2020.

ACKNOWLEDGEMENTS

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

➢ Return to IIC Journal of Innovation landing page for more articles and past editions