

The Potential of FMI for the Development of Digital Twins for Large Modular Multi-Domain Systems

Marcus Wiens¹ Tobias Meyer¹ Philipp Thomas¹

¹System Technology, Fraunhofer IWES, Germany,
{marcus.wiens,tobias.meyer,philipp.thomas}
@iwes.fraunhofer.de

Abstract

Digital twins enable the observation, prediction, and optimization of a physical system and thus allow to realize their full potential. However, their functionality is mainly based on simulation models of the entire system behavior. For modular multi-domain systems, this requires the extensive use of dynamically composed models that are made up of individual component models. The FMI-Standard forms a solid foundation for this problem and is very well known in the automotive engineering fields. However, composed system models using FMI are not widely adapted in renewable energy and wind energy yet. So far, the coupling of simulation models is limited. This paper discusses the strategy of building digital twins from individual FMUs with predefined model interfaces based on an ontology for renewable energy systems. An accelerated development is enabled by the exchange of sub-models in the digital twin without adjustments of interface. An example for the proposed process is given by the composed simulation model of a hydrogen generation process based on wind energy.

Keywords: Digital Twin, Functional Mockup Unit, Wind Energy, Renewable Energy, Ontology

1 Digital Twins in Wind Energy

Digital Twins are a major contribution for the digitalization of physical systems and processes. The digitalization of systems is a key part of Industry 4.0, which focuses on the development of cyber physical systems. As a result, a digital twin enables use-cases e.g., continuous observation, prediction, optimization of logistics etc., to maximize the potential of physical systems (Tao et al.2019). There are multiple definitions of digital twins, like the often-cited definition from NASA: *“The Digital Twin is an integrated multiphysics, multiscale, probabilistic simulation of an as-built vehicle or system that uses the best available physical models, sensor updates, fleet history, etc., to mirror the life of its corresponding flying twin.”* (Glaessgen and Stargel. 2012) and many more (Fuller et al.2020). Common to them is that there is a digital entity representing a physical entity and the adaptation of both objects based on an

exchange of data. To avoid misconception of digital twins, they are categorized (Kritzinger et al.2018) into distinct definitions by their level of integration:

- **Digital Model:** A digital copy of an existing or planned physical entity. There is no automated exchange of data between the digital copy and the physical entity. The data of the digital copy might be used in the development process of the physical entity, but a change in the digital copy has no immediate effect on the physical entity.
- **Digital Shadow:** A digital shadow extends the digital copy by an automated one-way exchange of data. The physical entity changes the state of the digital shadow, but not in the other way.
- **Digital Twin:** The digital twin has full level of integration, which is realized by the automated exchange of data between the physical entity and the digital copy. Both systems affect each other.

Those categories build on top of each other and can be interpreted as development steps of a digital twin. In fact, the development of a digital twin not only consist of the creation of the digital entity, but also of the surrounding elements for data exchange and interfaces for services. The Generic Digital Twin Architecture (Steindl et al., 2020) sets a well-defined structure for all elements of a digital twin. Furthermore, the use of Ontologies is demonstrated to organize the services and internal functionality of the digital twin. Ontologies give structure to information and make them processable by a machine (Gruber. 2016).

In wind energy, digital twins are primarily developed for components or very large systems. Among large-scale projects, Ramboll's "True digital Twin" (Tygesen et al. 2018), which represents offshore structures, is particularly noteworthy. At the component level, the drive train is represented in detail, for example by Winergy (Flender International GmbH) or Schaeffler (Schaeffler AG). However, they are either designed specifically for the capabilities of a specific product or are limited to the suppliers' own components. The development of holistic

full-turbine digital twins is currently at the stage of building the virtual entities, without the required surroundings. The wind turbine model adjustment for an existing real wind turbine (Pimenta et al.2020) shows potential for a virtual entity, but the process requires many steps. Another method (Branlard et al.2020) presents the use of linearized models in combination with a Kalman Filter to estimate real-time load and fatigue. The applications of the presented digital twins lie in the tracking of fatigue damage or evaluation of alternative operation strategies. However, those systems implement single component (one wind turbine model) digital twins, which are not designed for further coupling of simulation models. We assume that a coupled model will be necessary for the development of digital twins for large systems like an entire wind farm or green hydrogen production facilities.

Furthermore, current challenges in research of digital twins are stated in a review paper (Fuller et al.2020) for the fields of manufacturing, healthcare and smart cities, but their findings seem to be more general according to other papers (Tao et al.2019):

- Unified development framework
- Lack of clear definitions
- Standardization
- Lack of large-scale projects due to missing domain knowledge

Hence, the development of digital twins is in an early stage. This article addresses the above issues by proposing a development process for the digital model of the physical entity based on Functional Mock-up Interface (Modelica Association Project FMI) (FMI) and Co-Simulation. A Co-Simulation is the technique of the simulation of a coupled system based on the composition of single simulators (Gomes et al.). Multiple abstraction levels (van Nguyen et al.2017) have to be considered.

2 Digital Twins for Large Modular Multi-Domain Systems

Large multi-domain systems are composed modularly of individual components. Each component can be changed individually, e.g., by means of a reconfiguration of its behavior, by a change in physical properties or be a full exchange of the entire component. The entire system can be reconfigured by removing or adding components. All these aspects need to be reflected in the simulation models. The large number of possible configurations makes it impossible to cover them all with a dedicated full-system simulation model. Instead, a similar modular approach for the system model is required. Additional complexity arises, since systems with a large scope usually cover multiple domains of different engineering

fields. This usually requires the combination of models from different modelling tools.

The modular setup of a full-system model from individual component models is a prime candidate for the assembly of a full-system model from individual Functional Mock-up Units (FMUs). FMUs allow simulation models from various modeling tools to be coupled. The simulation models are contained as a binary library file and the underlying algorithms remain hidden. Two types of FMU are specified in the FMI standard: co-simulation and model exchange. Only co-simulation FMUs are used here. In addition to the simulation model, co-simulation FMUs also contain the simulation solver. As a result, the FMU independently computes simulation results for each simulation time step and does not require a higher-level solver. In this way, models with different integrator time steps can be coupled, for example the mechanical system of the wind turbine (time step: 10 ms) and the electrical system (1 ms). This makes FMUs an ideal base for digital twins of large multi-domain systems.

The process is most useful if a model database in combination with a dedicated full-system model assembly tool is used. This requires predefined interfaces and multi-dimensional connectors between single models for an efficient modelling. Predefined interfaces are needed, since large scale systems of multiple domains have a various number of elements and interconnections and therefore can make the combination of models complicated between domains. An ontology is one possible solution to this problem. In fact, the ontology can manage the knowledge about model interfaces and thereby be an instruction for in- and outputs for specific objects. This implies the implementation of specific adapters based on the ontology in the utilized modelling tools. Additionally, the ontology should give unique names for connections of the same kind. This eases the (automated) connection later. An example for an ontology is given in Section 3. Already existing models (e.g., engineering models), which are considered for the model database, can be connected with those adapters in their specific modelling tools. In this way, there is no coupling between the model in- and output and the required connections from the ontology. In the end the models are exported as FMU to the database.

The usage of an ontology for the definition of abstract interfaces yields the additional advantage that FMUs, which model components in different manners, can be exchanged without additional changes to the full-system model. In our digital twin approach, it is possible to exchange a detailed simulation model to a coarser simulation model or vice versa since the model interface remains unchanged. This allows for a high degree of flexibility when adapting to the requirements of different usage scenarios. Furthermore, the model database benefits from predefined “real world” interfaces to connect the digital models with the physical entity. This is represented by an FMU model which maps measurement data streams

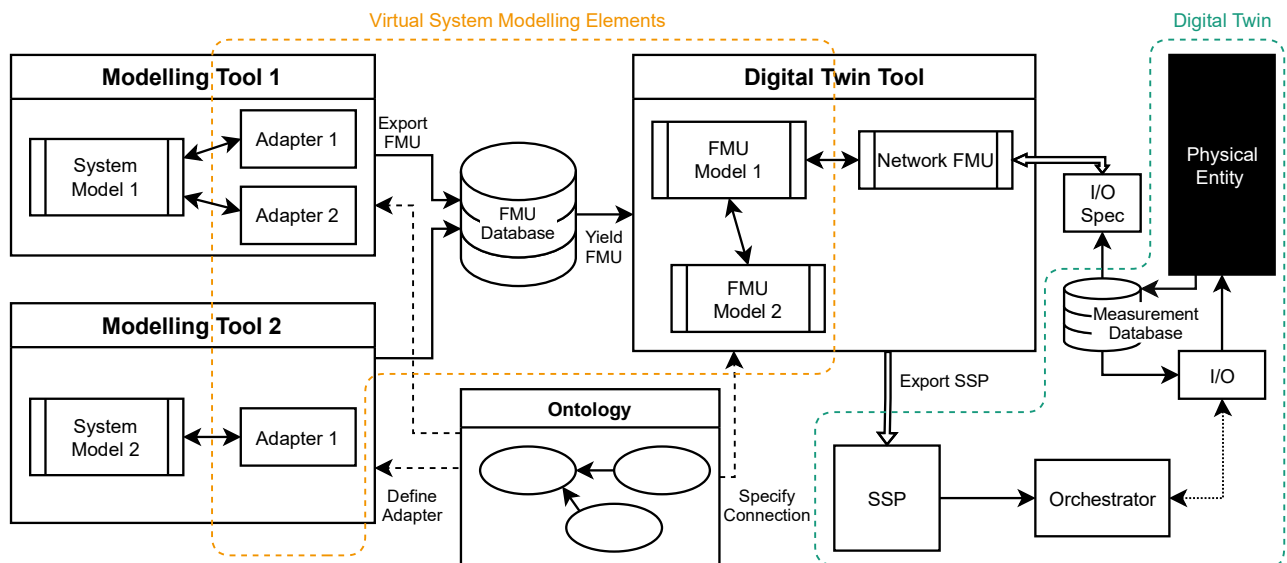


Figure 1 Visual representation of the development of digital twins for large modular multi-domain systems

or parameter data streams from the physical system to FMI input/outputs.

The implementation of multi-dimensional connectors that represent physical connections allows for easier handling of models with in- and output of high dimensions. Currently, in and output in FMUs are one dimensional and for high dimensional signals this results in hardly manageable input and output enumeration. While this drawback will be solved with FMI 3.0, wide adaptation in multi-purpose simulation tools, which are used to create FMUs, cannot be expected soon.

Finally, this leads to the idea of a dedicated development framework, which is visualized in Figure 1 along with the proposed modelling process. The framework accesses the database and eases the building of the digital twin based on single elements and real-world interfaces, similar to the “Open Simulation Platform” (DNV GL AS) for engineering with co-simulations. The adapters represent intuitive connectors which are based on the ontology: Connections between individual models are specified by the user on the top level and details are handled by the framework. A main differentiation between our approach and the open simulation platform is our desired full automation of the model assembly process. Instead of user-built models, the full-system model is derived from the ontology-stored structure information automatically. This creates further requirements for the tools used to assemble individual models into the full-system model. Furthermore, the use of co-simulation FMUs decouples the modeling and numerical solution of the individual simulation model (within one FMU) on the one hand from the simulation of the coupled full-system model on the other hand. The coupling of FMUs is described by the System Structure and Parameterization (SSP) Standard, which builds the output of the model assembler. The SSP and FMU files

are imported by an Orchestrator to run the virtual side of the digital twin.

Eventually, this procedure can be integrated into the development phase of a new product. The sensor layout can be optimized along with the product and a digital twin can be delivered faster, which is a step forward in the digitalization of renewable and especially wind energy.

2.1 General Process Description

We propose the following process for the structured development of model-based digital twins. The process can be separated into two distinct main steps, assuming the ontology is already defined beforehand. First, an FMU model database is created:

- Creating a simulation model: Use existing engineering models or develop new model. Use model reduction techniques to achieve real time capability, if necessary
- Define required adapter connection models in the specific modelling tool based on an ontology
- Connect the model to the corresponding adapter
- Export to FMU and upload into the database

This builds the base for development of digital models. The next step is the combination of single models into a complex system:

- Collect all components for the full-system multi-domain digital twin
- Connect single FMU models with a toolbox (see Section 2.2 for an example)
- Add the communication adapters for the digital twin

- Create SSP package with all FMU models and connection description
- Run Simulation using an orchestrator e.g., FMPy

The second step is basically the creation of the virtual entity for the digital twin. A complex virtual entity can be build based on component models, which are connected by intuitive adapters. The communication adapters take the responsibility of steering signals from inside the model to the outside for the digital twin.

2.2 The fmuToolbox

For ease of automated simulation, we implemented a Toolbox in Python for the setup of Co-Simulations based on SSP. Since the SSP file is written in an XML-Format, we use the `lxml` package in Python for file processing. The toolbox consists of multiple functions which represent the single steps of the Co-Simulation setup:

- XML-Description: Create a SSP with the component tag of one FMU. The information is extracted from the FMU with FMPy and written according to SSP
- Combination: Collect single XML description files into one SSP by listing all component tags in one file
- AutoConnect: Analyze the inputs-outputs and create connections tags for matching input-output pairs
- Packaging: All FMU files and the SSP description are packed together into an archive file format

Finally, the archive file can be simulated with FMPy orchestrator. The AutoConnect feature requires a standardized naming pattern for the in- and outputs. This can be built e.g., based on a common ontology.

3 Modelling Example: Hydrogen Electrolysis with Wind Energy

The setup of the full-system simulation model for the digital twin of a wind-to-hydrogen facility serves as modeling example. It is considered as a multi-domain system, as the domains wind energy and hydrogen generation are interacting. A digital twin of a wind-to-hydrogen facility would allow for operational optimization and error detection. Therefore, the system behavior needs to be modeled as closely as possible.

In this example, we concentrated on the following components: an aero-elastic wind turbine, wind turbine controller, generator, grid, transformer, converter, and electrolyser to build the wind-to-hydrogen facility. Separate models for the individual plant components exist in multiple simulation environments and are exported as FMUs. An ontology is defined for efficiently modelling and connecting all models from different tools and presumably different developers. The ontology is modelled according to the requirements of the specific use case. All implemented knowledge in the ontology has a specific purpose.

The exemplary ontology is based on the RDF Schema (Brickley et al.2014) to build upon the idea of classes, properties, and triplets. Our ontology defines the additional resources: “UseCase”, “Component”, “Connector”, “Signal”, “Measurement” and “Direction”. Classes and instances, which define the connection between components, are shown in Figure 2 (Remark: All visualizations of the ontology show a fraction of the single overall ontology). The ontology contains the definition of “use cases” to represent a specific simulation setup. An instance of “UseCase” like “Simulation_1” stores information about the instances of components by the property “:simulates”. Instances of any classes are identified by “rdf:type”. Only an instance of a wind

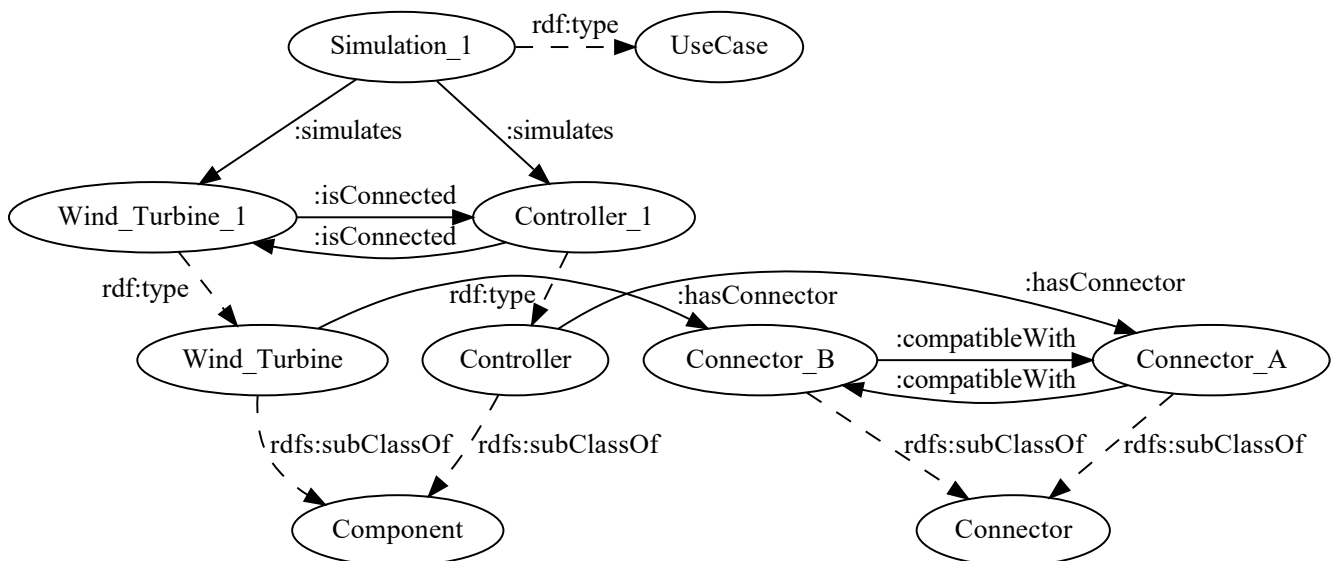


Figure 2 Example for the system ontology and interaction of parts

turbine, controller and their connectors are shown here to ensure clarity. The other components would be added to the ontology in the same manner. Instances of subclasses “Component” are defined to be simulated in a “UseCase” instance. Additionally, they can have any number of instances of subclasses of “Connector”. In the case of the component “Wind_Turbine” the only connector is “Connector_B”, which is compatible with “Connector_A”. Only when two subclasses of “Connector” are defined to be compatible with each other, then a valid connection between instances of “Component” can be made. The same applies for the “Controller” subclass and the corresponding connector instance. This, also shown in Figure 2, helps a modelling software to build valid connections.

The connectors are explained here for a simplified version of the controller connector. An example for the sub-ontology is shown in Figure 3.

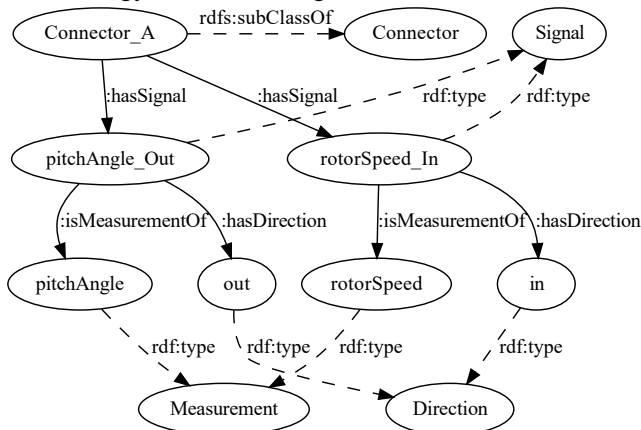


Figure 3 Example ontology for the definition of connectors

Instances of “Connector” have a list of instances of “Signal”. They are defined by their measurement and direction. A compatible connector would be built by inverting the direction of all signals in a connector. Names of input or outputs of FMUs can be derived from the ontology. The components and their required signals in this example are added to the full ontology in the described manner. Overall, all this information in the ontology is needed to create file according to SSP definition for the simulation with an orchestrator.

Moreover, specialized ontology tools, e.g., TopBraid Composer, enable to make a query to the ontology with SPARQL. A query as shown in Listing 1 is performed for the component class (Wind_Turbine) to get a list of its connectors. An in- and output list is obtained from the description of the found connectors, which define the required adapter model (as defined in Section 2).

Listing 1. SPARQL query example

```
DESCRIBE ?connector {
  ?subject rdfs:subClassOf :Component .
  ?subject :hasConnector ?connector
  FILTER(?subject = :Wind_Turbine)
}
```

After the ontology is fully defined, the models of the single components are of concern. The wind turbine has rated power of 8 MW and is modelled in MoWiT (Thomas et al. 2014) which is simulated with Dymola. Along with the wind turbine model, the controller is accessed through Dymola as well. For controller development, Matlab/Simulink is used. The Bladed-compatible DLL format is a widely adopted standard in wind energy for controller exchange. The Simulink model is compiled accordingly as DLL and tied into MoWiT using dedicated Modelica-code. The other models are created with Simulink and are directly exported as FMU with respect to the input and output names based on the ontology. All electrical components model the current flow, where the grid consumes all excessive energy. Lastly, the Electrolyser, which is an upscaled model (Espinosa-López et al.2018), uses up 1 MW for hydrogen production. These engineering models were validated in several research projects. The connection between the components is shown in Figure 4.

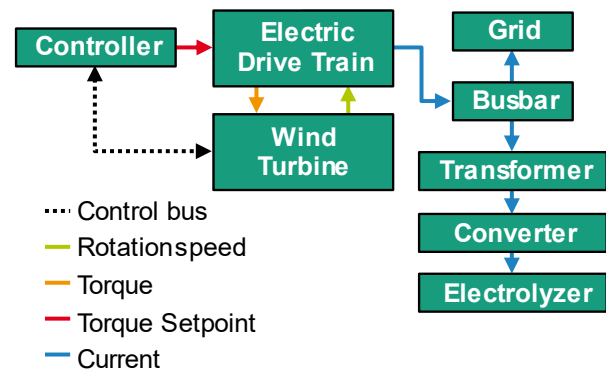


Figure 4 Connectivity diagram of the electrolysis process model

This model contains couplings in the drivetrain between the mechanical rotation and electromechanical torque. Both are controlled by the controller. The control exchanges a lot of data with the wind turbine (~ 40 connections), which are connected efficiently with the AutoConnect feature of the described toolbox. Furthermore, the electric drive train model has an energy output. The energy is distributed by current in the busbar to the grid and electrolyser. The transformer adjusts the voltage level, and the converter changes the current from AC to DC. In the end, the electrolyser consumes energy and produces hydrogen.

A demonstration of the simulation results is shown in Figure 5. The specific simulation case represents the startup of the wind turbine at constant inflow conditions. First, the wind turbine accelerates to enable power output and then continues to steady state operation. The electrolyser consumes the generated power. This process can be used to monitor the energy flow and detect errors when the physical model changes with respect to the digital model. To serve as digital twin for the real plant, the simulation model requires additional connections to take operational real-time data into account and to provide

a feedback into the plant operation. A full setup of a digital twin is not within the scope of this paper. The next step is to extend the presented process of building a digital twin full-simulation model onto operational data.

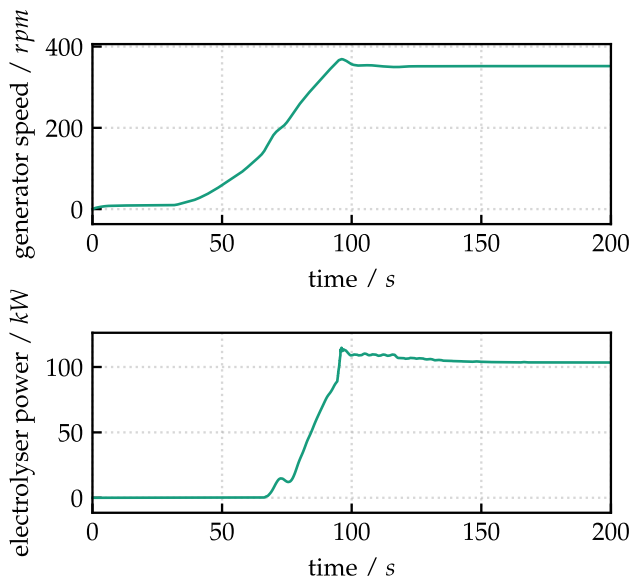


Figure 5 Simulation result for constant inflow conditions

4 Conclusion

In renewable energy and in particular in wind energy, highly modular systems require highly modular simulation models as well. Individual component models to exist from component engineering but linking them to a full-system model requires the efficient use of co-simulations using FMI. More standardization is needed in the development process of engineering models, to enable a fast connection process in the setup of the Co-Simulation. Ontologies offer a formalized way of the interface and overall system description. Thus, FMI has the potential to accelerate the development of multi-physics digital models, which can be utilized in digital twins. Model reduction techniques might be necessary to achieve real-time capability for digital twins or real time orchestrators need to be implemented. Furthermore, standardized interfaces and FMUs offers an easy way of adopting the level of detail of the digital twin, as models can be exchanged without further adjustments. To summarize, a standardized toolbox for Co-Simulation building shows high potential to enable digital twins.

Acknowledgements

The authors would like to thank Aline Luxa and Sebastian Frahm for their contribution of simulation models.

References

Branlard, Emmanuel, Jason Jonkman, and Scott Dana, and Paula Doubrawa (2020) "A Digital Twin Based on OpenFAST Linearizations for Real-Time Load and Fatigue

- Estimation of Land-Based Turbines." *J. Phys. Conf. Ser.*, vol. 1618, doi:10.1088/1742-6596/1618/2/022030.
- Brickley, Dan, Ramanathan V. Guha, and Brian McBride (2014) "Rdf Schema 1.1. W3C Recommendation." *World Wide Web Consortium*, vol. 2.
- Espinosa-López, Manuel, et al. (2018) "Modelling and Experimental Validation of a 46 KW PEM High Pressure Water Electrolyzer." *Renewable Energy*, vol. 119, pp. 160–73. doi:10.1016/j.renene.2017.11.081.
- Flender International GmbH *Digital Gearbox*. winergy-group.com/en/digital-gearbox. Accessed 6 May 2021.
- Fuller, Aidan, Zhong Fan, and Charles Day, and Chris Barlow (2020) "Digital Twin: Enabling Technologies, Challenges and Open Research." *IEEE Access*, vol. 8, doi:10.1109/ACCESS.2020.2998358.
- Glaessgen, Edward, and David Stargel (2012) "The Digital Twin Paradigm for Future NASA and U.S. Air Force Vehicles." *53rd AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conf.*, Honolulu, Hawaii, 53rd AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference. Reston, Virginia 2012.
- Gomes, Cláudio, et al. (2017) *Co-Simulation: State of the Art*. 1 Feb. 2017, arxiv.org/pdf/1702.00686.
- Gruber, Tom. "Ontology." *Encyclopedia of Database Systems*, edited by Ling Liu and M. Tamer Özsu, Springer New York, 2016pp. 1–3.
- Kritzinger, Werner, et al. (2018) "Digital Twin in Manufacturing: A Categorical Literature Review and Classification." *IFAC-PapersOnLine*, vol. 51, no. 11, pp. 1016–22. doi:10.1016/j.ifacol.2018.08.474.
- Modelica Association Project FMI *FMI Standard 2.0*. fmi-standard.org/downloads/. Accessed 12 Apr. 2021.
- Pimenta, F., et al. (2020) "Development of a Digital Twin of an Onshore Wind Turbine Using Monitoring Data." *J. Phys. Conf. Ser. (Journal of Physics: Conf. Ser.)*, vol. 1618, doi:10.1088/1742-6596/1618/2/022065.
- Schaeffler AG *Digital Services*. www.schaeffler.com/content.schaeffler.com/en/news_media/dates_events/windenergy_hamburg/digital_services/digital_services.jsp. Accessed 6 May 2021.
- Tao, Fei, He Zhang, and Ang Liu, and A. Y. C. Nee (2019) "Digital Twin in Industry: State-of-the-Art." *IEEE Transactions on Industrial Informatics*, vol. 15, no. 4, pp. 2405–15. doi:10.1109/TII.2018.2873186.
- Thomas, Philipp, et al. (2014) "The OneWind Modelica Library for Wind Turbine Simulation with Flexible Structure." *10th International Modelica Conf.*, March 10–12, 2014, The 10th International Modelica Conf. Linköping University Electronic Press, 2014pp. 939–48.
- Tygesen, Ulf T., et al. (2018) "The True Digital Twin Concept for Fatigue Re-Assessment of Marine Structures." *Volume 1: Offshore Technology*, 17.06.2018 - 22.06.2018, Madrid, Spain, ASME 2018 37th International Conference on Ocean, Offshore and Arctic Engineering 2018.
- van Nguyen, Yvon Besanger, and Quoc Tran, and Tung Nguyen (2017) "On Conceptual Structuration and Coupling Methods of Co-Simulation Frameworks in Cyber-Physical Energy System Validation." *Energies*, vol. 10, no. 12, doi:10.3390/en10121977.