Available online at www.sciencedirect.com

ScienceDirect

IFAC PapersOnLine 55-19 (2022) 97–102

Digital Twins modeling and simulation with Node-RED and Carla

Charles Steinmetz * Greyce N. Schroeder ** Alecio Binotto * Sanjay Panikkar **** Brian Papenfuß † Christian Schmidt † Achim Rettberg * Carlos E. Pereira ****

* *HSHL & UOL, Lippstadt, Germany, (charles.steinmetz@hshl.de, achim.rettberg@hshl.de).*
** *PPGEE/UFRGS, Porto Alegre, Brazil, (greyce.nogueira@ufrgs.br, cpereira@ece.ufrgs.br)*
*** *IBM Services, Munich, Germany, (alecio.binotto@ibm.com)*
**** *IBM Services, Bangalore, India, (psanjay@in.ibm.com)*
† *UOL, Oldenburg, Germany, (brian.papenfuss@uni-oldenburg.de, christian.schmidt1@uni-oldenburg.de)*

Abstract: Digital Twin (DT) is a key concept for digitization in the Industry 4.0. Two main elements of this concept are simulation and modeling since it allows engineers to have a digital representation of the system as well as enables them to test new configurations without changing the running system. Simulation of a real-world environment can be a challenge because it involves many different entities and variables. For example, in a smart city scenario, there might exist services that people can use for transportation, healthcare, delivery, energy supply, and so on. Creating this kind of scenario can be difficult since not all data is digitally available, and its digitalization might not be possible to be done. In this context, having a virtual environment capable of emulating the main components is an important piece for building these applications, not only to simulate new scenarios but also to prototype initial concepts. This paper proposes the use of Node-RED and Carla simulators to provide a platform witch DT-based systems can be model and simulated. The focus of this approach is to provide a concept that reduces the gap between models and real-world applications as well as connecting virtual representations to real-world applications keeping all parts synchronized. A use case of a Car-as-a-Service (CaaS) system has been implemented to demonstrate the applicability of the proposed concept.

Keywords: digital twin, simulation, model, synchronization, carla simulator, node-red, Smart City

1. INTRODUCTION

The Digital Twin is one of the essential concepts in the context of Industry 4.0, and its importance is continuously growing in academia and industry (Tao et al., 2019).

Applications can be built on the top of this concept and then be exposed as services (Aheleroff et al., 2021) to the world, enabling new kinds of business models such as Car-as-a-Service. Furthermore, this service is changing the way people see and use cars since digitalization opens new possibilities for interaction with final users.

These new application concepts are going to become a reality in the next few years. Therefore, research on software specifications and deployment is still necessary to be addressed, mainly in the direction of creating and updating standards, methodologies, and modeling languages (Petrasch and Hentschke, 2016).

Real-world applications can be composed by several assets that can be virtually represented in different ways and perspectives (Qi et al., 2018). Therefore, standards for modeling and integrating heterogeneous models are necessary (Yun et al., 2017).

Moreover, even though applications nowadays are producing a huge amount of data, this data is isolated, fragmented, or stagnant, which decreases the level of efficiency, intelligence and sustainability in the product life cycle (Fei et al., 2018). Keeping models close or connected to the real application can be one possible solution to keep the model as a mirror of the real world.

In this context, this paper proposes the use of Node-RED (NodeRED, 2022) for modeling and deploying DTs, combined with Carla simulator for building the use case scenarios. This allows models to be constantly updated by connecting them to the applications.

The paper is organized as follows: Section II presents the DT concept, applications related to it, and some available tools for modeling and deployment; in section III, the proposed approach is presented and described; in section IV, a use case implementation and discussion are shown

10.1016/j.ifacol.2022.09.190

and, in section V conclusions and future work directions are signaled.

2. DIGITAL TWIN

This section aims to give an overview of the DT concept, its applications, and how modeling and deployment can be done.

2.1 Digital Twin concept

The Digital Twin concept was presented by Michael Grieves in 2003 in his course on "Product lifecycle management" at the University of Michigan (Grieves, 2014). Later in 2012, the concept was reviewed by the National Aeronautics and Space Administration (NASA) as a multiphysics, multiscale, probabilistic, ultra fidelity simulation that reflects the state of an asset based on real-time and historical data, as well as physical models (Glaessgen and Stargel, 2012).

For Rosen et al. (2015), the DT is a virtual reflection of a physical asset which, via data and simulators, enables controlling, optimization, and monitoring and improves the decision making. Through the Internet of Things, the DT can bring several benefits such as simulations, assets monitoring, and management since it connects both physical and virtual worlds storing data of their whole lifecycle.

2.2 Digital Twin use cases

This section brings some applications available in the literature that use the concept of DT.

In manufacturing, there are several applications developed on top of the DT concept (Cimino et al., 2019). In (Qi et al., 2018) the authors specify how manufacturing services and DT can be combined, enabling manufacturers can use components of the system as services.

Industrial applications might have old machinery or complex production lines that can bring challenges to performing tests and deployments, especially while the plant is running in production. Therefore, the authors of (Ayani et al., 2018) presented an industrial application that uses an emulation model to maximize the use of DT and minimize the risks of stopping the production. Using emulation techniques is also part of the approach of this paper, which uses Carla to run a smart city scenario.

Smart city applications have also been developed using the DT concept (Farsi et al., 2020). This enables, for instance, to create systems for disaster management (Ford and Wolf, 2020). It also allows people to monitor, interact and give feedback regarding their own cities, improving the quality of life (White et al., 2021).

Still, in the context of cities of the future, where it will be possible to use products (or "things") as services, a concept for enabling Car-as-a-Service using DT is given in (Steinmetz et al., 2021b).

2.3 Digital Twin modeling and deployment

Modeling and deployment of DT have been the focus of several researchers (Rasheed et al., 2020) since there is a lack of standards for it (Yun et al., 2017). Some of the main approaches that address this topic are considered for this paper.

Schroeder et al. (2016) use AutomationML (Automation Markup Language - AML) in their methodology for modeling data exchange for digital twin systems. In similar way, other researches such as (Sierla et al., 2018), (Um et al., 2017) and (Zhang et al., 2020) propose the use of AutomationML to create a model for the DT.

Ontologies provide a powerful semantic way of modeling knowledge. A methodology based on ontology is given in (Bao et al., 2022), in which the authors propose a part digital twin modeling approach that can help suppliers as well as assembly workshops. Ontologies can also be used for modeling relationships between different equipments (Xu et al., 2015).

DT can be applied in many different areas with highly dynamic applications, which opens a challenge to track all changes in the real world. Therefore, research on how to keep models updated and accurate is still needed (Wright and Davidson, 2020).

2.4 Digital Twin and IoT modeling technologies

As the DT is continuously evolving and being increasingly applied in different use cases, new technologies have been proposed to support this concept.

Microsoft has proposed the Azure Digital Twin (Microsoft, 2021). It is an IoT platform that enables the creation of digital representations of real-world elements. An open modeling language called *Digital Twins Definition Language* is used to create the models. It is also possible to monitor live data coming from the real world and make queries on the model.

Another well-known tool in the context of IoT is the Node-RED (NodeRED, 2022). It is a programming tool that enables it to connect to hardware devices, APIs, and services. It has some similarities to Azure DT since it also provides a graphical interface for creating nodes and connections between them. However, Node-RED also offers the possibility to create custom flows with programming code that can be reused. It has been used in many IoT applications such as sensor integrator (Lekić and Gardašević, 2018), home automation (Kodali and Anjum, 2018) and others.

In (Thuluva et al., 2020) a set of nodes for Node-RED are defined to introduce semantic definitions such as *iot.schema.org*. This allows system designers and engineers to add semantics to the model consistently.

Most of the available tools in the market are still focused on the IoT concept, mainly connecting sensors and actuators to the cloud. In this context, this paper proposes an approach based on the ISO 23247, which enables the modeling of digital twins with standards elements. For this, the use of Node-RED for modeling and deploying DTs, following the elements provided by the ISO 23247.

3. USING NODE-RED FOR MODELING AND DEPLOYING DIGITAL TWIN

This work proposes the use of Node-RED for modeling and deploying DT. Therefore, key components based on the ISO 23247, organized into four layers, have been used. These layers help to understand the responsibility of each element.

3.1 Key-components for modeling DTs

This work is based on Steinmetz et al. (2021a) which defines key components for modeling DT. These elements follow the ISO 23247, and they are arranged into four layers: Observe, collect, model, and learn & act. Figure 1 shows the DT elements grouped by their responsibilities.

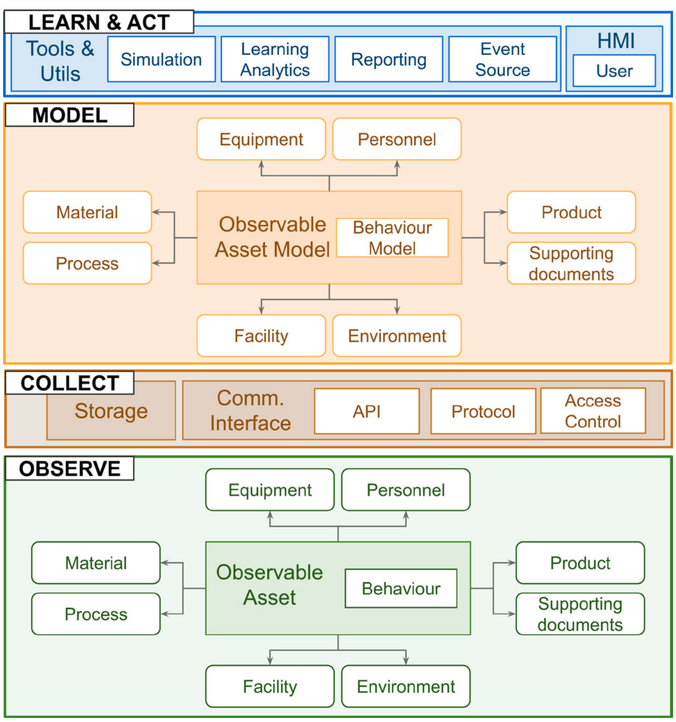

Fig. 1. Digital Twin elements Steinmetz et al. (2021a)

Data is collected in the Observe layer and transmitted via the Collect layer. Next, all models can be updated with actual data and be available to the last layer, Learn and Act. Furthermore, control signals can be sent back to the observable elements, which can change their state in the real world, as can be seen in Figure 2.

Each observable asset has a state which is composed of its current properties or parameters. Every time any change on this asset data happens, a new state is created. Also, assets can contain actions that can be performed or events that can be triggered and might cause changes and, consequently, create a new state.

3.2 DT nodes for Node-Red

For enabling the modeling of DT-based systems, some nodes have been created to be used in the Node-red platform.

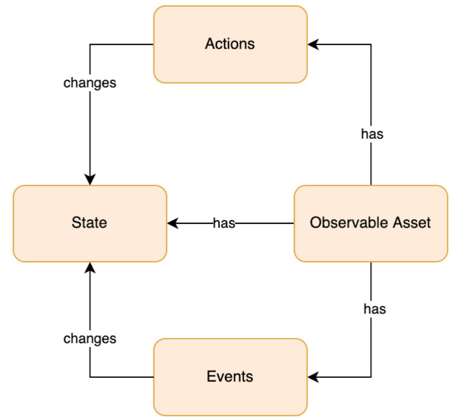

Fig. 2. Observable asset state

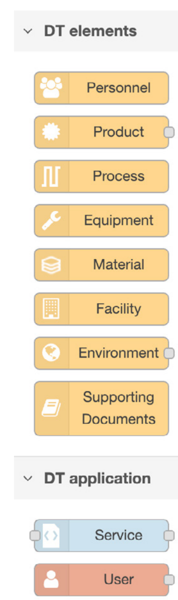

Fig. 3. NodeRED nodes

Figure 3 shows the developed nodes already imported in the modeling tool.

Using these nodes, flows can be built to represent a process or activity from the real world. It is possible to mode all elements and their relations as well as the data that is exchanged between them.

DT elements *Personnel*, *Product*, *Process*, *Equipment*, *Material*, *Facility*, *Environment*, and *Supporting Documents* (in yellow) are defined by the ISO 23247 for the manufacturing context, however they can also be used in other use cases.

Additionally, components for DT application have been defined:

- **User:** represent who will use the available services.
- **Service:** is how applications can be exposed to the world and the consumed by users.

Applications can be built on top of the mapped elements and can be exposed as services to the external world. These services can be executed by third-party platforms as long as they use the same communication protocol.

4. USE CASE IMPLEMENTATION

This section presents a use case implementation to demonstrate how the proposed concept can be applied in the CaaS scenario. This implementation is scoped to the main functionality of CaaS: requesting a car and driving the user from a point A to a point B. Therefore, a user can request a car and be taken to the destination. The main idea is to show how high-level models in NodeRED, together with a simulator, can enable the implementation of complex use cases.

4.1 Implementation of the application flows

Figure 4 illustrates some of the flows implemented for this use case. The main idea is to show how the components can be arranged and reused in the system.

- 1 - represents the highest level model of the applications. This representation level should not contain many technology-specific elements. This approach allows users without great technical skills but with solid knowledge of the use case to build this kind of system. It also enables the creation of applications without limiting them to specific hardware or platform.
- 2 - this bloc is part of the second level, where defined components such as auth, getBestCar, and log are used to define the CaaS. All inputs pass through authentication, which only allows messages are processed by the system. When a request for a car is received, the getBestCar is performed to select the car that fits better to the users who are requesting it. This selection can be based on several criteria, but at this level, engineers only see a visual representation of this process.
- 3 - also, in the second level, the third bloc shows how user data can be received via MQTT protocol from the real world. Like in step 2, all data pass through authentication and then is processed by a function that saves it in a database. This keeps the history of the users' data while using the system. Other elements, such as cars, have a similar flow, and their data is also stored in a database.
- 4 - the third level is composed of technical details about how each block should work. For instance, the authentication receives data and parses it to a JSON object to be processed by a function. This function can be implemented in different ways based on the algorithm that fits the best for the application. Finally, this flow provides two possible outputs: allowed or denied. This output is used in the upper blocs to continue or not the application flow.

4.2 Using Carla to simulate a Smart City environment

Real-world applications like the ones that will exist in smart cities might not be possible to build due to several factors such as complexity, price, availability of data in a digital form, and so on. Therefore, having ways of simulating this kind of scenario is essential to develop new ideas and test different configurations and possibilities.

In this use case, the Carla simulator has been used to create the scenario of the CaaS application. Figure 5 shows the Carla simulator with several actors.

Cars can be spawned in the scenario, and they will drive autonomously around the city. The number of cars can be changed based on the requirements of the application.

Likewise, users (walkers) can also be inserted into the scenario. They will walk randomly on the sidewalk while the simulator is running.

As Carla is an open-source project, modifications are possible, and this makes this platform suitable for different scenarios. By customizing each implementation of the cars and users, it is possible to send data from the simulated world to external systems via HTTP requests. In this way, these actors can be watched, and their virtual representations can be updated with their current state.

Users and cars are continuously sending data via MQTT to a defined broker:

- **cars:** *id*, *location*, *status* (busy or free). These actors are also listening for ride request at a specific MQTT topic.
- **users:** *id*, *location*. Additionally, users can request a car to take them to a desired *destination*.

When a user requests a car, his/her data is sent to the CaaS application, together with the desired location. As seen in Figure 4 (2), the *getBestCar* function selects a car based on a defined criteria. In this case, the criteria were the distance, but this could be implemented in several different ways, taking into account accessibility, user preferences, and so on.

As soon as the selected car is notified about the user request, the path is calculated from its location to the user and later to the final destination. The *waypoints* of a ride can be observed in blue in the Figure 5. While the ride is happening, the selected car changes its status to busy and switches back to free when it releases the user.

For better visualization, in this use case, only a few actors have been inserted in the scenario, however this can be dynamically changed.

4.3 Discussion

This use case has presented an implementation of DTs in the context of CaaS, using Node-RED for modeling the virtual representation and Carla simulator to build the simulated real-world scenario.

With the elements provided by ISO 23247, it is possible to model a system in an abstract way, leaving the technological details to be defined in the refinements of the model. This enables users without a deep technological background but a good knowledge of the use case to model by only connecting inputs and outputs of the components.

The generated top-level model, as well as the specific ones, can be reused to build future applications. For instance, the components *User* and *UserManager* can exist in many applications, but their implementations can be different in each case. Also, the *auth* bloc can be reused through all external communications that the applications may have.

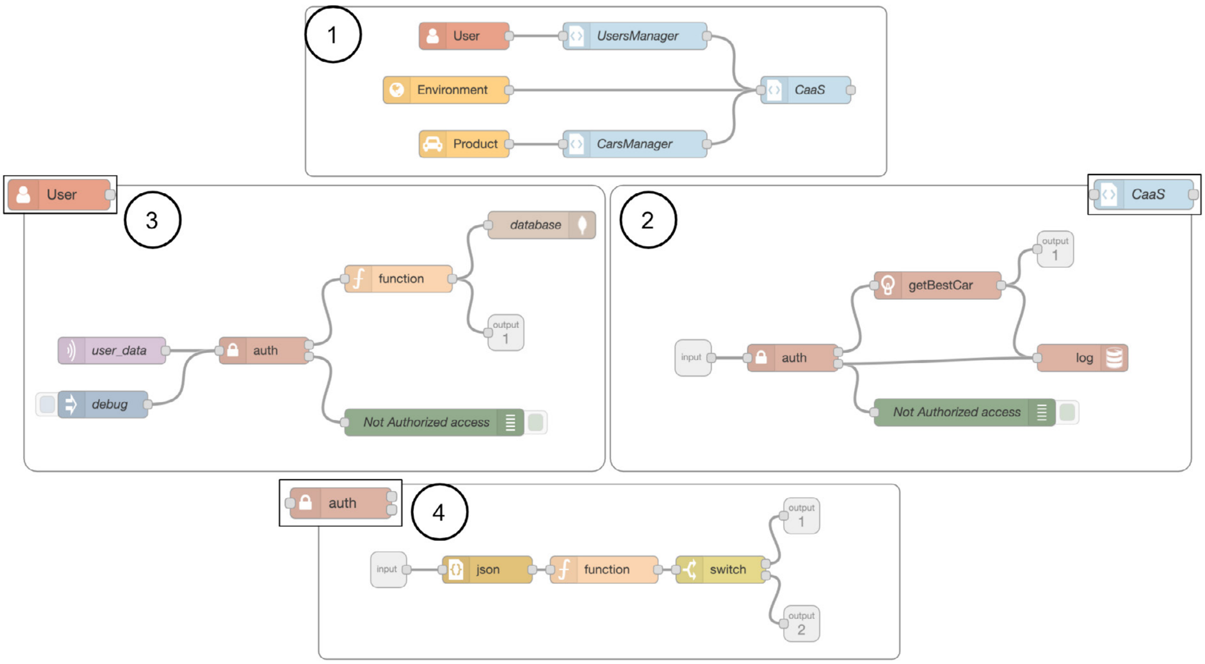

Fig. 4. Node-RED flows for a CaaS application

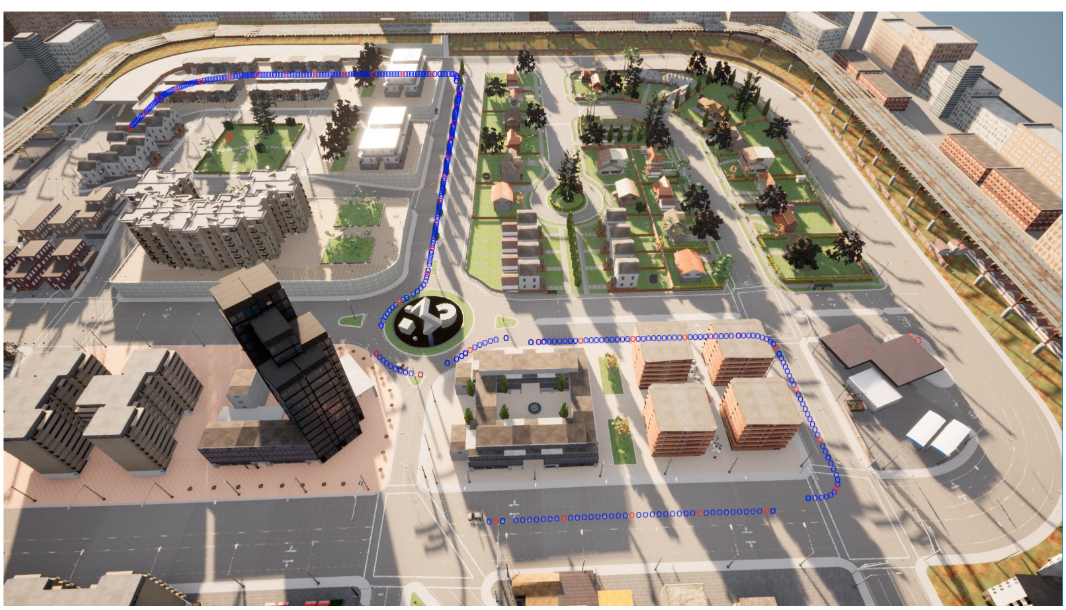

Fig. 5. Carla environment with a planned path

5. CONCLUSION

This work has proposed the use of Node-RED as a tool for modeling DT applications with elements defined by the ISO 23247. It allows applications to be built following a defined standard, making the models more maintainable and understandable.

To emulate the real world scenario, the Carla simulator has been used. This tool makes it possible to create different scenarios related to services available in a smart city, such as CaaS.

Simulating scenarios like this enable engineers to create scenarios that are too complex to reproduce in the real world. For instance, it can be too expensive, too dangerous or it is not possible to collect data in a digital way.

Connecting the model to the application brings several benefits, such as keeping the virtual representation updated with the last state of the physical world. Therefore, Node-RED is a potential tool to be explored for developing new concepts based on DT.

The proposed approach can be used as the basis for future works that need to implement digitalization concepts in the context of future business cases in the mobility area such as CaaS.

In bigger applications with more components, the models may become large and complex, therefore a topology in layers and performance evaluations are planned for future work. Enabling different models to be integrated or linked to each other, as well as adding semantics to these relationships, can be necessary, especially for existing applications that already have parts modeled. Also, integration to already existing tools is also a feature that can enhance the proposed approach.

REFERENCES

Aheleroff, S., Xu, X., Zhong, R.Y., and Lu, Y. (2021). Digital twin as a service (dtaas) in industry 4.0: An architecture reference model. *Advanced Engineering Informatics*, 47, 101225.

Ayani, M., Ganebäck, M., and Ng, A.H. (2018). Digital twin: Applying emulation for machine reconditioning. *Procedia Cirp*, 72, 243–248.

Bao, Q., Zhao, G., Yu, Y., Dai, S., and Wang, W. (2022). Ontology-based modeling of part digital twin oriented to assembly. *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture*, 236(1-2), 16–28.

Cimino, C., Negri, E., and Fumagalli, L. (2019). Review of digital twin applications in manufacturing. *Computers in Industry*, 113, 103130.

Farsi, M., Daneshkhah, A., Hosseinian-Far, A., and Jahankhani, H. (2020). *Digital twin technologies and smart cities*. Springer.

Fei, T., Jiangfeng, C., Qinglin, Q., Zhang, M., Zhang, H., and Fangyuan, S. (2018). Digital twin-driven product design, manufacturing and service with big data. *The International Journal of Advanced Manufacturing Technology*, 94(9-12), 3563–3576.

Ford, D.N. and Wolf, C.M. (2020). Smart cities with digital twin systems for disaster management. *Journal of management in engineering*, 36(4), 04020027.

Glaessgen, E. and Stargel, D. (2012). The digital twin paradigm for future nasa and us air force vehicles. In *53rd AIAA/ASME/ASCE/AHS/ASC structures, structural dynamics and materials conference 20th AIAA/ASME/AHS adaptive structures conference 14th AIAA*, 1818.

Grieves, M. (2014). Digital twin: Manufacturing excellence through virtual factory replication. *URL http://www. apriso. com*.

Kodali, R.K. and Anjum, A. (2018). Iot based home automation using node-red. In *2018 Second International Conference on Green Computing and Internet of Things (ICGCIoT)*, 386–390. doi: 10.1109/ICGCIoT.2018.8753085.

Lekić, M. and Gardašević, G. (2018). Iot sensor integration to node-red platform. In *2018 17th International Symposium INFOTEH-JAHORINA (INFOTEH)*, 1–5. doi:10.1109/INFOTEH.2018.8345544.

Microsoft (2021). Azure digital twins. URL `https://azure.microsoft.com/de-de/services/digital-twins/`.

NodeRED (2022). Node-red low-code programming for event-driven applications. URL `https://nodered.org`. Accessed: 2021-08-15.

Petrasch, R. and Hentschke, R. (2016). Process modeling for industry 4.0 applications: Towards an industry 4.0 process modeling language and method. In *2016 13th International Joint Conference on Computer Science and Software Engineering (JCSSE)*, 1–5. doi: 10.1109/JCSSE.2016.7748885.

Qi, Q., Tao, F., Zuo, Y., and Zhao, D. (2018). Digital twin service towards smart manufacturing. *Procedia Cirp*, 72, 237–242.

Rasheed, A., San, O., and Kvamsdal, T. (2020). Digital twin: Values, challenges and enablers from a modeling perspective. *IEEE Access*, 8, 21980–22012.

Rosen, R., von Wichert, G., Lo, G., and Bettenhausen, K.D. (2015). About the importance of autonomy and digital twins for the future of manufacturing. *IFAC-PapersOnLine*, 48(3), 567–572.

Schroeder, G.N., Steinmetz, C., Pereira, C.E., and Espindola, D.B. (2016). Digital twin data modeling with automationml and a communication methodology for data exchange. *IFAC-PapersOnLine*, 49(30), 12–17.

Sierla, S., Kyrki, V., Aarnio, P., and Vyatkin, V. (2018). Automatic assembly planning based on digital product descriptions. *Computers in Industry*, 97, 34–46.

Steinmetz, C., Nogueira Schroeder, G., Rodrigues, R.N., Rettberg, A., and Pereira, C.E. (2021a). Key-components for digital twin modeling with granularity: Use case car-as-a-service. *IEEE Transactions on Emerging Topics in Computing*, 1–1. doi: 10.1109/TETC.2021.3131532.

Steinmetz, C., Schroeder, G.N., Rettberg, A., Rodrigues, R.N., and Pereira, C.E. (2021b). Enabling and supporting car-as-a-service by digital twin modeling and deployment. In *2021 Design, Automation & Test in Europe Conference & Exhibition (DATE)*, 428–433. IEEE.

Tao, F., Zhang, H., Liu, A., and Nee, A.Y.C. (2019). Digital twin in industry: State-of-the-art. *IEEE Transactions on Industrial Informatics*, 15(4), 2405–2415. doi: 10.1109/TII.2018.2873186.

Thuluva, A.S., Anicic, D., Rudolph, S., and Adikari, M. (2020). Semantic node-red for rapid development of interoperable industrial iot applications. *Semantic Web*, 11(6), 949–975.

Um, J., Weyer, S., and Quint, F. (2017). Plug-and-simulate within modular assembly line enabled by digital twins and the use of automationml. *IFAC-PapersOnLine*, 50(1), 15904–15909.

White, G., Zink, A., Codecá, L., and Clarke, S. (2021). A digital twin smart city for citizen feedback. *Cities*, 110, 103064.

Wright, L. and Davidson, S. (2020). How to tell the difference between a model and a digital twin. *Advanced Modeling and Simulation in Engineering Sciences*, 7(1), 1–13.

Xu, W., Yu, J., Zhou, Z., Xie, Y., Pham, D.T., and Ji, C. (2015). Dynamic modeling of manufacturing equipment capability using condition information in cloud manufacturing. *Journal of Manufacturing Science and Engineering*, 137(4).

Yun, S., Park, J.H., and Kim, W.T. (2017). Data-centric middleware based digital twin platform for dependable cyber-physical systems. In *2017 Ninth International Conference on Ubiquitous and Future Networks (ICUFN)*, 922–926. IEEE.

Zhang, H., Yan, Q., and Wen, Z. (2020). Information modeling for cyber-physical production system based on digital twin and automationml. *The international journal of advanced manufacturing technology*, 1–19.