Digital Twin preparation for the prototyping phase, a use case

Gernot Pöchgraber[1], Sébastien Bougain[2] ,Thomas Trautner[1], Niphon Jeepjua[2] and Friedrich Bleicher[1]

1 IFT - Institute of Production Engineering and Photonic Technologies, TU Wien, Getreidemarkt 9, 1060 Vienna, Austria
2 CDP Center for Digital Production GmbH, Seestadtstraße 27/19, 1220 Vienna, Austria

gernot.poechgraber@tuwien.ac.at

Abstract. Digital Twin (DT) is a growing topic within the manufacturing industry. Moreover, early adoption of a Digital Twin is believed to benefit the rest of the product's life cycle, like production, maintenance, and recycling. As the physical counterpart of the product is built only in the prototyping phase, this step is thought to be the first phase of a Digital Twin. Still, it necessitates preparation to reduce time-to-market and hence the development costs. This paper shows the result of a Digital Twin preparation before the implementation on the prototype. This consists of creating a Digital Twin from the prototype of a complex logistic plant with a Discrete Event Simulation (DES) tool, and a dummy Open Platform Communications Unified Architecture (OPCUA) server communicating with each other. The methodology used is described with the use case of a logistic system simulated with an OPCUA Server on a Raspberry Pi. Although this work is specific to a particular use case, other researches may profit from the methodology applied and the experiences gathered for implementing a Digital Twin in the early phases of product development.

Keywords: Digital Twin, Discrete-Event Simulation, Product Development.

1 Introduction

Digital twins (DT) are created for a wide variety of purposes. Be it validation or also for predictive forecasting of scenarios [1]. In unique machine construction, especially in the logistics sector, the mechanical construction is often completed, and troubleshooting can only begin once a complex control system with integrated plant logic has been implemented. For developing a new product for logistic purposes, a simulation is built to check assumptions, assess the system capabilities and optimised if necessary. Yet to accelerate later optimizations and create new functions for the product (self-optimization, accuracy measurement, forecast), this simulation environment can be extended with DT capabilities. Hence the research aims to create a digital twin of the prototype of a logistic system being built with the help of a discrete event simulation environment. On the one hand, the simulation model controls the logic for testing the mechanical components. On the other hand, the real data obtained is used to demonstrate optimisations to the technical system. The approach's goal is to save resources in product development. Both aspects should be prepared before implementation within the prototype. This research focuses on how to prepare a Digital Twin for an implementation in the prototyping phase.

2 State of the Art

In 2005 the Eller School of Business at the University of Arizona introduced the Mirrored Space Model (MSM). Grieves defines the three elements of an MSM: (1) the real space, (2) the virtual space(s) and (3) linking mechanisms between the real and virtual space(s) [2]. One significant difference between MSM to other known systems at that time is how data was gathered and used: from now on, data was related to the virtual representations of their physical objects and independent of their functions. Taking the MSM concept into account and expanding it by data and information flow, the basics for a DT are set. The term Digital Twin as a virtual representation of its real, physical counterpart was seeded [3].
Building on this research, and although the term DT is defined in ISO23247, it is not always used and no single definition of DTs has emerged, leading to different uses of this term in different disciplines in the research and industry sectors. However, further differentiations can be categorised [1]: Kritzinger et al. define the Digital Model (DM) as a physical object's representation in the virtual space with no automated data- or information transfer between the virtual and real-physical space. A DM, which allows an automatic, single-directional communication from the physical space to the virtual space object, is defined as a Digital Shadow (DS). With the capability for a DM to communicate with the real system automated and bi-directionally, the authors describe the Digital Twin. The three concepts are depicted in Figure 1.

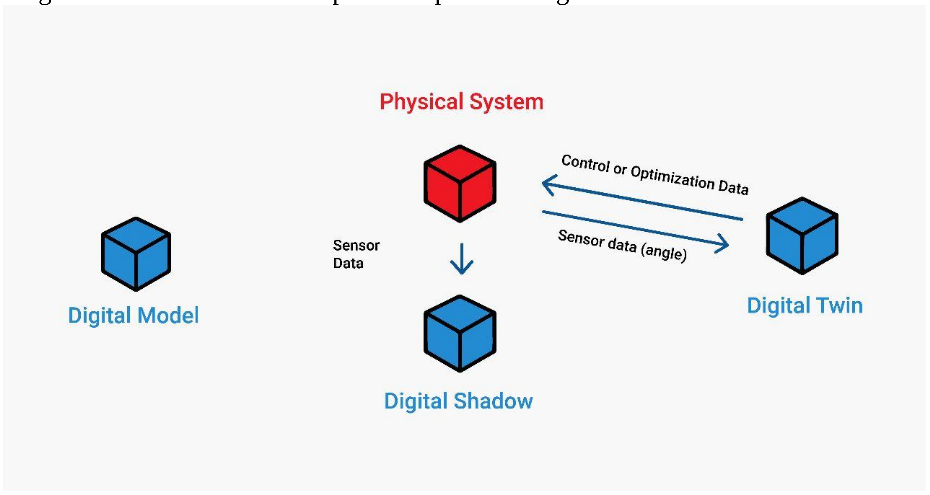


Figure 1, Digital Model, Digital Shadow and Digital Twin [1]

2.1 Digital Twin and Product development

The first phase applicable for a Digital Twin is the prototyping phase, as no physical counterpart exists before that. Hence research are conducted on the application of Digital Twin in the early stages of product development. Early adoption of a Digital Twin benefits the rest of the product's life cycle (production, maintenance and recycling) [12]. Moreover, using a Digital Twin for product design and development is also believed to shorten the development time by virtual prototyping, employing simulation and supporting decision-making. It has yet to be widely tested and needs further investigation [13].

2.2 Framework

The ISO23247 delivers a standardised digital twin framework for manufacturing. Before the ISO framework was published, the scientific community worked on several, mainly for the manufacturing industry [4]. The framework developed in those researches shares the same architecture in four parts: Physical Space for the real system and the sensors, the Virtual Space for hosting a virtual representation of the real system, the Service Space which contains functions responsible for the analysis of the system and the Data Space holding data information saved throughout the Digital Twin´s lifecycle [5].

2.3 Discrete Event Simulation for Digital Twin

Event-driven simulation or Discrete Event Simulation (DES) is a type of discrete simulation. In DES, the simulation progress is made by processing a list of events. Discrete Event Simulation can cover three of the four parts of the above-mentioned Digital Twin framework. A Digital Twin´s framework consists of, the Service Space for simulating consequences, the Virtual Space for representing the Physical Space virtually and the Data Space for holding data. In this research, the FlexSim Simulation Software package is used, which allows the modelling of real-life systems, DES, with visualisation and virtual reality experience. Customised dashboards and 3D animations in FlexSim provide high usability to track problems and quickly find alternative solutions. In 2017 FlexSim introduced an interface which allows connecting the virtual models to their physical counterparts. In the active connection mode, sensors and control data from Programmable Logic Controllers (PLC) can be transferred via an Open Platform Communications Unified Architecture (OPCUA) Server or with PLC using the serial communication protocol Modbus. This data has to be mapped adequately to software variables. As OPCUA is mainly used for data exchange in industrial automation [6], FlexSim is a good choice for prototyping and realising DT concepts. As mentioned above, DTs help to understand what may happen and what is happening in a process. A DT helps to control and optimise systems continuously with the help of real data. A DES can work as a DS on the way to a DT. This happens when real-time data utilisation is applied for the model to follow the real systems modifications. How communication with the real system occurs is essential for the DT. To reflect the real state, parameters

can be updated periodically. This is very important that the DES can provide adjustments or decisions for the DT and the real system. To achieve this, data-driven- or agent-based DES models are necessary [7].
Karakra et al. built a decision support tool based on a DT for real-time monitoring and predicting patients pathways in hospitals using DES. The connection of discrete event sensors like Wi-Fi, NFC, RFID etc., transforms the offline DES model into an online DES model. Karakra et al. use a three-model system. The process model is an online model which simulates the real data, whereas both the replay and prospective models are offline models. The replay model only accesses historical data and simulates past events, the prospective model randomly arranges the historical data and thus can affect future events. The two offline DES models are transferred to online models and used for monitoring and forecasting. The initialisation is performed with the connection to a real-time database. After validation of the two offline models with historical data and synchronisation with the real-time clock, the DT, as a clone, represents the current state of the hospital. The DES model's parameters must be synchronised with the real systems parameters to obtain the same state or value. On the one hand, Karakra et al. use the DT to predict the near future using a faster simulation run of the DES than the real-time. On the other hand, by simulating with the same clock speed, the DT can be used as a monitoring system. The DT gives a proactive view of the hospital, and based on those, bold decisions can be made [8].

2.4 Bidirectional Communication

Open Platform Communications Unified Architecture (OPC UA) aims to connect all industrial systems with a uniform and information model-based communication. The OPC UA standard consists of information modelling and an ethernet-based transport protocol. Hence the bidirectional communication between a physical prototype (PT) and a digital prototype (DP) can be realised with OPC UA [9]. Kutscher et al. solve the connections of a PT and DP with a central information point (CIP) as an import and export link between them. The CIP is an interconnection between PT and DP as a OPC UA server or client. The OPC UA standard combines standardised information modelling in one technology [10]. Protic et al. use OPC UA servers and clients to communicate between two cobots with other system devices. Data like the speed of joints and the tool centre point is synchronised with the CAD model in NX. The built-in OPC UA Server cannot interconnect directly with the OPC UA Server of the cobots. For that reason, the virtual simulation module is linked with a python program using an OPC UA server/client, which provides updated data [11].

3 Methodology

This research's methodology aims to realize a Digital Twin with a simulated prototype for a later live implementation as soon as the real prototype is built., in order to save ressources as data from the prototype could be gathered instantly and used within the

Digital Twin's Service Space for analysis purposes. To achieve this, the DES Environment, used as a Virtual Space, is connected to an OPC UA Server reproducing the data structure of the physical system on a Raspberry Pi. Moreover, a specific Service Space is developed, interacting with a DES solution, adapted for Digital Twin purposes to fit the system to development purposes (data gathering, comparison with simulations predictions) and future product services (self-optimization). The goal of the Service Space is to manage specific events that the real system is facing, simulate the event's consequences, test various solutions, inform the Virtual Space of its results and take appropriate measures to be sent to the PLC. Then specific events are triggered to test the solution's viability before extending it to its full capacity. This will constitute a DT before implementation with the real prototype. FlexSim is used as the DES solution, providing a PLC emulation solution. Yet FlexSim, as of version 20.2.3 used for this research, does not contain any preinstalled Digital Twin Management solution that would be useful for the research, e.g. managing several instances of one model. To achieve this Service Space, two solutions are possible: (a) the logic is written within FlexSim along with a save and load system being able to reload a model´s state without resetting the whole simulation and breaking the Digital Twin management flow or (b) the management system is defined in a separated program that runs the model and can be executed by FlexSim. A separate script managing the instances and delivering the result to the user via the 3D view of FlexSim is a better solution for this research as it allows for better portability of the solution across models and projects for dissemination. Although newer versions of FlexSim have more compatibilities with Python, those were not tested for license reasons. For launching FlexSim instances, a separate file is prepared, which can be identical to the monitoring file. Yet specific scripts for managing input parameters are developed. Figure 2 shows the architecture of the system developed for this research. It follows the main common parts in the scientific community mentioned before, Physical, Virtual, Service and Data Spaces.

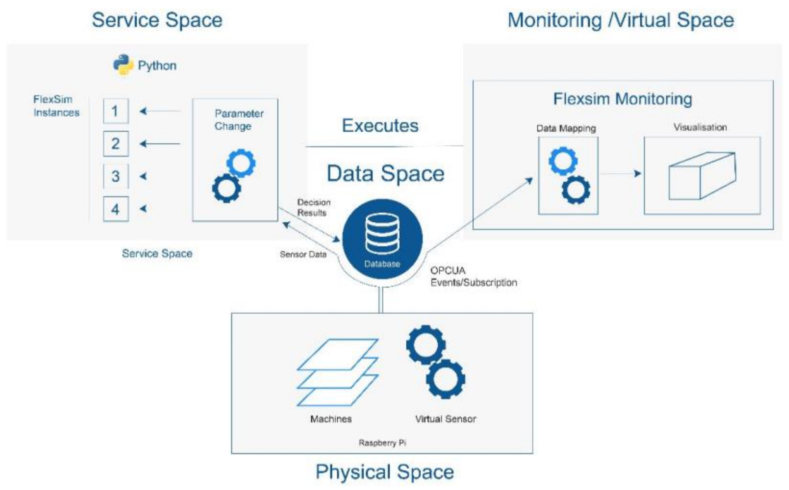


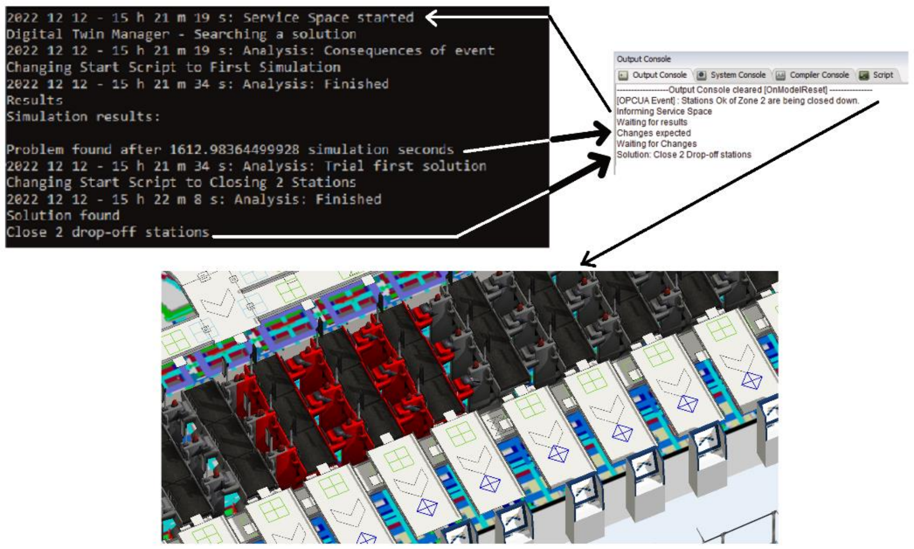
Figure 2, Architecture of the Digital Twin

4 Use Case

The use case of this research focuses on preparing a digital twin for a logistic buffer system. The logistic system consists of several modules. One module, the drop-off stations, are made for users to depose items for storing them for a specific time. The objects are then automatically stored in a second module, the buffer place, with specific priorities. Then when the users come back and request their items, it is delivered back in a delivery station. This system manages users and objects; a single event could lead to challenging consequences when they are not foreseen beforehand. For instance, a blocked station should be avoided for storing objects, and users' input should be reduced accordingly to not overload the system.
Following the methodology mentioned in 3, this research uses a Raspberry Pi running a Python OPC UA server with a fitting data model. This system is used as a parallel early development phase of product development as the system is being built. Various modules of the logistic system are modelled in the OPCUA Server, e.g. the drop-off stations, the buffers for delivery, and the delivery stations.
The Virtual Space consists of the FlexSim environment with the PLC emulation module getting and setting data via OPC UA protocol. Each node of the Raspberry Pi OPC UA Server has an equivalent in the FlexSim model, which subscribed to data changes. The process flow handles the Digital Twin aspects of the Virtual Space by controlling the variable change coming from OPC UA events, informing the users, monitoring the system and launching the Service Space on specific events.
The Service Space consists of a python program that runs instances of FlexSim models, stores the simulation results with the database and handles the results from the simulation to adapt the input parameters of the next simulation if one is necessary. The FlexSim models being simulated have a separate process flow for trying various solutions. Results from the Service Space are stored in the database and communicated to the monitoring system for user information and confirmation before being transferred to the Physical Space for controlling purposes.
The Data Space consists of a database holding simulation data and Service Space decisions.

5 Results and outlook

For testing purposes, an OPC UA client writes a new value for the variable of an OPC UA object in the Raspberry Pi OPC UA server for closing a delivery zone in the simulated real system. This simulates a problem in the delivery zone. FlexSim becomes instantly aware of the change via the subscription to the event and changes the monitoring view of the Virtual Space by coloring the delivery zone being closed down. Then a request for running the Service Space python program is made by the Virtual Space, using FlexSim runprogram() function, with the actual conditions. The Service Space launches several instances of FlexSim models sequentially and adapts the starting conditions by loading the data from the database. Several possibilities for adapting the system are preprogrammed, and the Service Space is looping through those possibilities, starting from doing nothing to the most critical state of the system. If the first simulation returns an event of the system being blocked, the Service Space launches another simulation by iterating to the next possible measure to save the system. Once the simulation runs for a predefined time, in the research case, one hour, the result is considered viable and the solution accepted. If the Service Space reaches a positive impact, it communicates it to the monitoring view for confirmation and stores its results in the database. The logs of the Virtual and Service Spaces are depicted in Figure 3.


```
2022 12 12 - 15 h 21 m 19 s: Service Space started
Digital Twin Manager - Searching a solution
2022 12 12 - 15 h 21 m 19 s: Analysis: Consequences of event
Changing Start Script to First Simulation
2022 12 12 - 15 h 21 m 34 s: Analysis: Finished
Results
Simulation results:

Problem found after 1612.98364499928 simulation seconds
2022 12 12 - 15 h 21 m 34 s: Analysis: Trial first solution
Changing Start Script to Closing 2 Stations
2022 12 12 - 15 h 22 m 8 s: Analysis: Finished
Solution found
Close 2 drop-off stations
```

```
-----------------Output Console cleared [OnModelReset] -------------
[OPCUA Event] : Stations Ok of Zone 2 are being closed down.
Informing Service Space
Waiting for results
Changes expected
Waiting for Changes
Solution: Close 2 Drop-off stations
```

Figure 3, Service Space with its log (upper left), the Virtual Space logs (upper right) and monitoring (bottom) with representation of communication between them

Time response is critical for our logistic system as a fast and accurate reaction can save the system before congestion. Depending on the computer the Service Space is running on, the complexity and fidelity level of the model and the time response varies. With a high-end laptop and a detailed model, an instance could simulate one hour in 34 seconds, yet when an appropriate response to a specific event is not found in the first iteration, time multiplies each step. In this research, the same data is used for all the FlexSim instances in the Service Space, which is not accurate as a few minutes can pass without finding solutions, and the system has already accumulated objects or users. Theoretically, other problems can occur for the real systems when the Service Space is running, and it should be updated with new conditions and events.
As the system will be instantly usable on the built prototype just by changing the OPC-UA server addresses, the approach leads to significant time savings. Depending on products, the Service Space can be reused and eventually adapted, leading to more time saved as it is built model independent. Moreover, concerning development time, the approach of this research can lead to the additional resource as the data structure of the prototype OPC UA servers is prone to changes and leads to mapping adaptations within

the Monitoring Space. During this research, modelling the system using a simulation environment was the most time-consuming task as the system did not exist at all. However, for existing systems, the research's Service Space can be adapted to search for OPC-UA Servers and values, extract information and prefill a simulation model with machines encountered in the scrapping, to reduce modelling time.
Moreover, before implementation with the real system, this research will focus on optimizing the time response by (1) updating each instance to the real-time data available, (2) integrating the multi-fidelity approach of the author's previous research within the Service Space to adapt the fidelity model of the simulation according to the threat level of the event and hence accelerating the delivery of results. Once implemented, the system will be connected to the prototype when built, and further analysis of the threat sensed by the Digital Twin and the real threats will be compared.

Acknowledgements

This work has been supported and funded by the Austrian Research Promotion Agency (FFG) via the “Austrian Competence Center for Digital Production” (CDP) under contract number 881843.

References

1. W. Kritzinger, M. Karner, G. H. J. Traar and W. Sihn, "Digital Twin in manufacturing: A categorical literature review and classification," IFAC-PapersOn Line, 2018.
2. M. W. Grieves, "Product lifecycle management: the new paradigm for enterprises," International Journal of Product Development, Vols. 2, Nos 1/2, pp. 71-84, 2005.
3. M. Grieves, "Origins of the Digital Twin Concept," 2016, 10.13140/RG.2.2.26367.61609.
4. Lo C.K., Chen C.H., Ray Y. Zhong. A review of Digital Twin in product design and development. Advanced Engineering Informatics. 2021
5. Lattanzi L., Raffaeli R., Peruzzini M., Pellicciari M..Digital Twin for smart manufacturing: a review of concepts towards practical industrial implementation. International Journal of Computer Integrated Manufacturing, 2021; 34(3):1-31. https://doi.org/10.1080/0951192X.2021.1911003
6. Luściński S., "Digital Twinning for Smart Industry," in MMS 2018 - 3rd EAI International Conference on Management of Manufacturing Systems, Dubrovnik, Croatia, European Alliance for Innovation, 2018.
7. Sakr A. H., Aboelhassan A., Yacout S. and Bassetto S., "Building Discrete-Event Simulation for Digital Twin Applications in Production Systems," 2021 26th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA), pp. 01-08, 2021.
8. Karakra A., Fontanili F., Lamine E. and Lamothe J., "A discrete event simulation-based methodology for building a digital twin of patient pathways in the hospital for near real-time monitoring and predictive simulation," Digital Twin, 2022.
9. Unified Architecture Core - UA. “OPC 10000-1 UA Part 1: Overview and Concepts”. Published on the first of November 2022.
10. Kutscher V., Olbort J., Steinhauser C., Anderl R., “Model-Based Interconnection of Digital and Physical Twins Using OPC UA”, 10.1007/978-3-030-51981-0_23, 2022

11. Protic A., Jin Z., Marian R., Romeo, K. Abd, Campbell D., Chahl J., Implementation of a Bi-Directional Digital Twin for Industry 4 Labs in Academia: A Solution Based on OPC UA. 979-983. 10.1109/IEEM45057.2020.9309953; 2020
12. Schuh G.Rebentisch E. S., Riesener M., Ipers T., Data quality program management for digital shadows of products, Procedia CIRP 86 (2019) 43–48.
13. Lo C.K., Chen C.H., Zhong Ray Y.. A review of Digital Twin in product design and development. Advanced Engineering Informatics. 2021