Industry 4.0-based Production of Precast Concrete Modules – Enabling Dynamic Scheduling Using the Digital Twin

Simon Kosse[1], Vincent Betker[2], Markus König[1], Thorsten Schmidt[2]
[1]Ruhr University Bochum, Germany
[2]TU Dresden, Germany
simon.kosse@ruhr-uni-bochum.de

Abstract. Modular construction is generally considered to be a very efficient construction method. However, disruptions in production can negate the advantages of modular construction. Dynamic scheduling (DS) is the strategic handling of production disruptions where real-time events' effects are mitigated by analyzing the state of the production system and automatically optimizing production planning. The prerequisite for DS is acquiring real-time data relevant to production planning. Thus, the digital capturing, evaluation, and provision of production-relevant data are needed. This paper develops a Digital Twin (DT) of a production system based on the Asset Administration Shell (AAS), known as the implementation of DT in Industry 4.0. The AAS of the production system serves as a virtual aggregation level for the relevant data. DS is implemented using a simulation-based scheduler. The presented approach is verified using a virtual production system for precast concrete modules as a proof of concept.

1. Introduction

Modular construction with precast concrete elements is characterized by time and cost efficiency. Precast concrete elements are produced independently of weather conditions and in technically and economically optimized sequences. Disruptions in precast concrete production are due to uncertainties related to their production processes, such as varying delivery dates, process times, or supply chain delays, and cause increased costs and time expenditures. This variability inherent in precast concrete production presents significant challenges in maintaining a fixed schedule that accounts for all relevant factors and constraints.

Continuous monitoring and simulation of the production processes can be used to respond dynamically to disruptions with an optimized production sequence. Dynamic scheduling (DS) is the strategic handling of production disruptions where real-time effects of events are mitigated by analyzing the status and automatically optimizing production planning. The prerequisite for DS is acquiring real-time data relevant to the production planning states of the production environment. Thus, the digital acquisition, evaluation, and provision of production-relevant data along the production process are needed. The availability of this data is also key to Industry 4.0 (I4.0), which refers to the digitization of production, in which actors in the production process, such as products, machines, and processes, are virtually mirrored and networked. In I4.0 production, actors can exchange data and information independently and self-controlled. The goal must therefore be to combine DS with the digital capturing of production-relevant data according to the I4.0 model to enable dynamic and self-adapting production of precast concrete modules.

This paper develops a Digital Twin (DT) of a production system based on the Asset Administration Shell (AAS) concept, known as the DT's technical implementation in I4.0. The AAS of the production system serves as a virtual aggregation level for the relevant data for DS. Although execution management is also an essential responsibility of the DT of a production system, this work focuses on aggregating data for DS. In this work, DS is implemented using a simulation-based scheduler. The DS system is connected to the AAS of the production system through a service that queries data from the AAS. Communication with the DS framework is realized via its API. The presented approach is validated using a virtual production system for precast concrete modules consisting of parallel circulation systems as a proof of concept. This paper will answer the following research questions: *How can data collected during production be used to optimize production scheduling?* and *How can a service for dynamic scheduling be integrated into an I4.0-based production system*?

2. State of the Art

Although automation approaches are available for most precast concrete module production processes, efficiency still needs to improve in the precast concrete production industry. It is essential to digitize the production process for automation to improve efficiency. Reichenbach and Kromoser (2021) note that digitization approaches are needed to interconnect production processes along the value chain. As Cheng et al. (2023) suggest, the precast concrete industry has to transition from experience-based manual production to data-driven automation. In the manufacturing industry, the digital transformation toward data-driven production is referred to as I4.0. The term is a reference to the upcoming fourth industrial revolution, which will lead to a convergence of the physical and virtual worlds by integrating advanced technologies such as Artificial Intelligence (AI) and the Internet of Things (IoT) (Monostori et al., 2016). The main goal of I4.0 is to create smart factories that are more efficient, flexible, and responsive to market demands through interconnecting machines, devices, and systems, enabling communication and interaction with each other in real-time. Production systems can react flexibly to disruptions and failures as well as changes by the customer (Kagermann et al., 2013). Coupling the prefabrication of concrete modules with modern information and communication technology associated with I4.0 can significantly improve the efficiency and productivity of the construction industry (Klinc and Turk, 2019). A major disadvantage of precast concrete construction is its inflexibility in reacting to uncertainties and risks in project scheduling, causing delays in the overall construction schedule and impacting project timelines and budgets (Li et al., 2018). Optimization and simulation approaches have generally been shown to efficiently mitigate the detrimental effects caused by schedule risks (Qi et al., 2021).

Scheduling the production of precast concrete components is a complex task that involves managing the stochastic nature of processing times during the production process. Recent studies have addressed this issue by considering parallel flow shops and uncertainties inherent in production processes (Ma et al., 2018). This area of research is now commonly referred to as prefabricated concrete component production scheduling (PCCPS), as coined by Du et al. (2021). Various approaches have been proposed for solving PCCPS problems, such as dispatching rules and genetic algorithms (Teymourifar et al., 2020). However, most research has focused on genetic algorithms. Dynamic scheduling has been introduced as a solution to mitigate process uncertainties in real-time. This concept is relevant for many manufacturing environments, as static schedules can become obsolete when an unforeseen event occurs on the shop floor (Toro, 2021). Altaf et al. (2018) exploit the possibilities of IoT for scheduling and present an integrated planning and control system for panelized construction that enables real-time monitoring and production scheduling through RFID-based data acquisition and simulation-based optimization. The proposed system uses the RANSAC data mining method and Discrete Event Simulation to simulate the production processing time at each workstation. However, the paper does not address the digital or virtual models needed to implement the framework presented. In construction, the established Building Information Modelling (BIM) method is often used as a digital basis for linking and integrating other technologies (Wang et al., 2020). However, BIM lacks the semantic integration of sensor and IoT data as well as real-time capability (Boje et al., 2020).

In I4.0, the DT is a key concept for capturing and providing data along the value chain (Uhlemann et al., 2017). In construction, a DT is generally understood to be a digital model going beyond a building information model through the sensor and IoT data integration and enabling data exchange between the digital model and its physical counterpart via a data link (Boje et al., 2020). BIM and the DT, however, are not mutually exclusive but complementary and respond to different industry requirements (Davila Delgado and Oyedele, 2021). When implementing DTs, the construction industry is still at an early stage. Up to now, there have mainly been isolated solutions based on different architectures and not readily compatible with each other. In previous work (Kosse et al., 2023), we have proposed a modular approach based on information containers where data is captured use case-related and exchanged via standardized interfaces enabling a flexible implementation of DTs. We introduced the AAS from I4.0 and Information Container for Linked Document Delivery (ICDD) as examples. In addition, to support the automated production of precast concrete modules, we have proposed (Kosse et al., 2022) presented an AAS-based DT for precast concrete modules containing production-relevant data stored in submodels, each representing a closed view of one aspect or use case. Standardized interfaces enable seamless data access, and the DT can be extended throughout all phases of the module's lifecycle. However, the paper is limited to the component level. The extension to the production system level and the use of data for data-based decision-making have yet to be considered. Only a few works exist on the implementation of dynamic scheduling. Villalonga et al. (2021) propose a decision-making framework based on DTs for DS of cyber-physical production systems (CPPS). The framework includes a DT of the CPPS with a decision-making module that generates scheduling decisions based on real-time data from the DT and a control module that executes the scheduling decisions in the production system. While their approach cannot easily be applied to the scheduling of precast production, it offers a sound foundation.

3. Conceptual Framework

The following section presents a conceptual framework for optimizing precast concrete production by implementing a dynamic scheduling system based on the DT concept. The framework is designed around a service-oriented architecture, which utilizes a central service to facilitate data exchange between the DT and the optimization application (cp. Fig. 1). The aim is to enhance production scheduling for greater efficiency. The structure of the DT is described first, followed by a detailed explanation of the central service's role in coordinating data exchange and optimization sequences. Finally, the dynamic scheduling approach for precast concrete modules is specified.

Production systems comprise numerous complex components, including products, machines, and processes. In the I4.0 model, each component has a corresponding DT that collects and stores data and information about the asset. The locally captured production-relevant data must be aggregated into a hierarchical DT, allowing a holistic view of precast concrete module production. Production-relevant data is collected by DTs of products, machines, and processes at the lowest hierarchical level. At the highest level of the hierarchy, the DT of the production system offers an overview of the entire production process. It collects data from the individual DTs of the products, machines, and processes. It integrates them to create a complete picture of the production system enabling real-time monitoring and control of the production process and the ability to make informed decisions based on the overall system performance.

DTs are used to manage data and typically do not include built-in data analysis and evaluation capabilities. These capabilities must be implemented through external applications that interact and communicate with DTs for data and information exchange. Data exchange must be coordinated, and the optimization sequences must be controlled to enable dynamic scheduling. A service connects the DT of the production system and the optimization framework, querying the relevant data and converting it into the respective input format for the scheduling application. As the optimization process proceeds, the service provides feedback on the results of the optimization process to the DT of the production system. This feedback enables the production system to improve its schedule based on the optimization results continuously.

The dynamic scheduling application is a crucial element of the presented approach as it generates information based on the input given by the service. The service application triggers each optimization sequence. New production schedules can be created in real-time based on changing conditions, such as the addition of new orders or unexpected delays resulting in improved efficiency, reduced wait times, and increased utilization of resources.

DTs and external services for DS in precast concrete module production have several benefits. For example, it allows for real-time monitoring and analysis of production data, which can help identify inefficiencies and bottlenecks in the production process. This data can also be used to optimize the production process, enabling companies to increase efficiency, reduce costs, and improve quality.

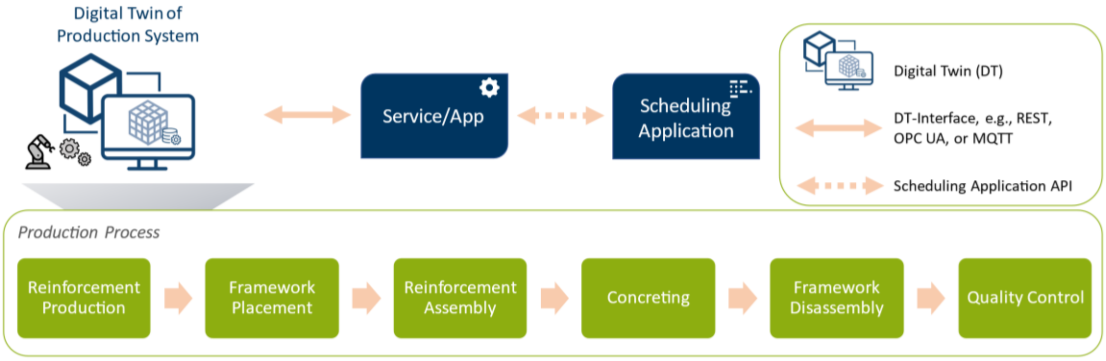

Figure 1: Schematic illustration of the concept for linking a scheduling application to a production system based on the I4.0 model to enable dynamic scheduling.

4. Implementation

This section presents the implementation of the conceptual framework. First, the AAS of I4.0 is presented as the DT's implementation (Plattform Industrie 4.0, 2020). Second, based on Sim4FJS, a dynamic scheduling application is implemented to optimize the production schedule (Schwemmer et al., 2020). Finally, a Python-based HTTP/REST web service is implemented to coordinate data exchange between the DT and the DS application.

4.1 DT of the production system

The AAS is a modular information container consisting of various submodels representing different aspects or use cases. Together, the submodels form a DT by capturing all the relevant data and information for the corresponding use case. This information can be stored as literal values or references to external data sources. Each data point has a semantic ID and a concept description to ensure machine readability and unambiguous interpretability. The AAS provides data access interfaces, including HTTP/REST, OPC UA, or MQTT. In addition, a package format containing the referenced data and a serialization of the data model is available. The AAS can be hosted on a server as an actively communicating or passively reacting component.

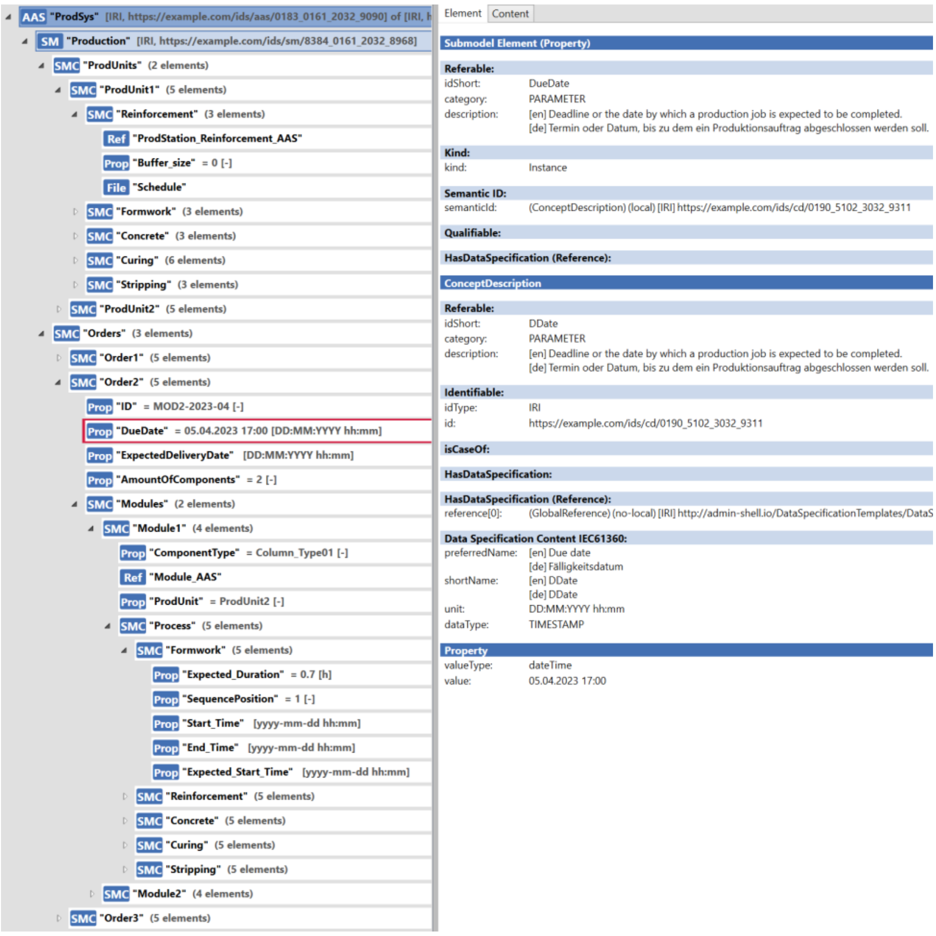

Figure 2: AAS of the production system: Submodel *Production*.

The AAS of the production system consists of data and information on the production units and orders in the production system (cp. Fig. 2). A production unit corresponds to a circulation system, which consists of their respective production stations, such as a station for producing reinforcement. In the AAS of the production system, a reference to the corresponding AAS is stored for each production station, and the capacity and utilization of the storage in the feed of the production station are recorded. Production jobs are stored with a unique ID, due date, and modules included in the production job. Each module is classified into a module type, for which the production quantity, parameters, and assigned production unit are recorded. In addition, references to the corresponding module AASs are stored. Production parameters are recorded separately for each production station. They include the expected start time, duration of the production step, and actual start and end time. For each property, a concept description is included consisting of a semantic ID and a data specification according to IEC61360. The I4.0 middleware, Eclipse BaSyx, hosts the AAS on a server. Eclipse BaSyx is an open-source software platform for implementing I4.0 solutions providing tools and libraries for developing and integrating smart manufacturing systems that can connect to and interact with other systems and devices.

4.2 Dynamic Scheduling Framework

A production system made of parallel circulation systems can be classified as a parallel flow shop. However, since the heat treatment process takes significantly longer than other processes, parallel heat treatment stations are needed to achieve even utilization rates of the stations. This feature makes the circulation system a flexible flow shop (FJS). The simulation-based scheduler is implemented using Sim4FJS, an open-source Python framework for simulating flexible job shops built upon the discrete-event simulation framework *Simpy*. Since flow shops are a special case of job shops, Sim4FJS is well suited to model the production of precast concrete modules. However, circulation systems critically differ from job shops, consisting of chained stations with limited or no buffer between them. Since product carriers can only move to the next station after the previous carrier leaves the previous station, the simulation model is modified to consider these interconnections. This is achieved by implementing a strict linkage of the stations, e.g., a product can only leave a production station when its next station becomes available. In this configuration, buffers can be added by introducing process stations with a processing time of zero. This is performed by the service presented in section 4.3. The optimization goal of the scheduler is to minimize the *total order tardiness* metric, assuming that the tardiness cost is the same for all orders. A genetic algorithm (GA) provided by the *geneticalgorithm* framework is used to find the schedule minimizing that metric. The GA alters the sequence of jobs entering the production system, which is then handed to the simulation model. The model then simulates the resulting production processes, from which the total order tardiness is calculated and fed back to the GA, which adapts the given start sequence according to the previous input sequences and respective results. To enable the dynamic generation of simulation models, an HTTP/REST API accepts system state and order backlog information in YAML syntax to create a simulation model and optimize the schedule dynamically. The simulation model is generated from this input to compute an optimized production schedule. The API returns the expected order completion dates, the expected start times, and chosen stations for every process step according to the optimized schedule. The simulation-based scheduler is called Sim4BFT and can be hosted on a server to be accessed remotely over its API.

4.3 Service

A Python-based web service works as a mediator between the AAS, which stores relevant data about the production system, and the dynamic scheduling optimization framework, which optimizes the system's operations in real-time, to optimize the production system's operations efficiently. The service includes three steps: data querying, data conversion and input to the optimization framework, and conversion and transmission of optimized schedules to the DT (cp. Fig. 3). First, the web service queries the AAS using an HTTP/REST interface to retrieve the submodel elements *ProdUnits* and *Orders* as JSON files. Once the data is retrieved, the web service preprocesses and converts it into the correct input format for the optimization framework. The conversion involves modifying certain data types and structures, e.g., time and duration data, due dates, and process times are converted into unified simulation time units. Also, the sequence of production steps is extracted from the Orders submodel element collection and specified in the appropriate format for simulation using process stations and available buffers as part of the preprocessing. The input data is stored in the YAML syntax and transmitted via the optimization framework's API. Once the input data is transmitted, the optimization run is triggered with another API call. After completion of the optimization process, the optimized schedule is queried by the service and converted back into the appropriate format for the AAS. Finally, the production schedule is transmitted to the AAS via its HTTP/REST interface. The schedule data from the production system AAS is distributed over the actors in the production system, such as products, machines, and processes, which are all represented through AASs. However, this step is not the focus of this work and will be discussed in detail in future research.

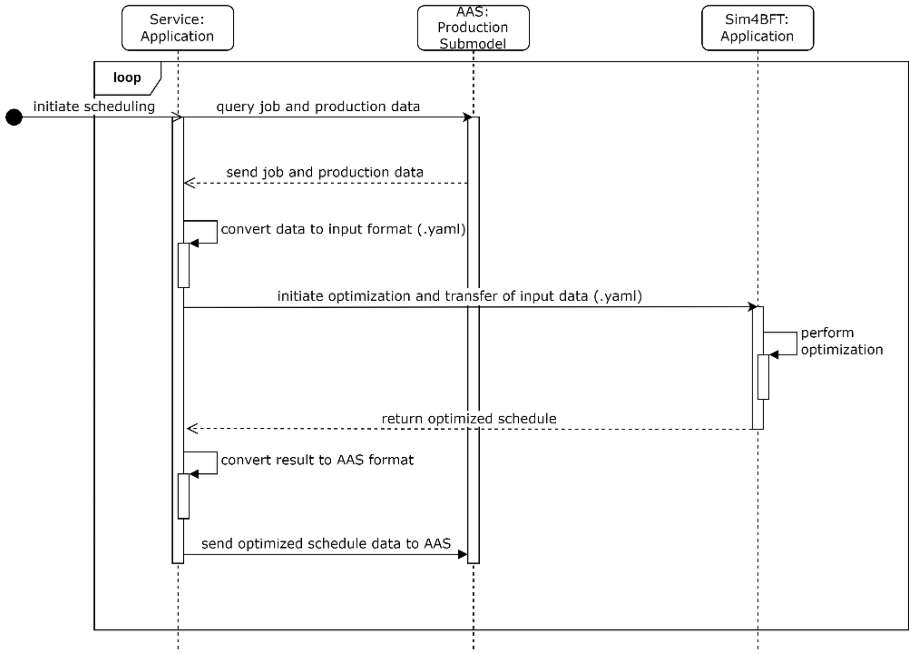

Figure 3: Sequence diagram of data exchange between AAS and optimization framework by a service.

5. Verification

In this section, we validate the proposed framework using a virtual production system consisting of two parallel circulation systems, each consisting of one workstation for formwork, reinforcement, concreting, and stripping, while having two stations for curing. The objective is to produce ten concrete modules distributed over three orders. One challenge in simulating the production flow of an automated production system for precast concrete modules is the need for production data that is the basis for the presented framework. In this regard, production data for the conventional production of precast concrete modules are used, as presented in Table 1 (Du et al., 2021), to verify our approach. Although this data does not represent production systems modeled on I4.0, they are suitable for verifying our framework. Figure 4 illustrates the optimized production flow visualized in a Gantt chart produced by Sim4BFT after a scheduling request. It shows that the scheduler has correctly modeled all modules and processes.

Aggregating relevant data from the production system enables real-time monitoring and optimization of production activities, allowing potential bottlenecks and delays to be detected early. The DT approach facilitates integrating and utilizing current production data from sensors and IoT devices. Moreover, applying AI techniques to predict future production states is possible and will be a focus of future research. DS allows for creating flexible production schedules that minimize the risks of delays and downtime. These schedules are adjusted in real-time as production conditions change, ensuring that production stays on schedule and is optimized for maximum efficiency. This approach's scalability depends on the specific needs of the production process and can be adjusted accordingly. However, the system's scalability can be expanded or reduced by adding or removing production units and jobs and adjusting the amount of data being collected.

Table 1: Processing times (in hours) of precast concrete modules according to Du et al. (2021).

Part Type	Formwork	Reinforcement	Concrete	Curing	Stripping
1	0.5	1	0.4	7	0.5
2	2	2	0.5	7	1
3	1.5	1.5	0.5	7	1
4	1.5	2	0.5	7	1
5	0.5	1	0.2	7	0.5
6	0.7	2	0.5	7	0.6
7	2	2.5	0.5	7	1.5
8	0.5	1	0.3	7	0.4
9	3.5	3	1	7	1
10	1.5	1.5	0.5	7	1

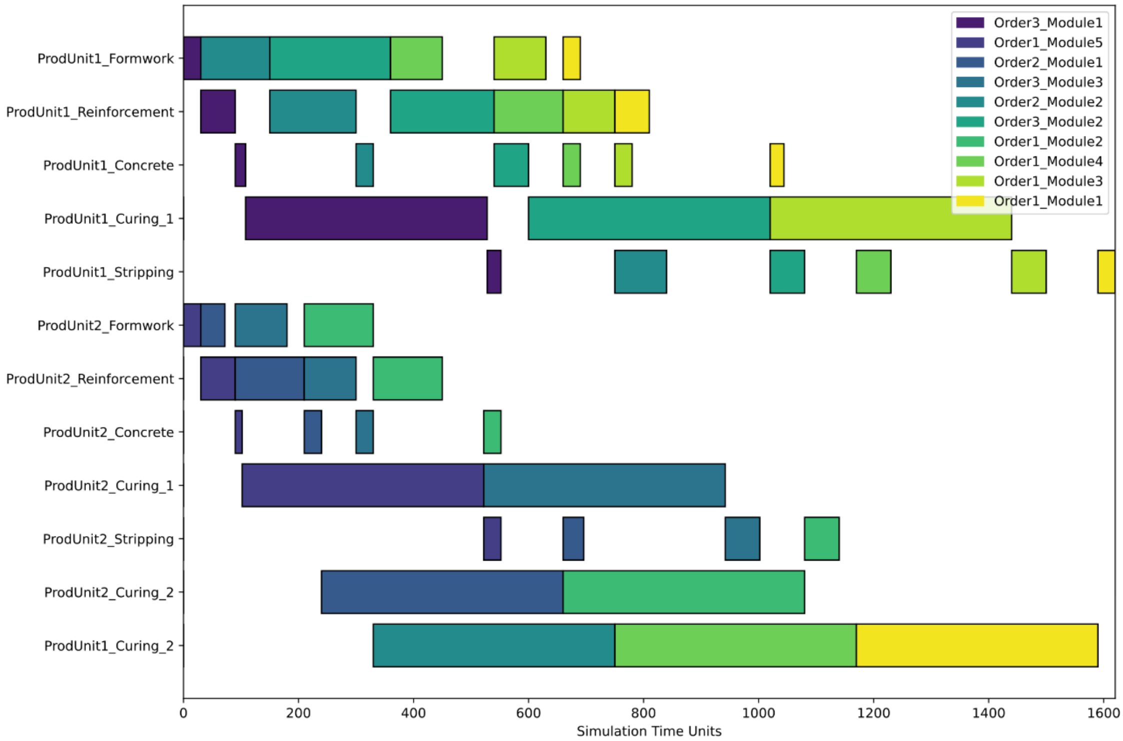

Figure 4: Gantt chart of the optimized production schedule.

6. Conclusion

This paper presents an approach to integrating dynamic scheduling into an I4.0-based production system. The proposed method employs a DT to aggregate data collected at the product and machine levels. This data is available to an optimization framework through a service, enabling dynamic production process scheduling. To demonstrate this concept, we use the AAS as the DT's technical implementation and the Sim4BFT optimization framework for dynamic scheduling. The key takeaways from this study are:

- Aggregating product, machine, and process data into a hierarchical DT structure of the production system is essential to achieve optimized dynamic scheduling of production processes for precast concrete modules.
- The AAS allows for modular implementation of production systems' DT providing standardized interfaces for the data exchange with a service.
- To make data stored in DT usable, it's necessary for external applications to have access to it. A service-oriented architecture enables a coordinated data exchange while allowing for a flexible and extensible implementation.

While this paper focuses on the interaction and communication between the production system's DT and the optimization framework via a service, future research will explore the communication and interaction with the DT at the product and machine levels providing a more comprehensive understanding of how data can be collected, aggregated, and utilized to optimize production processes.
The main limitation of this paper is based on the lack of production data and the resulting inability to demonstrate the effectiveness of the proposed approach in real-world scenarios. Since the approach has not been tested on real-life production systems and data, it is difficult to assess its efficacy.

Acknowledgments

The authors thank the German Research Foundation (DFG) for the financial support of the projects with grant numbers 423963709 and 423928620 within the priority program 2187, "Adaptive modularized construction made in a flux".

Data availability

We acknowledge the importance of data availability and transparency in scientific research. The code of the web service application developed within this research work is available at our public repository (https://tlscm.mw.tu-dresden.de/scm/repo/git/Sim4BFT).

References

Altaf, M. S., Bouferguene, A., Liu, H., Al-Hussein, M. and Yu, H. (2018) 'Integrated production planning and control system for a panelized home prefabrication facility using simulation and RFID', *Automation in Construction*, vol. 85, pp. 369–383.

Boje, C., Guerriero, A., Kubicki, S. and Rezgui, Y. (2020) 'Towards a semantic Construction Digital Twin: Directions for future research', *Automation in Construction*, vol. 114, p. 103179.

Cheng, Z., Tang, S., Liu, H. and Lei, Z. (2023) 'Digital Technologies in Offsite and Prefabricated Construction: Theories and Applications', *Buildings*, vol. 13, no. 1, p. 163.

Davila Delgado, J. M. and Oyedele, L. (2021) 'Digital Twins for the built environment: learning from conceptual and process models in manufacturing', *Advanced Engineering Informatics*, vol. 49, p. 101332.

Du, J., Dong, P., Sugumaran, V. and Castro-Lacouture, D. (2021) 'Dynamic decision support framework for production scheduling using a combined genetic algorithm and multiagent model', *Expert Systems*, vol. 38, no. 1.

Kagermann, H., Wahlster, W. and Helbig, J. (2013) *Recommendations for Implementing the Strategic Initiative INDUSTRIE 4.0 - Securing the Future of German Manufacturing Industry,* acatech - National Academy of Science and Engineering.

Klinc, R. and Turk, Ž. (2019) 'Construction 4.0 – Digital Transformation of One of the Oldest Industries', *Economic and Business Review*, vol. 21, no. 3.

Kosse, S., Hagedorn, P. and König, M. (2023) 'Analyzing modularity in construction digital twins for developing a dynamic modular reference architecture', *Computing in Civil Engineering 2023*. Corvallis, Oregon, June 25-28, American Society of Civil Engineers.

Kosse, S., Vogt, O., Wolf, M., König, M. and Gerhard, D. (2022) 'Digital Twin Framework for Enabling Serial Construction', *Frontiers in Built Environment*, vol. 8.

Li, C. Z., Xu, X., Shen, G. Q., Fan, C., Li, X. and Hong, J. (2018) 'A model for simulating schedule risks in prefabrication housing production: A case study of six-day cycle assembly activities in Hong Kong', *Journal of Cleaner Production*, vol. 185, pp. 366–381.

Ma, Z., Yang, Z., Liu, S. and Wu, S. (2018) 'Optimized rescheduling of multiple production lines for flowshop production of reinforced precast concrete components', *Automation in Construction*, vol. 95, pp. 86–97.

Monostori, L., Kádár, B., Bauernhansl, T., Kondoh, S., Kumara, S., Reinhart, G., Sauer, O., Schuh, G., Sihn, W. and Ueda, K. (2016) 'Cyber-physical systems in manufacturing', *CIRP Annals*, vol. 65, no. 2, pp. 621–641.

Plattform Industrie 4.0 (2020) *Details of the Asset Administration Shell - Part 1: The exchange of information between partners in the value chain of Industrie 4.0*.

Qi, B., Razkenari, M., Costin, A., Kibert, C. and Fu, M. (2021) 'A systematic review of emerging technologies in industrialized construction', *Journal of Building Engineering*, vol. 39, p. 102265.

Reichenbach, S. and Kromoser, B. (2021) 'State of practice of automation in precast concrete production', *Journal of Building Engineering*, vol. 43, p. 102527.

Schwemmer, J., Völker, M. and Schmidt, T. (2020) 'A New Simulation-Based Approach to Schedule Personnel Deployment Times in Decentrally Controlled Production Systems', *Proceedings of the 37th International Symposium on Automation and Robotics in Construction (ISARC).* Kitakyushu, Japan, International Association for Automation and Robotics in Construction (IAARC), pp. 19–23.

Teymourifar, A., Ozturk, G., Ozturk, Z. K. and Bahadir, O. (2020) 'Extracting New Dispatching Rules for Multi-objective Dynamic Flexible Job Shop Scheduling with Limited Buffer Spaces', *Cognitive Computation*, vol. 12, no. 1, pp. 195–205.

Toro, C. (2021) *Implementing Industry 4. 0: The Model Factory As the Key Enabler for the Future of Manufacturing*, Cham, Springer International Publishing AG.

Uhlemann, T. H.-J., Lehmann, C. and Steinhilper, R. (2017) 'The Digital Twin: Realizing the Cyber-Physical Production System for Industry 4.0', *Procedia CIRP*, vol. 61, pp. 335–340.

Villalonga, A., Negri, E., Biscardo, G., Castano, F., Haber, R. E., Fumagalli, L. and Macchi, M. (2021) 'A decision-making framework for dynamic scheduling of cyber-physical production systems based on digital twins', *Annual Reviews in Control*, vol. 51, pp. 357–373.

Wang, M., Wang, C. C., Sepasgozar, S. and Zlatanova, S. (2020) 'A Systematic Review of Digital Technology Adoption in Off-Site Construction: Current Status and Future Direction towards Industry 4.0', *Buildings*, vol. 10, no. 11, p. 204.