Proceedings of the 2024 Winter Simulation Conference
H. Lam, E. Azar, D. Batur, S. Gao, W. Xie, S. R. Hunter, and M. D. Rossetti, eds.

BREAKING BARRIERS IN SEMICONDUCTOR SIMULATIONS: AN AUTOMATED LOW-CODE FRAMEWORK FOR MODEL-STRUCTURE SYNCHRONISATION AND LARGE-SCALE SIMULATION STUDIES

Madlene Leißau, and Christoph Laroque

Research Group Industry Analytics, University of Applied Sciences Zwickau, Zwickau, GERMANY

ABSTRACT

The paradigm shift towards Industry 4.0 and the emerging trends of Industry 5.0 present ongoing challenges in production planning and control. In response to these dynamics, discrete event-driven simulation methods are gaining prominence as an operational decision-support-tool, particularly in the semiconductor industry. This paper introduces an automated low-code framework designed to synchronize model structures across simulation tool boundaries for extensive simulation studies, using the Semiconductor Manufacturing Testbed 2020 as a test reference, and aims to serve as a helpful tool for simulation experiments. Key aspects include model structure synchronization, Design of Experiments, and the distributed execution of large-scale simulation studies.

1 INTRODUCTION

The visions of Industry 4.0 and Industry 5.0 challenge established production planning and control (PPC) approaches with new demands on flexibility and dynamics. Due to a rising competition taking place for the companies in parallel, discrete-event simulation (DES) methods and tools are of growing importance as key technologies, as highlighted by Vieira et al. (2018). Thus, the adaptation of existing production plans during operation is often difficult, costly and still must face operational uncertainties. DES models allow the consideration and evaluation of different scenarios, e.g. to validate concrete production plans and/or product mixes under the assumption of defined conditions in a risk-free environment without the need for excessive financial or time resources (VDI 3633 2014, Reif et al. 2023). In addition, DES models can be used to conduct extensive simulation experiments and allow the generation of massive synthetic data for the prediction and evaluation of a test space, limited only by the number of considered factors of the simulation experiment. DES models and their simulation results can be used as a part of the training phase of machine learning methods (Mönch et al. 2006, cited in Lendermann et al. 2020). For this purpose, DES models are required to represent the real system with its dynamic (and stochastic) processes as good as possible, whereby the level of detail of the modelling always depends on the underlying objective.

Developing and applying DES models in simulation studies or as digital twins (DT) requires a thorough description of the production system, adequate data provision, and efficient data management. Simulation experts face high demands on time, expertise, and technical resources, driving the need for automated modelling and experiment execution (Lugaresi and Matta 2020; Schlecht et al. 2023; Reif et al. 2023).

One area in which the growing importance of DES is particularly evident is semiconductor manufacturing, where production systems and processes have an above-average degree of complexity compared to other industries and will become even more complex in the future (Bureau et al. 2006; Mönch et al. 2011; Mönch et al. 2013). Developments toward increased product diversity, smaller batch sizes, and greater automation complicate PPC. Factors such as limited equipment capacity, stochastic processing times, and dynamic constraints add to the complexity (Lendermann et al. 2020). DES models help predict and evaluate production scenarios, aiming to reduce cycle times, meet quality standards, minimize inventory, and ensure that agreed-upon delivery dates are met.

This paper presents an automated low-code framework for synchronising model structures across tool boundaries and for the execution of large-scale simulation studies, exemplary outlined with the

Semiconductor Manufacturing Testbed 2020, short SMT2020, in the form of the low-volume/high-mix simulation model (Kopp et al. 2020a). Besides a short presentation of related work, the focus will be on the developed approach, pointing out the synchronisation of different conceptual models, the translation into executable code snippets as well as the process of defining and performing large-scale simulation experiments. On the one hand, these sub-aspects address the weaknesses of existing simulation tools such as AutoSched AP in connection with licensing and the associated massive complication of distributed simulation. On the other hand, we want to focus on the user and provide a user-centric framework. Our experience shows that users sometimes find it difficult to switch to another simulation tool (in this case AnyLogic) when their requirements change to meet new goals, such as distributed simulation in this case. This is especially the situation when users have no or insufficient knowledge of programming languages. The goal of this work is therefore to overcome these challenges by transferring well-known (data-driven) modelling concepts from AutoSched AP to AnyLogic, while at the same time opening new possibilities. The implementation using the low-code development platform KNIME, and the provision of an intuitive user interface complete the presented framework.

Following a description of the test case and the presentation of selected results, we conclude with a summary of key findings and an outlook on future work within the research group.

2 RELATED WORK

This chapter provides a brief overview of related work that deals with the research area of model generation in the sense of automated modelling and simulation, and with the research area of data farming for the evaluation of large-scale simulation experiments, especially in the context of the of generation of synthetic data for machine learning algorithms.

2.1 Model Generation

In general, the creation of DES models and their execution can be divided into several sub-problems. In addition to issues of data availability, data acquisition and data management, there are numerous tasks related to the creation, validation and deployment of simulation models and their use to answer specific questions given an objective function and corresponding constraints (VDI 3633 2014).

Accurate simulation models play a critical role in optimising processes and facilitating informed decisions within production scenarios. Model generation in the sense of automated modelling and simulation is an important area of research, especially in the context of production systems, where different methods and approaches are being explored (Schlecht et al. 2023). Based on the literature, Lugaresi and Matta (2020) divide the process of automated modelling into several phases, such as data acquisition and management, analysis and identification of material flows, statistical analysis and identification of influencing factors, identification of priority and control rules, translation of the process model into executable code, and validation of the generated simulation models. Schlecht et al. (2023) also mention model synchronisation, adaptation and application in simulation studies and experiments and refer to the concepts of generic, data-driven, and dynamic data-driven simulation described by Haouzi et al (2013). According to Haouzi et al. (2013), generic (template-based) simulation (GTS) uses standardized building blocks to create and adapt applications, which facilitates model reuse. In contrast, in data-driven simulation (DDS), the model is automatically created and updated using the available in-house data. Dynamic data-driven simulation (DDDS) represents a new paradigm in which the simulation system is continuously influenced by real-time data for improved analysis and prediction.

There are some interesting contributions and approaches in the literature using different sets of methods and techniques: Lugaresi and Matta (2021) and Zhu et al. (2023) include approaches for automated (or dynamic) generation of simulation models using process mining methods and techniques. In contrast, Goodall et al. (2018) describe a data-driven modelling approach consisting of an adaptive algorithm for generic modelling and an information model for structuring the underlying data. Such a data-driven approach has already been developed by Wang et al. (2011), with the aim of automatically generating a

model of a production system and adapting the model according to dynamic requirements and real-time information. In the context of data-driven approaches, Johansson et al. (2003) and Vieira et al. (2018), among others, highlight the need for automated data processing systems to reduce manual work and minimise associated sources of error. Additionally, the use of reusable model building blocks (or sequences, see Reif et al. (2023)) can help to reduce the time required to create simulation models. Finally, the models can be adapted and updated more quickly and with more flexibility if process structures or operations change (Vieira et al. 2018).

The semiconductor industry has also seen some interesting contributions and approaches to the development of simulation models. For example, Tulis et al. (1990) present a DES model for a semiconductor fab to study general characteristics, especially with respect to limited equipment capacities and stochastic events (e.g. equipment failures), as well as prioritisation and control strategies in queues. Later, Kim et al. (2009) and Rank et al. (2015), among others, deal with fundamental concepts related to generic simulation models. Unfortunately, these works are limited to solving a specific problem related to the presented model. In contrast, Sadeghi et al. (2016) and Khemiri et al. (2021) discuss data-driven simulation models for semiconductor fabs and evaluate different scenarios. The model by Sadeghi et al. (2016), developed from real fab data, can import external data but has limitations such as excluding equipment failures, maintenance, and material transport. At this point, the model by Khemiri et al. (2021) addresses these issues by incorporating these elements into a more modern fab simulation. In addition, the authors discuss challenges related to data adequacy and semantic integration in the development and deployment of such models.

2.2 Data Farming

The concept of data farming emerged from military simulation challenges to improve decision making in dynamic environments. Horne and Schwierz (2016) define it as an interdisciplinary method that combines simulation, high-performance computing, and statistical analysis. Sanchez (2021) notes that data farming studies aim to systematically generate and analyze large amounts of simulation data to gain deeper insights into complex systems and to access the most comprehensive and qualitatively sufficient data possible. Consequently, data farming studies enables the investigation of uncertain events with many possible outcomes and provides the ability to conduct enough experiments so that both common and unexpected outcomes can be captured, analyzed, and used to learn from them (Horne and Schwierz 2016).

With the background of data farming being characterized by military and combat simulations, this is also reflected in the related literature. Accordingly, existing models and case studies are often categorized in this context, see for example Horne and Schwierz (2008) and Horne and Meyer (2010).

In the semiconductor industry, the contribution of Pappert et al. (2017) describes an approach to estimating utilization targets that uses data farming to create a comprehensive dataset for rapid capacity planning. This tool uses regression analysis to interpolate missing data, enabling fast and accurate estimates of equipment utilization under different production scenarios. A later contribution from these authors (Pappert and Rose 2022) describes how data farming and neural networks streamline decision support by pre-generating data points from simulations that train a neural network to provide immediate responses. This approach addresses the challenge of time-consuming simulations that are often required when product mixes and fab conditions change. As the example shows, data farming, i.e. the generation and evaluation of large-scale simulation experiments, can help to improve the quality of forecasts and make planning processes more efficient on the basis of a more robust and comprehensive database.

Despite the fact that the presented works all represent promising applications in the context of data farming, Król et al. (2013), Sanchez et al. (2021) and Pappert et al. (2023) point to an increasing demand for computing resources and time and see this in the context of an growing complexity of the decision and planning problems to be considered and thus the size and depth of detail of the simulation models. The authors address this challenge and present approaches for large-scale data farming studies on distributed infrastructures. The aim is to support and significantly accelerate the execution of experiments.

3 SOLUTION APPROACH

Below, the results of the above-mentioned research in Section 2 are considered in relation to the challenges described and an approach is proposed that aims to break down existing barriers between different simulation tools and the execution of large-scale simulation studies. For this purpose, we propose an automated approach inspired by the widespread use of commercial simulation tools and their advantage of not (necessarily) requiring familiarity with the underlying programming language. Accordingly, Lugaresi and Matta (2020) point out that the graphical user interfaces of existing simulation tools already make a significant contribution to reducing the programming effort. For very large models and/or complex model structures, modelling in the simulation tools can nevertheless be very time consuming.

3.1 Concept

The development of DES models and their subsequent investigation and application in simulation studies requires a lot of time, (expert) knowledge and technical resources. In practice, depending on the choice and/or preferences developed over time regarding the simulation tools used, limitations quickly arise, whether in terms of the entry barrier for new employees, insufficient traceability of the underlying calculations during the simulation and/or limited licence functions (esp. for distributed execution of simulation experiments). Beyond these tool-specific challenges, large DES models require an adequate (and accessible) data management process via databases, adding to the overall complexity.

Based on the cooperation with industrial partners and own experiences, the considerations in this paper refer to the synchronisation of existing model structures between the simulation tools AutoSched AP (https://automod.de/autosched-ap/) and AnyLogic (https://www.anylogic.com/). AutoSched AP is an independent simulation tool specialised for the semiconductor industry and tailored to the Excel spreadsheet program. Modelling in AutoSched AP uses (predefined) input tables to enter data relevant to simulation and analysis. In contrast, AnyLogic is a dynamic simulation tool that supports all simulation methods, i.e. system dynamics as well as discrete event and agent-based modelling. Modelling in AnyLogic is done either by drag & drop and configuration of pre-built model blocks or by Java programming and integration of custom Java classes.

The development of a comprehensive solution concept requires a structured design and subsequent implementation of the individual sub-modules. These sub-modules should include (1) the selection and preview of a model in AutoSched AP or AnyLogic with the underlying data structure, (2) the synchronisation or translation of the selected model structure to the corresponding other simulation tool, (3) the generation of executable code snippets, (4) the detailing of the simulation study, (5) the distributed execution of configurations and (6) the (time-delayed) visualisation of results (Figure 1).

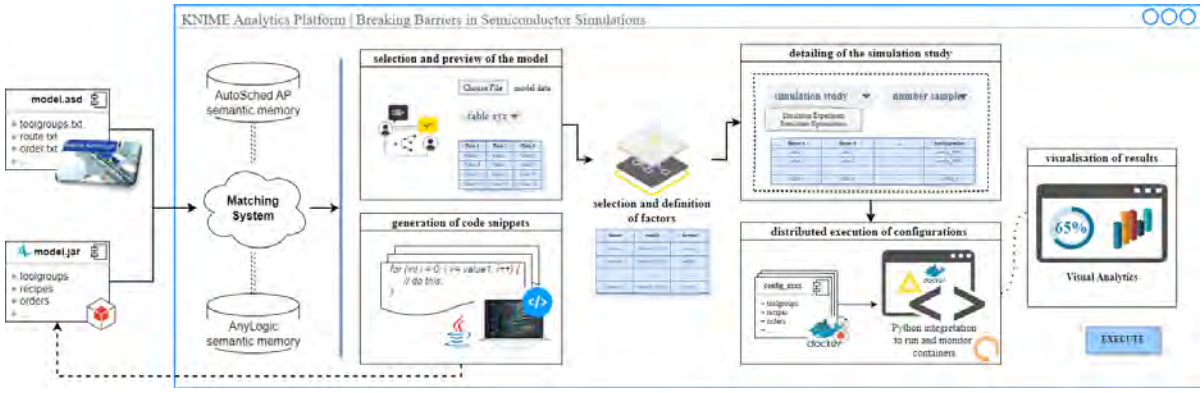

Figure 1: Overview of the solution concept.

The realization of the solution approach using the low-code development platform KNIME (https://www.knime.com/) is intended to ensure that the solution approach can be used independently of a programming language and can be adapted to individual requirements (and further tools in the future).

The implementation of the individual sub-modules in a standardised framework is intended to provide the user with a simple and intuitive user interface. As a development environment, KNIME should also enable users to make adaptations to individual use cases without in-depth programming knowledge. It is therefore important to ensure that the framework can be used independently of the operating system and the respective DES model.

3.2 Implementation

This section provides a brief overview of the implementation of each sub-module.

3.2.1 Selection and Preview of a Model

The sub-module for selecting and previewing a model in AutoSched AP or AnyLogic with the underlying data structure is intended to provide the user with a quick and uncomplicated procedure for importing a DES model for the automated synchronisation or translation of the selected model structure to the corresponding other simulation tool and more over execution of large-scale simulation experiments in the context of distributed simulation.

While an AutoSched AP model requires users to select the local .asd folder, an AnyLogic model requires users to select the exported and executable .jar file via the graphical user interface. All related files and folders are then automatically imported into the KNIME workflow resource directory. Users can preview the data structure and navigate through the available tables. This setup ensures OS independence and uses flow variables (such as strings, integers, doubles, arrays, or paths) to ensure accurate re-import of files and matching of table names with the model selection, keeping the data preview up to date.

3.2.2 Synchronisation of the Selected Model Structure

The sub-module for synchronisation or translation of the selected model structure into the corresponding other simulation tool is used to automatically match the different conceptual spaces of the underlying model structures in AutoSched AP and AnyLogic. Depending on the input, a matching system acts as an interface for synchronising or translating the model structures. The matching system is like a database (and a bit like an ontology) where both worlds are conceptually stored and related to each other.

The user interface in KNIME allows the user to switch between the model structures as required and provides a useful tool for validating the model being created by comparing the results of the models across tools based on ‘identical’ inputs.

3.2.3 Generation of Executable Code Snippets

The sub-module for the generation of executable code snippets is intended to provide the user with a useful tool regarding the translation and implementation of a certain model structure into AnyLogic. The underlying KNIME workflow processes the input data according to pre-defined specifications via a matching system and a series of character-based transformation steps, generating individual code snippets in Java that simplify modelling in AnyLogic.

3.2.4 Detailing of the Simulation Study

The sub-module for detailing the simulation study allows the user the user to select and define the factors required for the subsequent simulation study and then choose between a conventional simulation study in the form of a selectable Design of Experiments (DoE) or a simulation-based optimisation.

For a conventional simulation study, the user has several options to choose a DoE-design, such as a full factorial design or a Latin Hypercube Sampling (LHS). The DoE can be selected using the provided

selection element ('design selection') and then automatically translated into Python code and displayed next to it. This is done by accessing a stored table of defined factors, which also ensures independence from the specific model and, above all, from the model structure.

The DoE is calculated using the Python package DOEPY (https://doepy.readthedocs.io/), which is an open-source alternative to many commercial statistical software packages and focuses on making the generation of DoEs accessible in a user-friendly way (Sarkar 2019). The output is presented as a table preview and string output with information on the number of configurations required. The resulting configurations can be saved for distributed execution in a corresponding DoE folder on the disk in the experiment directory of the KNIME workflow. This is done by converting the configured factors from the original input file to the desired format and naming the configuration file according to the configuration name. This step is critical to the rest of the procedure in the following section, to ensure that input and output data is uniquely associated when running distributed using Docker (https://www.docker.com/).

3.2.5 Distributed Execution of Configurations

Some preliminary considerations are required to automate distributed execution using Docker. For example, it is essential that the input and output data of a configuration are uniquely assigned throughout the process to ensure that the simulation is initialised with the correct input values and that the output data can be backed up and assigned to the input values, thus enabling proper analysis and visualisation of the results. In addition, users should be able to independently determine the utilisation of available resources and, in addition to the number of configurations to be run in parallel, also define the percentage utilisation of processor cores and memory (RAM) for each configuration. With this specification, we were able to test and validate the process outlined in Figure 2, in accordance with the automation of the distributed execution of configurations using Docker.

The implementation and automation of the selection and writing of the required resources for Docker, the creation and start, the monitoring and reading of the individual Docker containers and their termination were successively realised in KNIME. By providing a user interface, the user has the possibility to define various positional arguments.

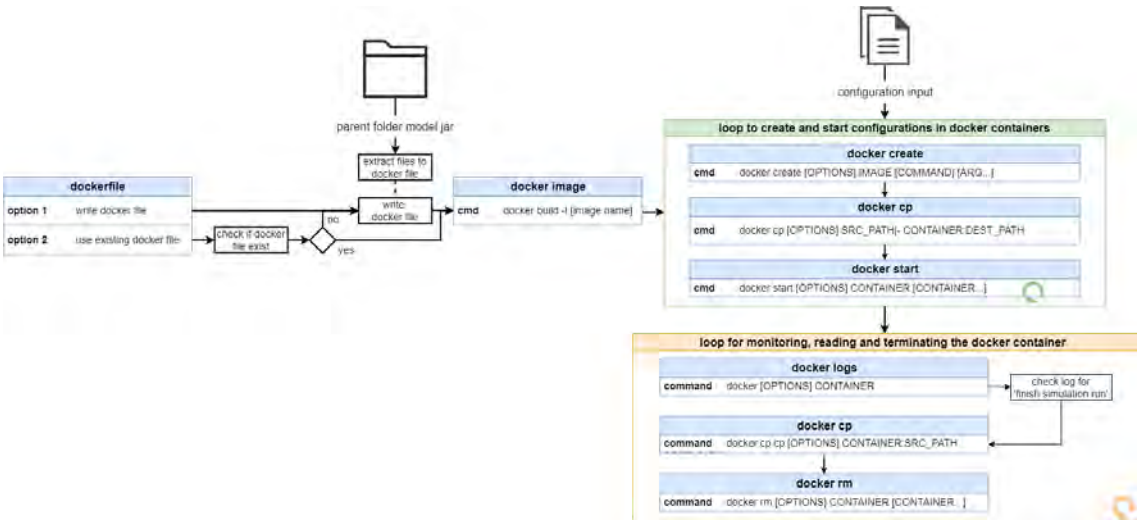

Figure 2: Using Docker to automate distributed configuration execution.

3.2.6 Visualisation of Results

The sub-module for the visualisation of results is one of the biggest challenges in terms of automation. Accordingly, and with the requirement that the proposed framework should not only be independent of the

operating system, but above all independent of the DES model, it would only be conceivable to implement a selection of frequently used visualisations in consultation with the user. This would have the advantage that the user would be directly involved and would receive a visualisation module individually adapted to his specific wishes and requirements, which he could use continuously for himself, provided that the modelling structures recur.

4 TEST CASE: SEMICONDUCTOR MANUFACTURING TESTBED 2020

In this section, we give a brief overview of the main characteristics of the selected test case in the form of the low-volume/high-mix model (Dataset 2) according to the SMT2020 by Kopp et al. (2020a), before describing the modelling process using the AnyLogic simulation tool and discussing the model generation following an initialisation at startup.

4.1 Case Description

The Semiconductor Manufacturing Testbed 2020, shortly SMT2020, according to Kopp et al. (2020a) includes four simulation models, which are supposed to represent the complexity of modern fabs in the semiconductor industry sufficiently detailed and can be regarded as an update or extension of the MIMAC simulation models. A detailed description of the models is given in Hassoun et al. (2019), Kopp et al. (2020a) and Kopp et al. (2020b). In addition, the models (and raw data) can be downloaded from https://p2schedgen.fernuni-hagen.de/.

For the implementation of the approach proposed in this paper, we choose the AutoSched AP model developed and validated by Kopp et al. (2020a) of a low-volume/high-mix scenario (Dataset 2) with more than 900 tools according to 105 tool groups distributed over 11 functional areas and 10 product types with routings and associated operations ranging from 242 operations for product type 5 to 585 operations for product type 3. The model assumes a weekly load of 10,000 wafers and the dispatching decisions are defined according to the ratio between the time remaining until the due date and the processing time required for completion, and thus correspond to the critical ratio rule.

Compared to the other three models in the testbed, the selected Dataset 2 offers due dates for the predefined schedules. This was of particular interest for future studies and influenced our decision to choose this model.

4.2 Modelling

For modelling in AnyLogic, a generic and data-driven approach has been chosen and the model is initialised with the appropriate dataset at the start of the simulation. For this purpose, we use an adapted version of the generic model structure, which was provided by the developers of SMT2020, as a result of the synchronisation of model structures between AutoSched AP and AnyLogic described in Section 3.2.2.

Basically, the modelling in AnyLogic is based on the development of 2 agents, representing a tool group and a representation of the tools contain in it. In the main agent, the tool group agent is implemented as a population. This is used for initialization at startup and ensures that each of the 105 tool groups and the delay do not have to be modelled or mapped individually. According to an infeed list and the start date shown there, production lots are injected into the model and receive a defined number of wafers via a parameter called wafer size. Furthermore, a recipe is attached to the lots using the parameter product type, in which all necessary steps are indexed and can be called up with the respective variables such as process time and sampling probability.

The selection of the tool group is then based on a search for the properties specified for the subsequent process step in the recipe and the associated step population. Finally, all processes are performed in the agent of the tool group itself, where it makes sense to perform them before or after the selection and subsequent processing at the tool level, e.g. regarding sampling, ranking, dispatching, and reworking. Similarly, all batching processes are performed at the tool group level to avoid unnecessary waiting times

at the tools and to combine suitable lots with the same product types and process steps into a batch in advance, see Figure 3. The tool groups are parameterized by initializing the main agent at startup.

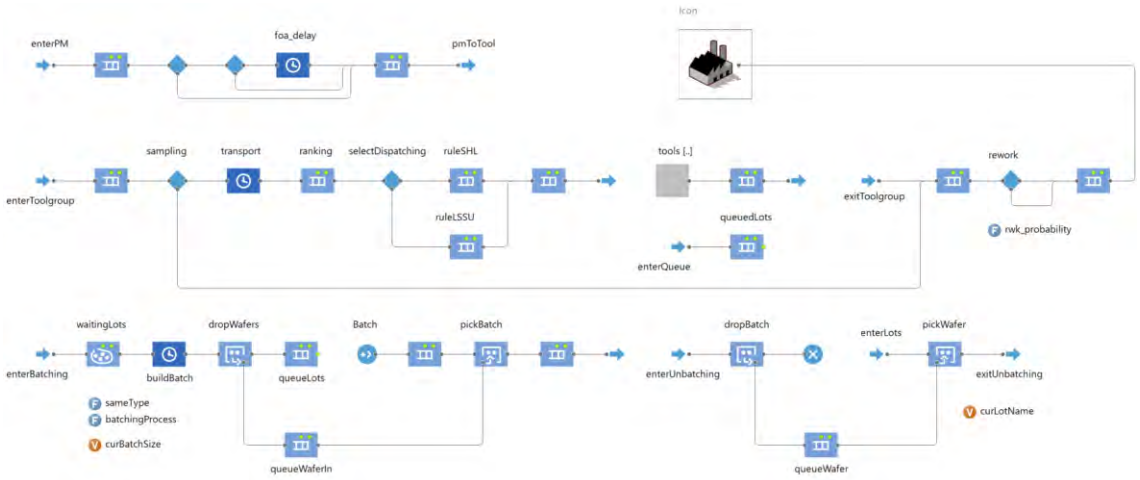

Figure 3: Tool group agent containing the tools as a population.

Once the processes that precede processing on the tool itself have been completed at tool group level, the tool is selected based on availability and, if necessary, lot-to-lens dedication or the required setup. Each tool's capacity is limited by a restricted area, which is automatically parameterized for certain tools according to a cascading setting via the initialization in the main agent at startup. While downtime can be defined directly in the implemented resource pool, preventive maintenance is modelled separately so that both time-based and counter-based maintenance events can be considered, see Figure 4.

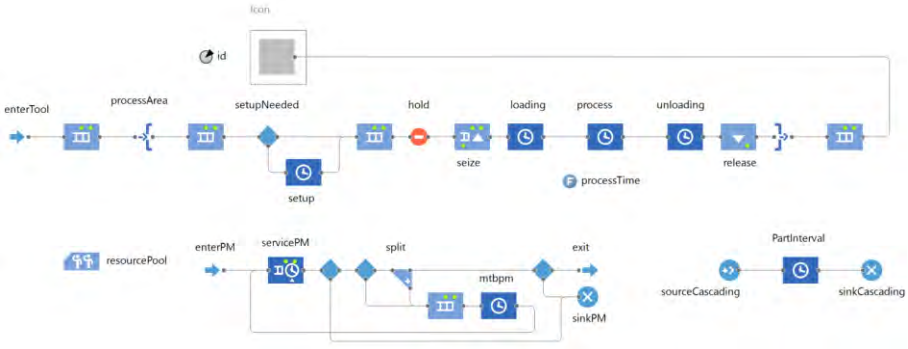

Figure 4: Tool agent for the main processing.

4.3 Validation

As this paper goes to press, the validation of the model has not yet been finally completed. However, the results of the validation phase to date indicate that the modelling is progressing successfully and that the expected objectives are likely to be achieved. Accordingly, key figures such as throughput and the ratio between breakdowns, preventive maintenance and utilisation already show comparable values to the original modelling in AutoSched AP. The remaining discrepancies must be analysed and reduced to a minimum in the further course of the validation. Once validated, the model will be made available at https://github.com/mdlnlss/smt-2020-lvhm-anylogic.

5 EXPERIMENT

Based on the solution approach as a proof of concept, the test case described in the previous section is now considered within the framework in terms of the execution of large-scale simulation studies.

5.1 Objectives and Definition of the Simulation Experiment

We set out to achieve two objectives using a particular setting: Firstly, to showcase the applicability of the solution approach, verifying that the framework can handle all necessary inputs, build designs using various DoE methods, and execute across different operating systems. Secondly, we aimed to demonstrate the effectiveness of the solution via a large-scale simulation study.

For the first goal, we rigorously tested and optimized the framework on both Linux and Windows platforms, focusing on reducing run-time issues. As a result, the primary computational demand now stems from starting the containers and storing results in .xlsx files, while other delays have been minimized.

For the second goal, we examined how tool availability in the Diffusion functional area of the simulation model affects processing, particularly through a batching process in the underlying tool groups that groups lots into batches based on a minimum and maximum number of wafers per batch – ensuring that the lots grouped together share the same product type and process step.

The analysis involves a simulation at a simulated semiconductor fab for one year, starting with initial work-in-progress (WIP). The objective is to collect data on the impact of tool availability, crucial for further analysis such as scheduling preventive maintenance to enhance tool group performance and stabilize material flow throughout the fab. Within this setup, factor constraints on the number of tools are set for each tool group in the Diffusion area, ranging from a minimum of 50 percent to the original number of tools, see Table 1. Exceptions are made for the Diffusion_FE_100 and Diffusion_FE_94 tool groups, which have high usage rates of 99% and 95%, respectively, according to the current validation phase. In this paper, the LHS design was used to implement an experimental design, performing 5,000 simulation runs. LHS provides a comprehensive statistical analysis by evenly sampling across all potential scenarios, thereby enhancing the reliability of the simulation results by ensuring thorough exploration of the parameter space.

Table 1: Definitions for the Latin Hypercube Sampling

Tool group	Minimum number of tools	Maximum number of tools
Diffusion_BE_123	5	9
Diffusion_FE_101	5	10
Diffusion_FE_120	4	8
Diffusion_FE_122	3	6
Diffusion_FE_125	3	5
Diffusion_FE_126	2	4
Diffusion_FE_127	4	8
Diffusion_FE_44	4	7

5.2 Results

All configurations were automatically started in Docker containers on a large virtual machine with Ubuntu 22.04 LTS as operating system on a dedicated simulation server of the University of the Bundeswehr Munich. An Intel Xeon Platinum 8362 with 2.80 GHz is installed as the CPU in the simulation server. The virtual machine is assigned 128 cores and a total of 2 TB RAM and 1 TB hard drive storage.

As part of the simulation study, 50 Docker containers were started in parallel and the results from the Docker container were extracted following the completed simulation run and saved them systematically in a corresponding folder structure. For 5,000 configurations and a real duration of around 12 minutes for each simulation run, this meant a total processing time of ~24 hours. By way of illustration, it would take

approximately 42 days to run the configurations sequentially – assuming they can be started immediately one after the other.

The results for all 5000 configurations and a correlation analysis between the number of tools in the tool groups and the outcome variables relating to the number of completed lots, the average cycle time and the number of lots on time (compared to the specified due date) show a high correlation between the number of completed lots and the number of lots on time (Figure 5). This simply indicates that a higher number of completed lots is often associated with a higher number of punctual lots. The correlation between the average cycle time and the number of lots completed on time is similarly unspectacular, suggesting that longer cycle times make it more difficult to complete lots on time.

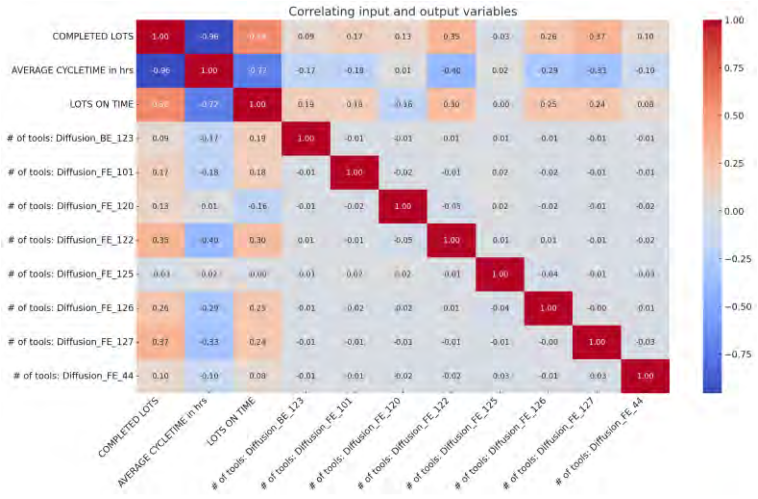

Figure 5: Correlation matrix for input and output variables.

These results show that certain tool groups correlate to varying degrees with the number of completed batches, the average cycle time and the lots on time. In particular, the tool groups Diffusion_FE_122, Diffusion_FE_126 and Diffusion_FE_127 appear to have a consistent influence across the different performance indicators.

6 CONCLUSION AND OUTLOOK

Simulation experiments take time and effort. There is little difference between whether a model must be created or already exists. Even if an existing model only needs to be updated with new parameters, the simulation itself takes time to run (Pappert/Rose 2022).

Our proposal and the implementation of a solution approach in the context of model generation and distributed simulation, and the underlying (partial) automation of these processes, can not only lead to time savings for the simulation experts themselves, but also enable trained users to start modelling simulation models and run experiments. The experiments themselves achieve a significant increase in efficiency and a lower susceptibility to error due to the elimination of manual intervention.

The modular structure of the framework allows it to be customized and easily adapted to specific requirements at any time. Accordingly, we are planning to implement further test cases in the context of a backward simulation (see Leißau and Laroque 2023). The investigations regarding model structure synchronization will also be pursued with the integration of ontologies as well as the integration of other DoE methods and optimization techniques.

ACKNOWLEDGMENTS

We want to thank Professor Oliver Rose, Falk Stefan Pappert and Daniel Seufferth for their support and the provision of a virtual machine on their simulation cluster.

This work was partially funded by the Future Mobility project. The Future Mobility project is co-funded by the European Regional Development Fund (ERDF) and by tax revenues based on the budget approved by the Saxon State Parliament.

REFERENCES

Bureau, M., S. Dauzère-Pérès, and Y. Mati. 2006. "Scheduling Challenges and Approaches in Semiconductor Manufacturing". *IFAC Proceedings 39 (3)*, 739–744.

Goodall, P., Sharpe, R., and A. West. 2018. "A data-driven simulation to support remanufacturing operations". In *Computers in Industry (105)*, 48–60.

Haouzi, H., Thomas, A., and P. Charpentier. 2013. "Toward Adaptive Modelling & Simulation for IMS: The Adaptive Capability Maturity Model and Future Challenges. In 11th IFAC Workshop on Intelligent Manufacturing Systems (IMS'2013), https://doi.org/10.3182/20130522-3-BR-4036.00104.

Horne, G. E., and K.-P. Schwierz. 2008. „Data Farming around the World Overview". In *2008 Winter Simulation Conference (WSC)*, https://doi.org/10.1109/WSC.2008.4736222.

Horne, G. E., and T. E. Meyer. 2010. "Data farming and defense application". In *MODSIM World Conference and Expo*., 13. October 2010, USA, Hampton (VA).

Horne, G. E., and K.-P. Schwierz. 2016. *Summary of Data Farming*. http://dx.doi.org/10.3390/axioms5010008.

Johansson, B., Johnsson, J., and A. Kinnander. 2003. "Information Structure to Support Discrete Event Simulation in Manufacturing Systems". In 2003 Winter Simulation Conference (WSC), https://doi.org/10.1109/WSC.2003.1261564.

Khemiri, A., Yugma, C., and S. Dauzère-Pérès. 2021. „Towards a Generic Semiconductor Manufacturing Simulation Model". In 2021 Winter Simulation Conference (WSC), https://doi.org/10.1109/WSC52266.2021.9715349.

Kim, B.-I., Jeong, S., Shin, J., Koo, J., Chae, J., and S. Lee. 2009. "A Layout- and Data-Driven Generic Simulation Model for Semiconductor Fabs". In *IEEE Transactions on Semiconductor Manufacturing 22(2)*, 225–231.

Kopp, D., M. Hassoun, A. Kalir, and L. Mönch. 2020a. "SMT2020 – A Semiconductor Manufacturing Testbed". In *IEEE Transactions on Semiconductor Manufacturing 33 (2020) 4*, 522-531.

Kopp, D., M. Hassoun, A. Kalir, and L. Mönch. 2020b. "Integrating Critical Queue Time Constraints into SMT2020 Simulation Models". In *2020 Winter Simulation Conference (WSC)*, https://doi.org/10.1109/WSC48552.2020.9383889.

Król, D., Wrzeszcz, M., Kryza, B., Dutka, L, and J. Kitowski. 2013. "Massively Scalable Platform for Data Farming Supporting Heterogeneous Infrastructure". In *4th International Conference on Cloud Computing, Grids, and Virtualization*, 144-149.

Leißau, M., and C. Laroque. 2023. "Reverse Engineering the Future – An Automated Backward Simulation Approach to on-time Production in the Semiconductor Industry". In *2023 Winter Simulation Conference (WSC)*, https://dl.acm.org/doi/10.5555/3643142.3643311.

Lendermann, P., Dauzere-Pérès, S., McGinnis, L., Mönch, L., O'Donnell, T., Seidel, G., and P. Vialletelle. 2020. "Scheduling and Simulation in wafer fabs: Competitors, Independent Players or Amplifiers?" In *2020 Winter Simulation Conference (WSC)*, https://doi.org/10.1109/WSC48552.2020.9383872.

Lugaresi, G., and A. Matta. 2020. "Generation and Tuning of Discrete Event Simulation Models for Manufacturing Applications". In *2020 Winter Simulation Conference (WSC)*, http://dx.doi.org/10.1109/WSC48552.2020.9383870.

Lugaresi, G., and A. Matta. 2021. "Discovery and digital model generation for manufacturing systems with assembly operations". In *17th International Conference on Automation Science and Engineering (CASE)*, https://doi.org/10.1109/CASE49439.2021.9551479.

Mönch, L., J. W. Fowler, S. Dauzère-Pérès, S. J. Mason, and O. Rose. 2011. "A Survey of Problems, Solution Techniques, and Future Challenges in Scheduling Semiconductor Manufacturing Operations". In *Journal of Scheduling 14 (6)*, 583-599.

Mönch, L., J. W. Fowler, and S. J. Mason. 2013. *Production Planning and Control for Semiconductor Wafer Fabrication Facilities: Modeling, Analyses, and Systems*. New York: Springer.

Pappert, F. S., Rose, O., and F. Suhrke. 2017. "Simulation based Approach to calculate Utilization Limits in Opto Semiconductor Frontends". In 2017 Winter Simulation Conference (WSC), https://doi.org/10.1109/WSC.2017.8248099.

Pappert, F. S., and O. Rose. 2022. "Using Data Farming and Machine Learning to Reduce Response Time for the User". In *2022 Winter Simulation Conference (WSC)*, https://doi.org/10.1109/WSC57314.2022.10015466.

Pappert, F. S., Seufferth, D., Stein, H., and O. Rose. 2023. "Using Kubernetes to Improve Data Farming Capabilities". In *2023 Winter Simulation Conference (WSC)*, https://dl.acm.org/doi/10.5555/3643142.3643312.

Rank, S., Hammel, C., Schmidt, T., and G. Schneider. 2015. "Reducing Simulation Model Complexity by Using an Adjustable Base Model for Path-Based Automated Material Handling Systems - A Case Study in the Semiconductor Industry". In *2015 Winter Simulation Conference (WSC)*, https://doi.org/10.1109/WSC.2015.7408393.

Reif, J., Jeleniewski, T., and A. Fay. 2023. "An Approach to Automating the Generation of Process Simulation Sequences". In IEEE 28th International Conference on Emerging Technologies and Factory Automation (ETFA), http://dx.doi.org/10.1109/ETFA54631.2023.10275718.

Sanchez, S. M., Sanchez, P. J., and H. Wan. 2021. "Work Smarter, Not Harder: A Tutorial On Designing and Conducting Simulation Experiments". In *2021 Winter Simulation Conference (WSC)*, http://dx.doi.org/10.1109/WSC52266.2021.9715422.

Sanchez, S. M. 2021. "Data Farming: The meanings and methods behind the metaphor". In *Proceedings of the Operational Research Society Simulation Workshop 2021 (SW21)*, 22nd – 26th March.

Sarkar, T.: Design of Experiment Generator in Python. 2019. https://doepy.readthedocs.io/, accessed January 28th, 2024.

Sadeghi, R., Dauzere-Pérès, S., and C. Yugma. 2016. "A Multi-Method Simulation Modelling for Semiconductor Manufacturing". In *19th World Congress of the International Federation of Automatic Control*, https://doi.org/10.1016/j.ifacol.2016.07.860.

Schlecht, M., De Guio, R., and J. Köbler. 2023. "Automated generation of simulation model in context of industry 4.0". In *International Journal of Modelling and Simulation*, https://doi.org/10.1080/02286203.2023.2206075.

Tulis, B., Mehrotra, V., and D. Zuanich. 1990. "Successful Modeling Of A Semiconductor R&D Facility". In *IEEE/SEMI International Symposium on Semiconductor Manufacturing Science*, https://doi.org/10.1109/ISMSS.1990.66131.

VDI 3633. 2014. *Simulation of systems in materials handling, logistics and production – Fundamentals*. Part 1, Düsseldorf: Beuth.

Vieira, A. A. C., Dias, L. M. S., Santos, M. Y., Pereira, G. A. B., and J. A. Oliveira. 2018. "Setting an industry 4.0 research and development agenda for simulation – a literature review". In *International Journal of Simulation Modelling*, http://dx.doi.org/10.2507/IJSIMM17(3)429.

Wang, J., Chang, Q., Xiao, G., Wang, N., and L. Shiqi. 2011. "Data driven production modelling and simulation of complex automobile general assembly plant". In *Computer in Industry*, https://doi.org/10.1016/j.compind.2011.05.004.

Zhu, L., Lugaresi, G., and A. Matta. 2023. "Automated Generation of Digital Models for Production Lines Through State Reconstruction". In *19th IEEE International Conference on Automation Science and Engineering*, https://dx.doi.org/10.2139/ssrn.4605293.

AUTHOR BIOGRAPHIES

MADLENE LEIßAU is a research assistant at the Institute of Management and Information at the University of Applied Sciences Zwickau, Germany and works in the research group Industry Analytics (www.industry-analytics.de) in the project Future Mobilty. She holds a M.Sc, in management. Her main interest is the application of simulation and machine learning methods in relation to operational decision support for production. Her email address is Madlene.Leissau@fh-zwickau.de.

CHRISTOPH LAROQUE is a full Professor of Business Analytics at the University of Applied Sciences Zwickau, Germany. His research group Industry Analytics (www.industry-analytics.de) is working on the development of manufacturing companies towards data-driven organizations. One focus is the application of simulation-based decision support techniques in manufacutring operations. His email address is Christoph.Laroque@fh-zwickau.de.