Proceedings of the 2021 Winter Simulation Conference
S. Kim, B. Feng, K. Smith, S. Masoud, Z. Zheng, C. Szabo, and M. Loper, eds.

TOWARDS A GENERIC SEMICONDUCTOR MANUFACTURING SIMULATION MODEL

Abdelhak Khemiri
Claude Yugma
Stéphane Dauzère-Pérès

Mines Saint-Etienne, Univ Clermont Auvergne
CNRS UMR 6158 LIMOS
CMP, Department of Manufacturing Sciences and Logistics
F-13541 Gardanne, FRANCE

ABSTRACT

Simulation is one of the most used approaches to analyze semiconductor manufacturing systems. However, most simulation models are designed for single-use applications and study a limited set of problems that are not reusable afterwards. Recently, in order to overcome this problem, the idea to develop a generic wafer fab simulation model has emerged. Nonetheless, few papers address the development of a generic wafer fab simulation. This paper proposes a generic, data-driven simulation model to evaluate and analyze a wide range of problems arising in modern semiconductor manufacturing systems. We discuss the issues related to the genericity of such a simulation model and the data and semantic integration issues faced by users.

1 INTRODUCTION

Semiconductor manufacturing is one of the most complex, advanced, and competitive processes. The underlying factors include multiple product types, large production processes, re-entrant process flows, a large number of expensive and sophisticated machines, and batch processing (Uzsoy et al. 1992). Over the past few years, technical changes have frequently affected semiconductor manufacturing processes. Thus, in an industrial context where models, manufacturing processes, and their interactions are more and more complex and where the level of automation is increasingly high, semiconductor manufacturers have to quickly (re-) evaluate their strategy, processes, and systems to remain competitive.

In this context, and to overcome the different challenges arising from the shop floor to the supply chain level, simulation-based approaches are widely used as they allow building a detailed model of the real system by incorporating realistic features (Negahban and Smith 2014). Recently, the idea of developing a virtual manufacturing and supply chain environment has emerged, with the promise of offering the possibility to model, investigate and improve all relevant processes on a global scale in a risk-free world. Data contained in such environments can be used for simulations where scenarios (e.g. 'What-if' scenarios) are run and whose performances are evaluated before decision-making (Fowler and Rose 2004). This virtual environment allows companies to collaborate to exchange ideas, knowledge, work together on shared problems, and minimize different types of risks. To do so, these companies must exchange data and information to develop investigative problem-solving approaches in a simulated environment.

However, the quality of decisions based on simulations is determined by the quality of the model. The process of developing an accurate, reliable, and a realistic simulation model is very costly and time-consuming and should generally be conducted by a simulation expert. Most simulation models are designed for single-use applications and study a limited set of problems that are not reusable afterward. In most cases, the components of the model, like the machines, are manually defined by the developers. Extending

978-1-6654-3311-2/21/$31.00 ©2021 IEEE

such models to large-size problems requires internal changes of the source code that is generally nearly impossible for the user. The manual procedure for constructing a simulation model of a large-scale system made up of hundreds of pieces of equipment is time-consuming and error-prone.

Besides the challenges related to the simulation model and its genericity, another challenge that arises when developing a generic data-driven simulation is related to data management and information modeling aspects. Indeed, as data are required for the automatic simulation model building, the simulation tool expects those data to be in a predefined format. Typically, this predefined format results from the first step of information modeling using conceptual modeling methods such as Entity-Relationship (ER) or Unified Modeling Language (UML) modeling. The resulting information model is then used to define the database used by the simulation tool.

To obtain the necessary data for the simulation tool, required information such as bills of materials (BOM), resource capacities, process times, and other information, have to be extracted from the enterprise requirements planning (ERP) system and manufacturing execution system (MES) with their database schema. This leads to several data integration and semantics issues that are not covered in the literature.

This paper discusses the issues related to the development of a generic, data-driven simulation model. We propose such a model able to capture a wide range of problems occurring in complex semiconductor facilities. We also address the data and semantic integration issues faced by users through the use of an ontology of the domain.

2 RELATED WORK

In the literature, few papers specifically address the development of generic simulation models for semiconductor manufacturing systems, while at the same time, simulation has been widely applied in various applications in semiconductor manufacturing. Thus, the majority of simulation models are designed for a single-use and address a particular range of problems that are not reusable. In addition, model elements, such as machines, are typically implemented manually by developers. This last point leads to some rigidities when extending or updating these models, as it implies modifying the internal source code of the model, which is most of the time not available to the user.

Based on this observation, some works propose different approaches to mitigate the complexity related to the development of a semiconductor manufacturing simulation model. In (Tullis et al. 1990), a discrete event simulation model of a fab is presented. The authors use the ManSim simulation tool to analyze capacity limitations, capacity changes, product cycle times, equipment failures, repair times, and different queuing strategies of a particular fab. Other aspects are considered, such as material transfer and equipment reliability parameters (e.g., MTBF and MTTR) and preventive maintenance programs.

In (Kim et al. 2009), the authors propose a generic simulation modeling framework to reduce the simulation model construction time. A simulation model of a 300 mm Intel HVLM plant is described in (Pillai et al. 2004). The model allows the user to perform tasks such as capacity planning, automation system design, and tactical manufacturing execution. To do this, the simulation model considers several aspects such as production tools and their associated business rules, engineering lots, AMHS, and WIP rules. Regarding the instantiation of the simulation model, it is facilitated by an automatic model generator and the automated integration of input data. Towards the same objectives, the article of Fowler et al. (2015) describes the main ingredients of semiconductor facility simulation models. Other works propose a basic generic simulation model, but they are limited to a specific problem and aspects such as handling systems (Rank et al. 2015).

More recently, a generic data-driven simulation model of a semiconductor plant that is reusable for a wide range of problems in semiconductor manufacturing has been proposed (Sadeghi et al. 2016). The objectives of building such a generic model are to develop a simulation model that is flexible in the number of resources (tool groups) and product types. The presented model combines discrete event and agent-based simulation methods and is implemented with the Anylogic Simulation software. However, this model has some limitations, as it does not take into account aspects such as batching machines, machine breakdowns, and maintenance or material transfers.

Finally, another work presented by Ocker et al. (2019), proposes an approach using semantic web technologies to automatically initialize agents in the context of the Cyber-Physical System of Systems. In their approach, the authors propose the automatic initialization of each physical agent with a pre-built knowledge base using semantic web technologies. As a feasibility study, the authors propose to test their approach through the simulation of a manufacturing facility. However, the considered manufacturing facility is inspired by the Intel Mini-Fab model, which no longer reflects the dynamics and problems arising in modern plant (Laipple et al. 2018) as transportation aspects and machine qualifications.

The originality of the work is to propose a basic generic simulation model (agent-based) that take into account several aspects of a modern semiconductor plant (e.g., material transfer, machine maintenance, tunnel constraints, etc.) that can be used and extended to address a specific problem for instance test a dispatching rule.

3 THE PROPOSED SIMULATION MODEL

To simulate a semiconductor manufacturing plant, complex interactions of different components need to be captured. For this purpose, the modeling and simulation literature proposes multiple modeling approaches. The proposed simulation model is based on the work from (Sadeghi et al. 2016), using a combination of Discrete-event and Agent-based simulation modeling. Hence, in this study, the AnyLogic tool is chosen to implement a facility simulation model of a semiconductor manufacturing plant. In the following sections, the simulation model, the agents, their roles, and responsibilities are described.

3.1 Simulation Model Components

To effectively design, and manage the complex and interdependent agents in our simulation model and to enable components reuse, the overall model is designed in a modular fashion with a separation of concern. This allows us to avoid having an “all-knowing” agent responsible for most of the information about the simulation model. Each agent addresses a separate concern regarding a high-level element of a semiconductor manufacturing plant. This approach gives more opportunities for agent upgrades, reuse, and independent development. Moreover, as the proposed model comes up with a visual animation, each agent can have a responsibility regarding elements rendering in the animation and a responsibility of managing business logic. The proposed model consists of multiple agents listed in Table 1. Each agent is defined by specifying its business process as a discrete event model and interacts with other agents, thus generating the overall system behavior.

Table 1: The different agents in the model and their responsibilities.

		Responsibility	
Agent	**Description**	**Visual Rendering**	**Business Logic**
Main	Represents the root agent	✓	×
Lot	Represents a lot	✓	×
StationGroup	Represents a set of StationFamily	✓	✓
StationFamily	Represents a set of Station	✓	✓
Station	Represents an equipment	✓	✓
TransporterFleet	Represents a fleet of transporter	×	✓
Operator	Represents an operator	✓	✓
Route	Represents a production route	×	✓
StepT	Represents a step in a Route	×	✓

The simulation model is organized hierarchically, where each agent can encapsulate other agents (see Figure 1). The model includes 9 agents in a tree-like structure, with only one parent per agent except for the Main agent.

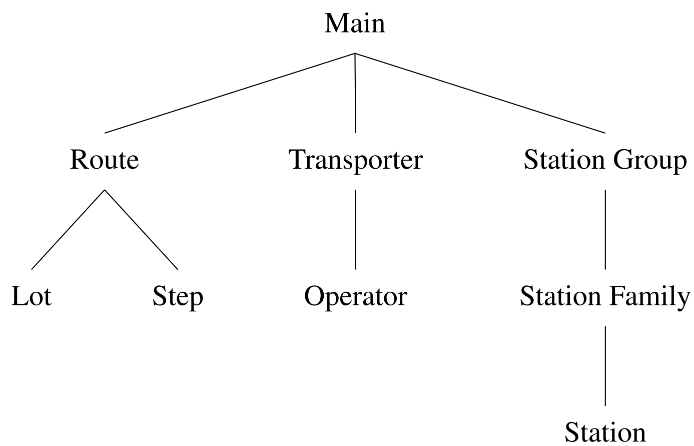

Figure 1: Agents hierarchy in the simulation model.

This allows a model to be broken down to the desired level of detail since each agent generally represents a logical and/or graphical section of the model. For instance, the transporter fleet agent can contain operator agents. To take into account the automated transporter, you only need to create a new agent, and add it to a transporter fleet agent. Table 1 shows the set of agents present in the simulation model and their responsibilities.

3.2 Data-driven Simulation Model Instantiation

A data-driven generic simulation model as "one which is designed to apply to a range of systems which have structural similarities" is defined in (Pidd 1992). Such a model is generic to a target domain and can simulate different instances of the systems in the domain without any change of the code. More precisely, a simulation model is automatically generated from an external data source using algorithms for creating the model and interfaces, which allows users to easily interact with the simulation environment without being aware of the code. Data-driven generic simulation modeling has already been identified as a promising approach for speeding up the simulation model building time, reducing the validation and verification time of the model, and in addition decreasing the need for simulation experts.

However, there is no uniform way to describe semiconductor plant data meaning and context. It leads to non-standard naming conventions, vocabularies, and challenges regarding data interpretation. Since there is no semantic definition of the data that are required by the simulation tool, it is difficult to reuse the tool in various contexts. Furthermore, even when the semantic is well defined, the complex manual development of SQL queries to retrieve the data from the real plant information system leads to a clear gap in such a data-driven simulation model.

To fill this gap, we propose to move from proprietary fab-specific data description for common non-proprietary semantic descriptions by using ontologies formalized with the Web Ontology Language (OWL) (McGuinness and Van Harmelen 2004).

OWL is a formal language for the description of terms and their relationship in a certain domain. It enables information exchange among heterogeneous applications and devices in a machine-readable format. Recently, there has been a growing interest in the development of ontologies in the semi-conductor field, leading to the development of an industrial reference platform, the Digital Reference (DR). The DR comprises a combination of different ontologies ranging from the semiconductor internal supply chains to supply chains containing semiconductors (Ehm et al. 2020).

In our simulation model, each agent is created by its direct parent. However, there is no central agent responsible for the data-driven initialization of all agents. In effect, such an approach results in a complicated agent with too many responsibilities. On the contrary, once an agent is created, it is its responsibility to retrieve the data to create and initialize its direct child, by using SPARQL to query the ontology.

Figure 2, shows the initialization of the simulation model in the form of a sequence diagram. The "Main" agent queries the knowledge base and retrieves information about the different StationGroups present in the plant. For each station group present in the Knowledge Base, a new StationGroup Agent is instantiated. Then, each newly created StationGroup instance queries the knowledge base, retrieves the information about the StationsFamily that compose it, and instantiates them. Finally, in the same way, each StationsFamily queries the knowledge base retrieves the information of each station in this StationsFamily and instantiates the Stations based on this information.

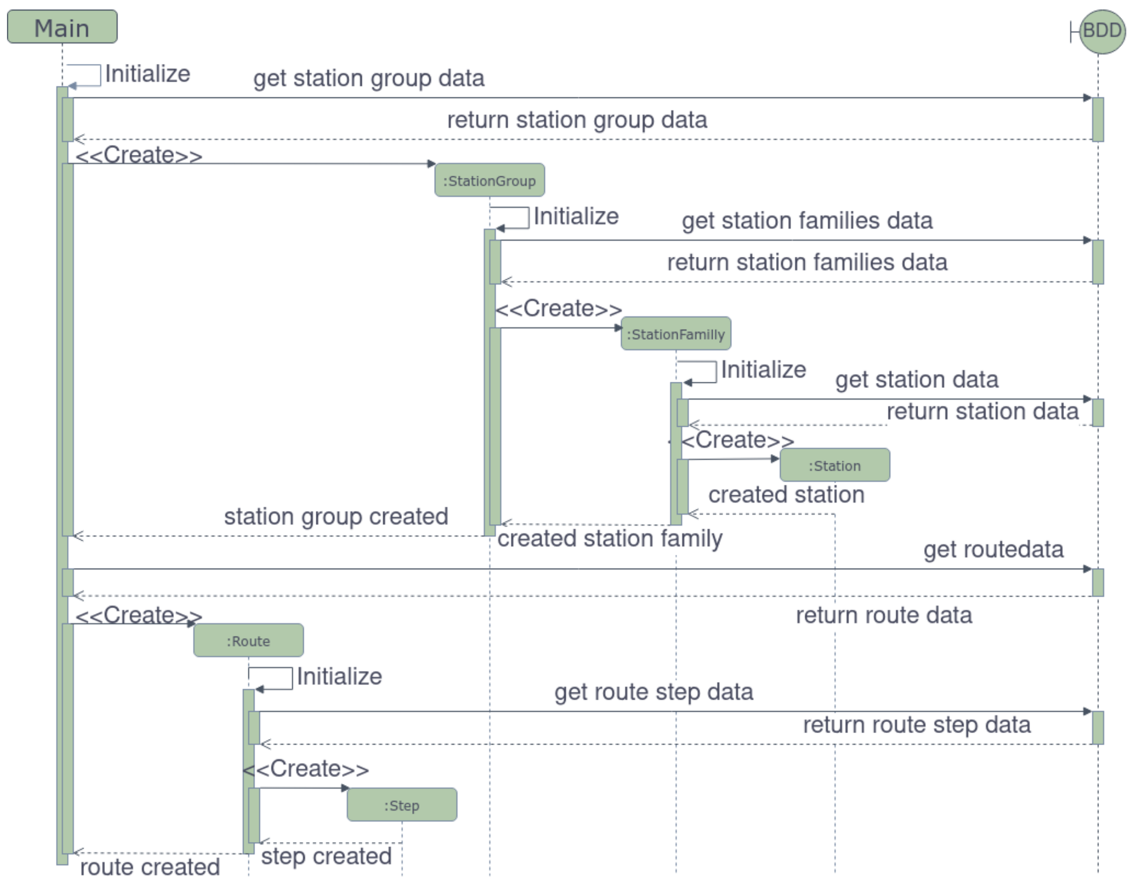

Figure 2: Sequence Diagram of the Data-Driven Simulation Model Initialization.

At this stage, the different stations and resources are defined and placed in accordance with a specific layout defined in the knowledge base. Then, agents related to the manufacturing business process are created in a similar fashion. The Main agent retrieves the manufacturing route and creates the corresponding agent. Finally, each Route agent retrieves the steps that compose it.

The initialization of the final step of the final route is the end of the construction of the data-driven simulation model.

3.3 Simulation Model Dynamics

Besides the definition of the different model components, one of the biggest challenges of fab simulation modeling is related to modeling decision rules such as scheduling or dispatching rules, and other business rules. In the proposed model, two principal rules are defined:

- `find_StationFamily()`: This function implements the scheduling policy by assigning a set of machines to a lot,
- `get_prioritary_lot_in_queue()`: This function implements the dispatching policy. When several lots wait to be processed, this function is in charge of selecting the lot that will be processed.

These rules can be easily extended to implement and test different strategies. In Figure 3, the overall dynamic of the simulation model is defined through a UML sequence diagram.

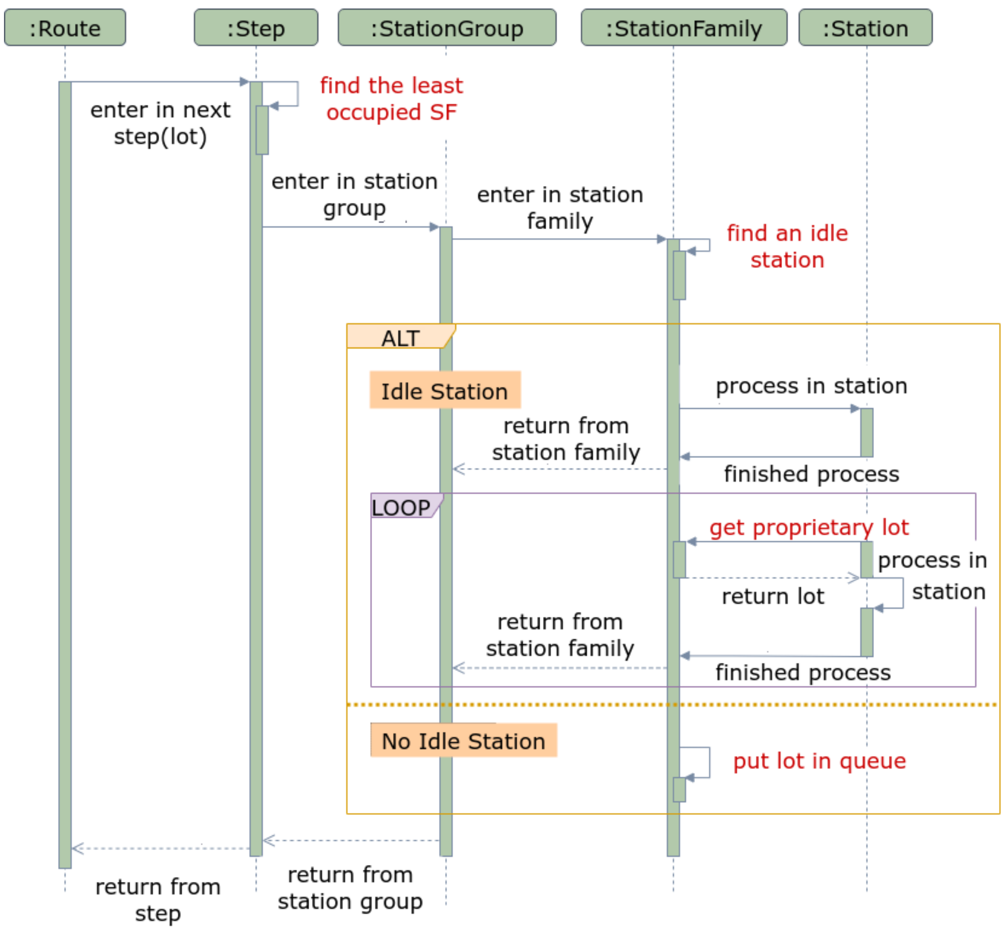

Figure 3: UML sequence diagram of the simulation model execution.

1. A lot is created by the "Route" agent,
2. The "Route" agent sends this lot to the "Step" agent corresponding to the first step of this route,
3. When the lot arrives in the step, the "Step" agent assigns it a station family according to a policy defined by the `find_SF()` function, in our case this function returns the station family with the lowest waiting time,
4. The "Step" agent sends the lot to the "StationGroup" agent corresponding to this station family,

5. The “StationGroup” agent has two main responsibilities:
   (a) Reserve a free station in the station family, in case there is no free machine, the lot is marked as waiting when it arrives in the station family,
   (b) Send a transporter to pick up the lot.
6. When the lot arrives in the “StationFamily” agent, there are two cases to consider:
   (a) The lot is marked as to be put on hold, in this case, the lot is transported and placed in the slots provided for this purpose and remains in the station family agent,
   (b) The lot has a machine reserved for it, in which case the lot is transported from its position to the machine.
7. When the batch arrives in the “Station” agent, the “process time” (processTime) of this batch is determined. The processing of the lot by the Station agent is defined as waiting for the lot for a processTime period (in minutes). When the processTime is over, the function `f_postProcess_procedure()` is executed. In the case where some batches are waiting in the “StationFamily” agent corresponding to this machine, the function `f_postProcess_procedure()` aims to select the next batch that it will process, and thus to inform the “StationFamily” agent to send it this batch. The policy according to which this lot is selected is defined by the function `get_prioritary_lot_in_queue()`.

3.4 Use Cases in Semiconductor Wafer Fab

Simulation case studies are used to address concerns about how changes to the existing fab simulation model impact production performance. The purpose of the generic simulation model is to be able to model and study a wide range of problems arising in modern semiconductor wafer fabs. We use here our model to analyze different scheduling strategies.

The experiments are conducted on an industrial instance, which includes three types of products. The basic information of the simulated instance is given in Table 2.

Table 2: Characteristic of the simulated fab.

Route	Nb step	Nb Station Family	Nb Lots
A	501	126	1560
B	440	131	1560
C	352	124	540

In total, there are 441 stations in 192 station families in the fab which are shared between the process steps of 3 types of products. There is also one transporter fleet composed of 60 operators that are responsible for transporting a lot from a piece of equipment to another or a designated stocker when a lot is waiting for a piece of available equipment. By default, a stocker exists for each station group. As the layout of the fab is an important aspect that is studied in several simulation models, our model can also take as input the layout of the fab to generate the fab accordingly. However, as this data is not always available, our generic simulation model provides a default layout as shown in Figure 4.

During the simulation, the user can use a real-time 3D navigable rendering of the fab as shown in Figure 5 that allows seeing the dynamic of the fab, with the equipment that is processing (displayed in orange), and the idle one (displayed in grey). Depending on the purpose of the simulation, other visual indicators can be added to this view.

Finally, another view allows the user to keep track of several key performance indicators and metrics during the simulation.

Figure 6 shows the different metrics collected during a simulation run of the modeled Wafer Fab that corresponds to the ramp-up phase of the Fab. It is important to notice that the aim of this section is not to analyze the results of the simulation here, but rather to highlight the analytical capability of the model.

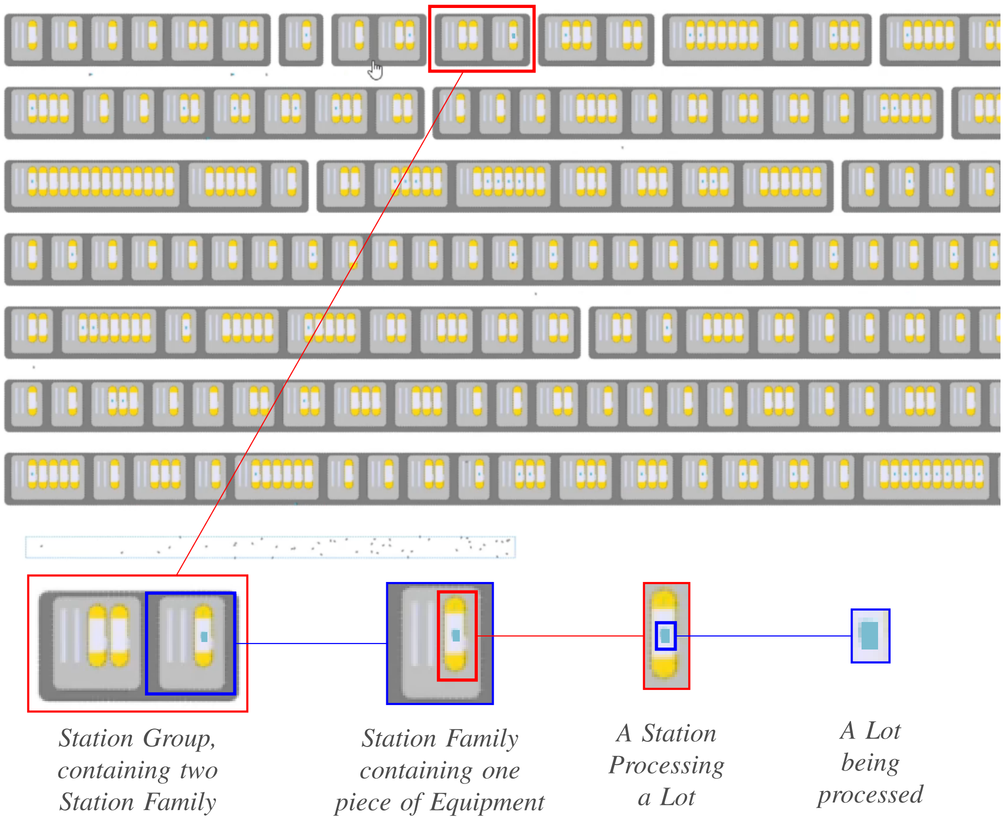

Figure 4: 2D Simulation rendering

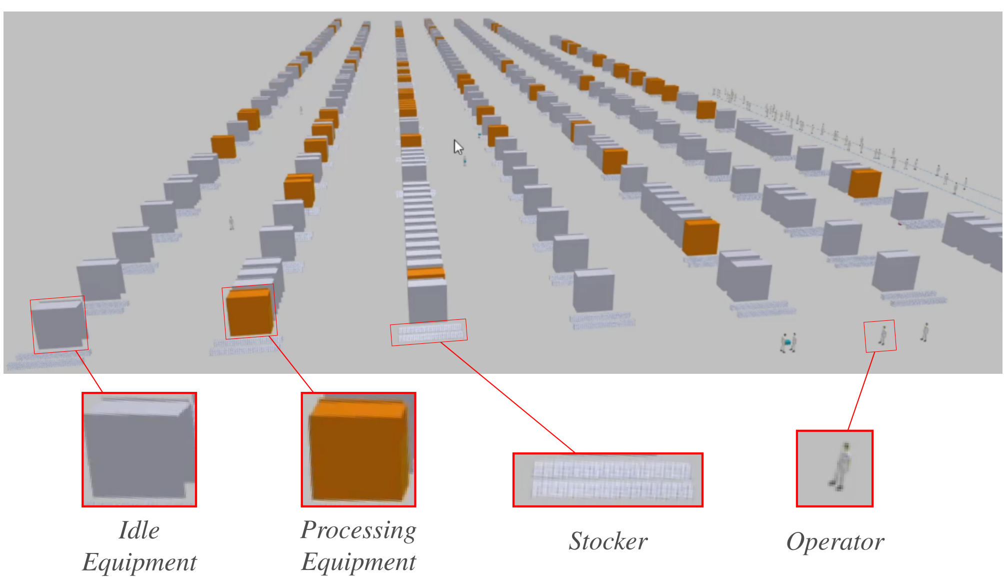

Figure 5: 3D Simulation rendering.

The different metrics give several performance measures at different granularity levels (Global Fab Level, Station Family level). Moreover, the user can get more detailed statistics as each agent have a dedicated view for statistics (Station, Station Family, Station Group, Lot, Step, and Operator). However, depending on

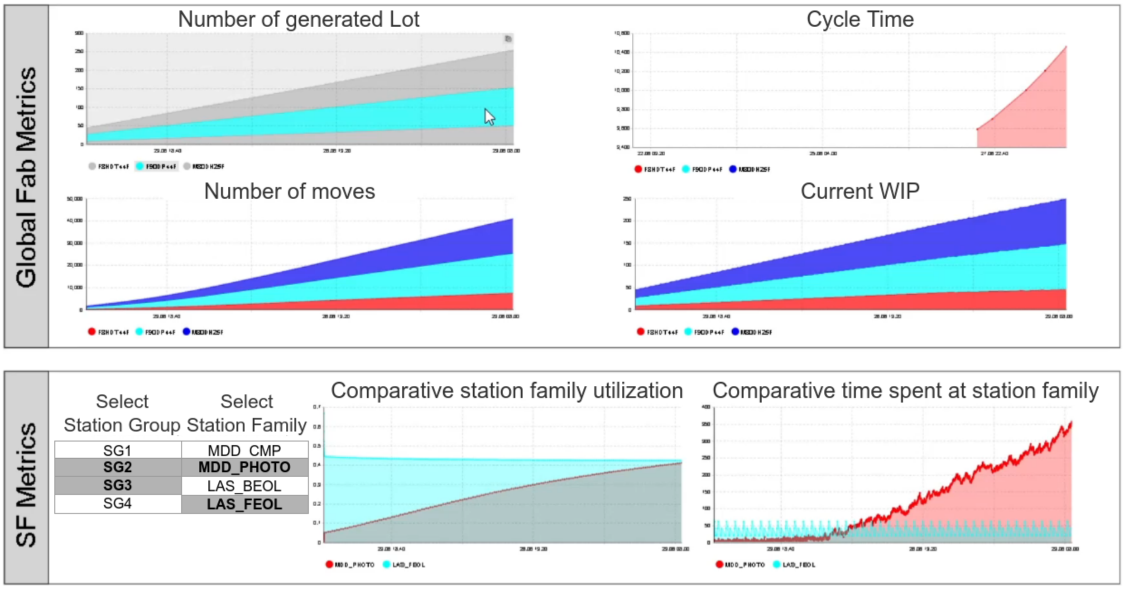

Figure 6: Simulation metrics overview.

the use case, some metrics may have to be defined by the user. As a result, the proposed simulation model can thus be used to study and address a wide range of problems that arise in semiconductor manufacturing plants with little development time.

4 CONCLUSION AND PERSPECTIVES

In this paper, we proposed a generic, data-driven wafer fab simulation model. After discussing the purpose of such a simulation model, we discussed some of the limitations encountered during modeling regarding the genericity of the model. We also addressed another limitation regarding the data used by a data-driven simulation model. Indeed, collecting data to populate a simulation model can take a long time and can lead to several semantic integration problems, which can be difficult to detect. We proposed to tackle this problem, using an ontology in the field of semiconductors.

For other research directions, it would be interesting to take advantage of semantic trends and initiatives in the semiconductor domain like SC3. Also, we plan to use the proposed simulation model to address in detail a wide range of problems arising in a semiconductor manufacturing plant and allow us to study their interactions.

ACKNOWLEDGEMENT

This work is part of the European project SC3 (Semantically Coordinated Semiconductor Supply Chains) which receives funding from the ECSEL JU under grant agreement No 101007312.

REFERENCES

Ehm, H., E. Schoitsch, J. W. van der Weit, N. Ramzy, L. Luo, and D. L. Gruetzner. 2020. "Digital Reference: A Quasi-Standard for Digitalization in the Domain of Semiconductor Supply Chains". In *Proceedings of the 2020 IEEE Conference on Industrial Cyberphysical Systems (ICPS)*, Volume 1, 563–570.

Fowler, J., L. Mönch, and T. Ponsignon. 2015. "Discrete-Event Simulation for Semiconductor Wafer Fabrication Facilities: A Tutorial". *International Journal of Industrial Engineering: Theory, Applications and Practice* 22(5).

Fowler, J. W., and O. Rose. 2004. "Grand Challenges in Modeling and Simulation of Complex Manufacturing Systems". *Simulation* 80(9):469–476.

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

Laipple, G., S. Dauzère-Pérès, T. Ponsignon, and P. Vialletelle. 2018. "Generic Data Model for Semiconductor Manufacturing Supply Chains". In *Proceedings of the 2018 Winter Simulation Conference*, edited by M. Rabe, A. A. Juan, N. Mustafee, A. Skoogh, S. Jain, and B. Johansson, 3615–3626. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

McGuinness, D. L., and F. Van Harmelen. 2004. "Owl Web Ontology Language Overview". *W3C recommendation* 10(10):2004.

Negahban, A., and J. S. Smith. 2014. "Simulation for Manufacturing System Design and Operation: Literature Review and Analysis". *Journal of Manufacturing Systems* 33(2):241–261.

Ocker, F., I. Kovalenko, K. Barton, D. Tilbury, and B. Vogel-Heuser. 2019. "A Framework for Automatic Initialization of Multi-Agent Production Systems Using Semantic Web Technologies". *IEEE Robotics and Automation Letters* 4(4):4330–4337.

Pidd, M. 1992. "Guidelines for the Design of Data Driven Generic Simulators for Specific Domains". *Simulation* 59(4):237–243.

Pillai, D. D., E. L. Bass, J. C. Dempsey, and E. J. Yellig. 2004. "300-mm Full-Factory Simulations for 90- and 65-nm IC Manufacturing". *IEEE Transactions on Semiconductor Manufacturing* 17(3):292–298.

Rank, S., C. Hammel, T. Schmidt, 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 *Proceedings of the 2015 Winter Simulation Conference*, edited by L. Yilmaz, W. K. V. Chan, I. Moon, T. M. K. Roeder, C. Macal, and M. D. Rossetti, 2896–2907. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Sadeghi, R., S. Dauzère-Pérès, and C. Yugma. 2016. "A Multi-Method Simulation Modelling for Semiconductor Manufacturing". In *Proceedings of the 19th World Congress of the International Federation of Automatic Control*, Volume 49, 727–732.

Tullis, B., V. Mehrotra, and D. Zuanich. 1990. "Successful Modeling Of A Semiconductor R&D Facility". In *Proceedings of the IEEE/SEMI International Symposium on Semiconductor Manufacturing Science*, 26–32.

Uzsoy, R., C.-Y. Lee, and L. A. Martin-Vega. 1992. "A Review of Production Planning and Scheduling Models in the Semiconductor Industry Part I: System Characteristics, Performance Evaluation and Production Planning". *IIE Transactions* 24(4):47–60.

AUTHOR BIOGRAPHIES

ABDELHAK KHEMIRI is a post-doctorate researcher in the Department of Manufacturing Sciences and Logistics at Mines Saint-Etienne. He received his Ph.D. in Computer Science from University of Aix-Marseille in 2020. His research interests include discrete event simulation, artificial intelligence, and formal methods. His email address is abdelhak.khemiri@emse.fr.

CLAUDE YUGMA is Professor at the Provence Microelectronics Center of the Ecole des Mines de Saint-Etienne. He received the Ph.D. degree from the Institut National Polytechnique of Grenoble, France, in 2003; He was a Postdoctoral Researcher at the Ecole Nationale Supérieure de Génie Industriel, Grenoble, from 2003 to 2004 and from 2005 to 2006 at the Provence Microelectronics Center. His research interests include scheduling, supply chain, and planning problems with application to semiconductor environments. He has published more than 20 papers in international journals and contributed to more than 90 communications in conferences. Claude Yugma has been involved and has supervised different research projects. His email address is claude.yugma@emse.fr.

STÉPHANE DAUZÈRE-PÉRÈS is Professor at the Center of Microelectronics in Provence (CMP) of the EMSE. He received the Ph.D. degree from the Paul Sabatier University in Toulouse, France, in 1992; and the H.D.R. from the Pierre and Marie Curie University, Paris, France, in 1998. He was a Postdoctoral Fellow at the Massachusetts Institute of Technology, U.S.A., in 1992 and 1993, and Research Scientist at Erasmus University Rotterdam, The Netherlands, in 1994. He has been Associate Professor and Professor from 1994 to 2004 at the Ecole des Mines de Nantes in France. He was invited Professor at the Norwegian School of Economics and Business Administration, Bergen, Norway, in 1999. Since March 2004, he is Professor at the Ecole des Mines de Saint-Etienne. His research interests broadly include modeling and optimization of operations at various decision levels in manufacturing and logistics, with a special emphasis on semiconductor manufacturing. He has published more than 60 papers in international journals and contributed to more than 120 communications in conferences. Stéphane Dauzère-Pérès has coordinated multiple academic and industrial research projects. His email address is dauzere-peres@emse.fr.