A Discrete Event Simulation logic for Semiconductor Production Planning and Control within Industry 4.0 Paradigm

Ahmed H. Sakr
Mathematics and Industrial Engineering Department
École Polytechnique de Montréal
Montréal, Canada
ahmed-1.sakr@polymtl.ca

Soumaya Yacout
Mathematics and Industrial Engineering Department
École Polytechnique de Montréal
Montréal, Canada
soumaya.yacout @polymtl.ca

Samuel Bassetto
Mathematics and Industrial Engineering Department
École Polytechnique de Montréal
Montréal, Canada
samuel-jean.bassetto@polymtl.ca

Abstract

The recent advancement in digital technologies has encouraged manufacturers to adopt intelligent data analysis and decision-making support tools to leverage their competitiveness. They have become challenged by the need for more flexible and complex manufacturing systems. This complexity and flexibility cannot be handled efficiently with traditional production management paradigms, such as lean management. Semiconductor industry is an example of a complex production, in which high level of flexibility is required to meet a continuously changing customer demand. This limits the efficiency of static modeling approaches and it reflects negatively on the overall production yield.

In this research, we propose a discrete event simulation logic for assessment of several challenging production management-related problems at a semiconductor manufacturer. The logic includes scheduling and processing a large variety of products, while respecting relative priorities and customers' needs. It also accounts for production resources, equipment and operators, with variable availability. Our simulation logic is data-driven, in which data that drives the different actions taken within the execution of the simulation, is compiled. Consequently, the model is highly flexible to continuous changes in the model's parameters. The proposed logic provides a quick and risk-free tool for capacity planning and reliable experimentation with different 'What-If' scenarios.

Keywords

Complex production, Discrete event simulation, Data-driven Agent-based model, decision-making support, Industry 4.0.

1. Introduction

The recent advancement in digital technologies, which is adopted by Industry 4.0 paradigm, have put a challenge for traditional production management-based industry, especially if the product or production is normally complex. These communication and information technologies, based on the Internet of Things (IoT) perception, help interconnecting shopfloor components for better management and control (Negri et al. 2017). Moreover, the IoT allows the exchange and collection of information between different system components, things, through the internet. This gives a chance for smart integrated systems that communicate with each other to achieve a certain task or to improve the general performance of the system. Similarly, the same concept is industrially applied which gives opportunity to monitor the system components, devices or equipment, remotely. This allows the integration between physical systems and virtual models which results in more representative and reliable decision-making support tools, hence more efficient and profitable production strategies (Negri et al. 2017) (Lee et al. 2014, 2015).

There are different approaches to virtually model that represent a physical system. These approaches differ in whether the system is static or dynamic, and whether it is deterministic or stochastic. According to the structure and the dynamics of the system, the suitable type of model should be selected in order to provide a good level of similarity and valid simulation results. The model parameters should be updated regularly for more accurate reality representation. This becomes more crucial in case of complex production systems where such parameters are continuously changing according to different states of the system. A product or production is considered to be complex if it is subject to variable customer demands, or if its production flow is not fixed (Freitag and Hildebrandt 2016). These parameters may include, but not limited to, the different restrictions or qualifications of the production resources. In addition, other managerial decisions that may affect different products' relative importance. Upon respecting such modeling aspects, the model will be able to accurately mirror the reality and provide consistent information for decision-making support. Moreover, this will provide a reliable tool for conducting many experiments and answering different 'What-if' scenarios (Vachlek et al. 2017).

In semiconductor industry, the fabrication areas are currently considered the most complex production systems. Traditionally the focus was on the design consideration and yield improvement rather than on the production management. However, the manufacturers have been always seeking for some tools that put the focus on the production management as a key solution (Fowler et al. 2015). This industry is defined as high-mix of products at low-volume with an end-user market-based demand which adds a source of complication for the production (Said et al. 2016). The discrete event simulation is considered a successful tool for performance assessment/ improvement and decision support in this industry (Fowler et al. 2015). The success of discrete event simulation lies in its ability to better represent the dynamics of the system along with its stochastic nature. The simulation for this complex production should consider also the dynamics of system parameters that the model is built on. This is a new consideration for simulation of dynamic systems, and the foundation for model-based decision-making support within the Industry 4.0 concepts (Vachlek et al. 2017).

In this paper, we present a simulation logic for semiconductor manufacturing system modeling with a discrete-event simulation. The proposed logic is designed to be applied using Arena® simulation software. We address the different aspects that should be considered in modeling such system, and how to include it to the model's logic efficiently. The proposed logic includes the structuring of the input data that is changing continuously. It is designed to be ready for accepting different variations in the system parameters with no need for redesign. Our simulation logic is data-driven, but it also includes agent-based logic that controls the flow of products between the different model's subsystems and modules.

2. Problem statement

2.1 Semiconductor industry background

The manufacturing process in semiconductor industry consists of a repeated cycle of several processes, which are applied to the wafers. A sequence of cycles is performed. In each cycle, a layer is built on the wafer through different processes. The number of cycles depends on the complexity of the produced device. Within the processes, wafers are tested for quality conformance before proceeding to the next steps. The wafers are either bonded with the same or different wafers, or diced into pieces that are called dies, according to the customer requirements (Fowler et al. 2015). The manufacturing firm usually consists of several fabrication areas with clean rooms environment. Each clean room contains several work areas that consist of several equipment of the same type and capabilities.

2.2 Process description and sources of complexity

The wafers are moving in lots from one step to the other. A lot is a cassette that has a fixed number of wafers of a certain size. The wafers are processed by the equipment according to its capacity. For example, some equipment handle only one wafer at a time and requires manual feeding by the operator, while others can be loaded with the whole cassette, or even more than one cassette. Other equipment processes part of the cassette, hence it requires splitting the lot into unit batches according to the equipment capacity. Some equipment are clustered in one single equipment that normally has an inside robot manipulator to handle the wafers. Accordingly, it processes several steps without the need for an operator for loading and unloading or transferring the lot. Normally, the operators are qualified to operate many equipment types according to the level of training they achieved, but not necessarily all equipment.

The processes that are achieved in each step are called recipes. Different products that have the same recipe can be processed in the same equipment, simultaneously. However, some recipes may be restricted from being processed by certain equipment due to quality issues. These restrictions are defined according to both product and recipe. Some recipes have a constraint of minimum delay between consecutive steps. In this case, the wafers have a waiting time before being processed in their next step. Other recipes have also a maximum delay constraint, or a timer limit, before which the wafers should reach a specific step. If the limit is exceeded due to different delays, the lot is scrapped in case of no possibility for a rework. The part that needs rework is sent back, while the rest of the lot proceeds to the next steps until a certain rejoin point, where it waits for the reworked part. According to the process flow of each product, some steps contains if-conditions in which the lot is split into two parts according to the result of the condition. These parts are rejoined at a certain step. Usually, the fabrication areas are not connected to each other. Manual transfer is thus required. Specific operators are only responsible for the transfer of lots between different fabrication areas.

There are different aspects for assigning relative importance, priority, to each lot. These aspects may include: the customer or product importance, the slack time, and most importantly how close to the timer limit in case there is an active timer for the lot. These priorities are used to order the waiting lots in each equipment station so that the lot which has the highest priority is processed first. The priority that is related to timers is continuously changing after each step to reduce the number of junked wafers due to the exceeding the timers' limits.

As a result of all these aspects that should be considered during processing the lots in its different steps, the process flow becomes highly dynamic. Consequently, the flow of the product is no longer fixed and is subject to stochastic alternatives that depend on each of the product itself, the manufacturing resources availability and qualification, the managerial decisions, and the variable customers' requirements. The decision-making tasks become very hard when such system is not properly modeled, hence it is nearly impossible to assess the impact of any taken decision.

3. Methodology

3.1 Selecting the appropriate simulation modeling type

Due to the high level of complexity in such manufacturing system, it is clear that such system cannot be handled efficiently with traditional production management paradigms such as lean management (Zhang et al. 2017). Such system is highly dynamic as it is vastly time dependent. It also contains many probabilistic events such as the uptime of the equipment. As shown in Figure 1, there are four types of simulation models. The Discrete-event simulation is the most appropriate for semiconductor industry modeling due to its dynamic probabilistic characteristics.

3.2 Data-driven agent-based modeling

In semiconductor industry, the products are processed according to their process flow, which consists of several steps that include specific recipes. These recipes are defined and shared among different products. Each recipe has information regarding the type of equipment that is needed, the processing time, and the information that specifies the minimum and maximum delay among steps, if any. The process flow of each product is not fixed. Some steps are added or deleted due to the possible production technological improvement. For example, some production and inspection steps are possible to be performed by a single new equipment. In this case, the process flow must be updated to account for this upgrade.

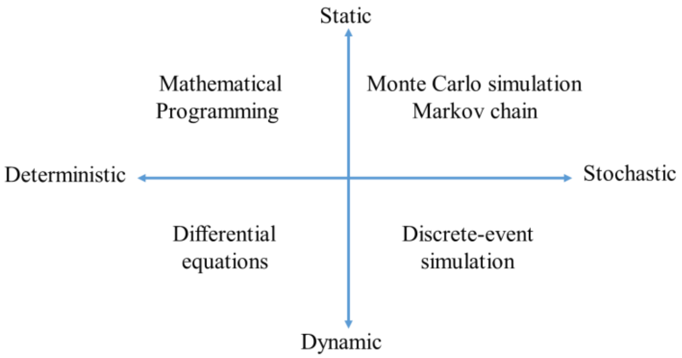

Figure 1. Simulation model types

Also, the equipment restrictions for recipes and/or product depend on some aspects including the equipment state which are continuously varying. For all these reasons, we adopted a data-driven modeling technique rather than product-driven. In data-driven models, the data controls the product flow among different steps in the model. This gives the flexibility to adopt any possible variation in flow, delay or restrictions, without rebuilding the model. It also has the advantage of building relatively more compact and simpler model to represent complex production systems that normally contain large number of resources. A simple model means that any possible errors are tracked and fixed, and any desired logic modification are easily applied.

The agent-based models have the advantage of using agents that check various aspects in the model. Accordingly, they take predefined actions that are applied to the lots to move them through the model. The advantage of having agents within the model is that many human logical procedures and actions are modeled by them, in addition to the flow of products themselves. In our logic, agents simulate the operators' logics, with different schedules and interruptions during their work shift, and the execution of complex logic such as processing a lot within a timer's limits. They are also data-driven, which means that their actions are defined according to the input data. These actions are ready for any alternation to reflect any variation in the actual system.

Our model logic has the advantage of combining the complexity of an agent-based model, and the simplicity of a data-driven model. Also, it has the ability of continuously reflecting the actual system's state without the need for rebuilding or modifying the model itself, because the data will do this task. This is in accordance with the new paradigm of simulation modeling that is among the key technologies of Industry 4.0-based manufacturing.

3.3 Input data structuring

The data is structured and is fed to Arena using MS Excel. This MS Excel is updated by the actual system databases to continuously providing the most updated system parameters for the simulation. The input data consist of the following:

1. Products basic data
2. Process flow data for all products
3. Recipe information
4. If conditions information
5. Rework information
6. Timers
7. Equipment information
8. Equipment restrictions
9. Operators information

The product basic data contains an ID, which is given for each customer and product to be used for assignments of different information within the model. It contains the interarrival time distribution function or schedules for each lot production at start, and the distribution of the lot types for each product. The lot type refers to whether the lot issued to satisfy a customer request, or it is for testing and process development reasons. A default priority is given for each customer, product, and lot type. This priority is usually predetermined according to the relative importance of each customer or products. The process flow data includes all the processing steps for each product, with all important information such as: the recipe name and ID, the fabrication area, the type of the step whether it is a normal step or an if condition step, the condition parameters to be checked, the possible rework the rework and rejoin steps for the lot splits, the then/else steps of an if condition, and the alternative recipe if any.

The recipes are indexed with an ID to be easily accessible by Arena. Each recipe has the information about its fabrication area, equipment type, the minimum delay and timer if it requires a time limit, the average yield based on calculations from the historical lots data, the recipe processing time and operator time, the recipe type whether it is normal, bonding, dicing step, and if this recipe has an equipment restriction.

The 'if' conditions are defined for each product and the average percentage of having a 'Then' statement is defined based on the historical lots' data. Similarly, the rework information for each product at each step is defined.

The timers are defined for each process flow of each product. This includes an ID for that timer to be accessed by Arena, the time limit of that timer, the number of steps within the timer, and the start and end steps. This information is used by a special logic for processing lots that have a time limit for delay.

The equipment information includes an ID for each equipment, along with an ID for its type and its fabrication area. The capacity is also defined with the number of wafers or full cassettes that equipment processes. Also, the uptime/downtime distribution for each equipment, based on the history of that equipment, is included. These distributions are used by the agents that are responsible for simulating the availability and the failure of the equipment. The equipment restrictions' information contains the restrictions or dedication of specific equipment for certain recipes, or certain recipes for specific products. This is used when assigning an equipment to process the lot at each step.

The operators' information includes the operators' IDs of each fabrication area and their qualification for different equipment types. It also includes their working shifts and break times to be used for their activity simulation by the agents who are responsible for this task. Finally, it contains the interarrival times for interrupting events and their time distributions for simulating possible interruptions of the operators during their working shifts.

3.4 Simulation logic

The input data is ingested to the model to drive its different actions. The run logic begins with reading the input data and assigning it to appropriate variables or expressions that can be handled by the simulation modules. Then different agents are created to execute different tasks according to information from the input data; specifically, the number of operators, the number of the equipment types, and the total number of equipment.

Figure 2 shows the working logic for the equipment stations' agents. These agents are responsible for moving the lots from the queue of each equipment type to be processed. This logic helps respecting the priority for the waiting lots, and provides simplicity for the modeling as each lot may have restrictions or dedications for different equipment within the same equipment type station, hence the traditional method of seizing resources, equipment and operators, will be more complicated to realize the same logic. The agent searches for all other waiting lots that have the same recipe to be processed with the first waiting lot according to the equipment capacity, as usually the equipment cannot be accessed once processing a lot.

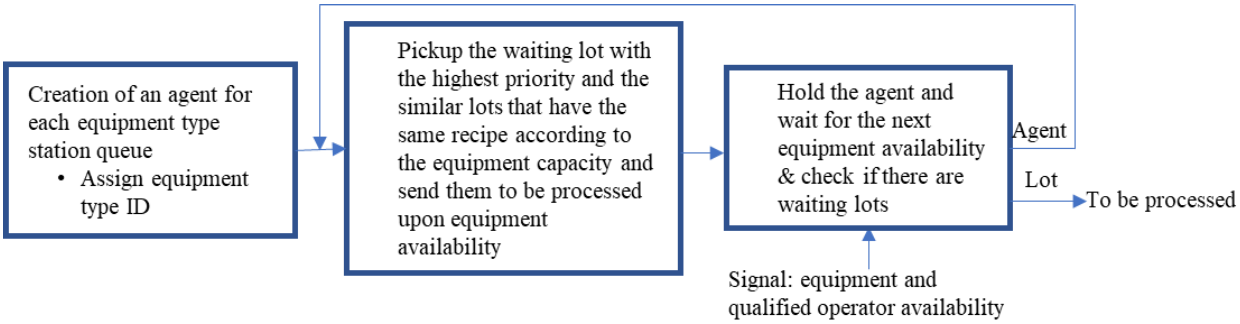

Figure 2. Working logic of equipment stations' agents

This logic is facilitated by the attributes, or information, that are assigned to each lot including its product ID and step ID. The agent identifies the lots through this information. Accordingly, it acquires other information regarding the lot's current step from the input data. This also reduces the unnecessary duplication of data and eases its synchronization with the actual system database for continuous updating.

Figure 3 shows the working logic for equipment uptime/downtime agents. The corresponding agents are responsible for assigning active or failure states to their equipment according to the time distribution functions for both uptime and downtime. The distributions are obtained, from the input data, during the model run. These distribution functions are calculated from the history of each equipment, and these calculations are updated regularly, to accurately detect any change in the uptime/downtime due to improved maintenance policies, for example.

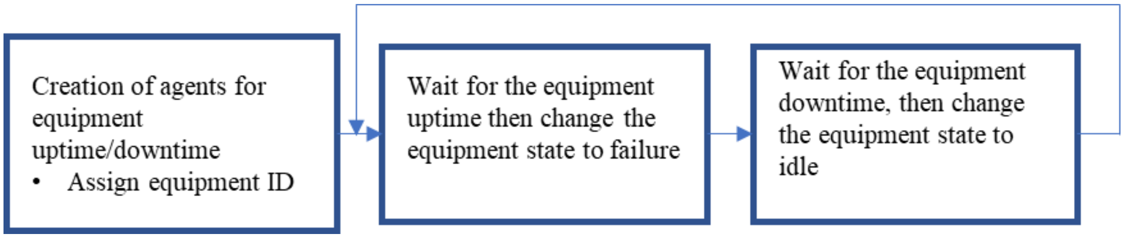

Figure 3. Working logic of equipment uptime/downtime agents

Figure 4 shows the working logic for the operators' schedules and interruption events' agents. These agents are responsible for assigning the active/inactive states to their operators according to each operator's schedule and assigning the active/failure states due to any possible interruption event. These events arrive randomly according to a given interarrival time and for a given duration which is calculated from records of observed interruptions for the operators in the actual system. The schedules are given in the input data for each operator, hence it is updated upon any possible modifications of the actual operators' schedules.

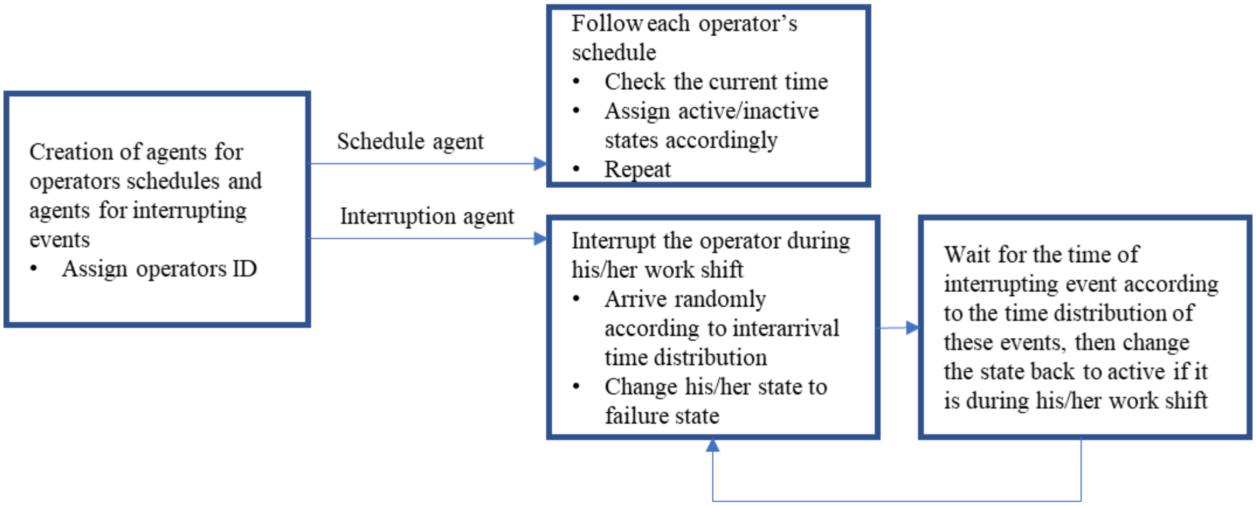

Figure 4. Working logic of operators' activity/interruption agents

Figure 5 shows how the lots are created and how they are prepared for their first processing step. The important information is assigned for each lot upon creation. This information includes the product ID that is used to define the process flow and information about the steps. The relative importance information of a product is checked along with other prioritization aspects such as timer's limits. Consequently, the priority is assigned and is used for ranking the lot among others. The lots then wait, in the equipment station queue corresponding to their steps, for the equipment station's agent to move them for processing upon availability of the equipment and the qualified operator.

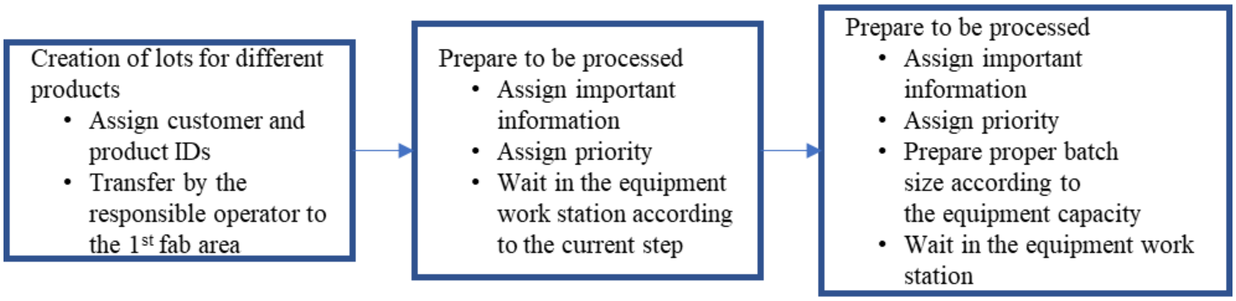

Figure 5. Lot creation and assignment of important information logic

Figure 6 shows how the lot is getting ready to be processed in its current step. The lot is first checked for any equipment restrictions, then it is assigned to the available equipment and operator to be processed. The lot is then delayed for the processing time, and then it is unloaded by the operator from the equipment and is sent to the after-processing logic. If there is an equipment restriction, the lot is sent back to wait in the station's queue until another equipment becomes available within the same station.

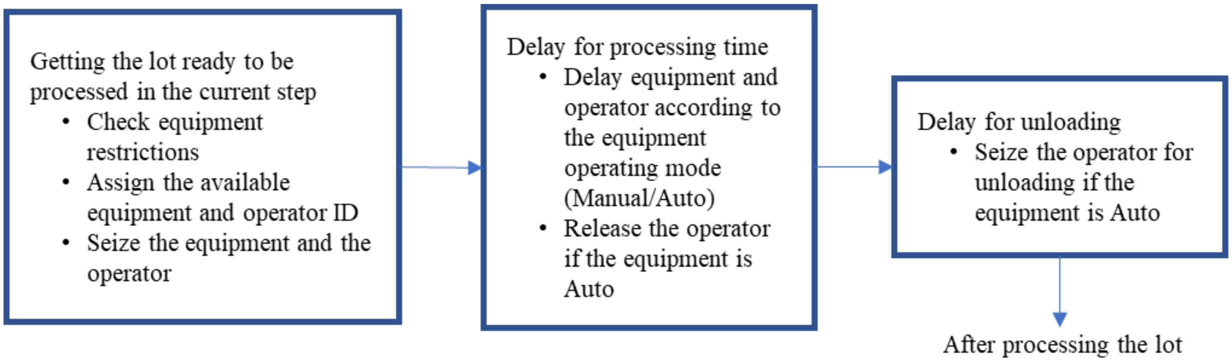

Figure 6. Lot processing logic

Figure 7 shows how the after-processing logic. This logic includes the following:

1. Batching the wafers from the same lot if the lot was not processed at once due to the limited equipment capacity.
2. Disposing the disqualified wafers form the lot according to the yield of each recipe.
3. Checking if the current step may have a rework procedure and if any wafers requires a rework. In case a rework is required, the lot is split and the part that requires a rework is sent to the corresponding rework step.
4. Checking if the current step is a rejoin step for a rework procedure. The lot will wait in this step until its reworked parts rejoin it.
5. Checking if the next step is an if condition. If yes, the lot is split and sent to the Then/Else steps of that condition.
6. Sending the lot to its next step. The lot may wait if the next step is not within the same fabrication area. In this case, the lot waits for the responsible operator to make the transfer.

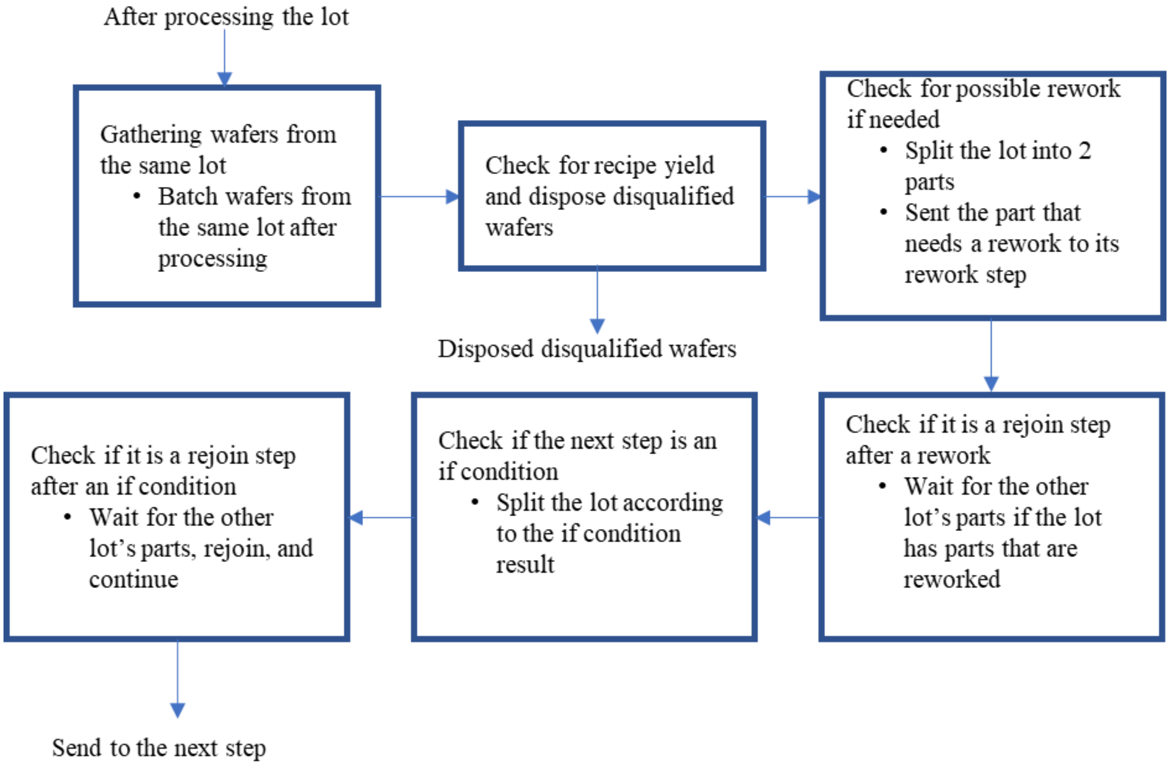

Figure 7. The after-processing logic

The lots are subject to many splits such as reworks or 'if' conditions. A log is created for each lot within the logic and is realized by global variables in Arena. These logs are used for tracking of the splits of each lot according to the lots ID and the number of wafers in each split. Normally, the splits are subject for scrapping due to many reasons such as low yield recipes. In this case, the logs are important for rejoin steps to determine the number of lot's parts that are rejoined to form a complete lot before proceeding to the next steps.

Due to the importance of processing of lots within timers, an agent is created for each lot that is entering several steps within a timer. This agent assesses the criticality of this timer by comparing the total processing time for the steps within the timer to the time limit of the timer. If the timer is critical, the agent seize all equipment in advance before processing the original lot within that timer, while respecting the priorities of the waiting lots. If the timer is not critical, the agent check if all required equipment are up or not, and decide to proceed or hold the lot. If the equipment is down, the agent waits until the equipment become up and then check if there is a waiting lot that is higher in priority than its lot. If there is no higher priority waiting lot, the agent's original lot is processed. This logic respects the lots priority and processes the lots within timers safely to reduce the junked or reworked lots due to exceeding the timers' limits.

4. Goals

By applying our proposed logic, the simulation provides default results including the following:

1. Lots:
    - Total number in/out and total time
    - Value added/ non-value-added time
    - Waiting time
    - Total work in process (WIP)
2. Resources: the instantaneous utilization
3. Queues: average number and time of waiting in each queue

These indicators provide a general insight into different bottlenecks and their possible reasons. More details are acquired by using counters or tallies for each type of product to detect where and why the product has the most delay.

When addressing different problems in the semiconductor's industry, some manufacturers cannot assess the future impacts of their decisions due to the lack of a real representative model. Furthermore, problems are addressed in a reactive mode rather than a proactive. As a result, the cycle time increases dramatically for most of the products because of the increased downtimes or improper capacity planning. Upon realizing a representative model, decision makers are able to evaluate their different decisions and realize the possible upcoming problems in advance, and how to prepare the actual system to tackle them.

Our model logic is a basis for conducting different experiments and assessing different 'What-if' scenarios at no risk or any financial loss. This efficiency is due to the continuous synchronization of different parameters with the actual systems, which ensure that the model is always reflecting the reality. ANOVA is beneficial for evaluating the significance of different parameters on the various performance indicators of the model. This helps the decision makers to focus on solving different problems with the most significant aspects rather than applying temporary solutions without assessing their future impacts.

5. Verification and validation

After building the simulation model, it must be verified and validated according to its purposes. Verification is achieved if the modeling assumptions are correctly translated into a computer model. This is done by debugging the simulation model to detect any deviation from the modeling assumptions. For example, our model is tested to check whether the priorities of the waiting lots are correctly assigned and respected or not throughout the model's execution. Thus, the model is verified to follow the modeling assumptions if no assumption is violated due to a modeling error. For validation, the model is tested for its representation accuracy of the actual system that it simulates. This is done within the scope of the particular objectives of the simulation study (Law 2015).

A good technique for verifying our simulation logic is to debug the model in modules or submodels. This is due to the various logical aspects that are included in the simulation such as assigning priorities, batching wafers, lot split and rejoin. This task is possible during the model building. The model building is divided into building submodels in which each realizes a certain logic. Upon building a submodel, it is tested for its conformance to the relevant modeling assumptions. Hence, debugging and correction, if needed. The submodels are added and debugged successively until reaching the full model.

The model validity is established when its outputs closely resemble the actual system's ones. This validity test is achievable through the comparison with the actual system under the same conditions. After verifying our proposed simulation logic, the historical data from the actual system is used to recreate the same manufacturing scenarios through the simulation model. This approach is called the 'correlated inspection' which results in comparable output statistics from both the model and the actual system. If the model's outputs compare closely to that of the actual system, then it is considered a valid model. According to the correlation between the simulated and the actual outputs, the confidence in the model is built. Such comparisons are realized through numerical statistics. Our model validation relies on sample means comparison for many production scenarios. If the difference is negligible, then we conclude the model's validity.

6. Conclusion and Future works

The semiconductor industry is an example of a complex production due to the manufacturing technologies and the flexibility required to satisfy customer requirements. The manufacturers always seek for reaching production performance improvement in order to keep a high level of competitiveness. This implies the implementation of untraditional production management strategies that can handle the increased level of production complexity. Such problems are addressed by the applications of the Industry 4.0 concepts that adopt the IoT technologies and synchronized system models which copies the actual system continuously.

The basis of having a good representative model is selecting proper methodology and type of modeling that is able to simulate the behaviors of the actual system. Nevertheless, it should be continuously monitoring the actual system to adapt for any modifications that may occur. We proposed a methodology for modeling a production system for semiconductors using discrete-event simulation model that is both Data-driven and Agent-based. The proposed logic is easily applicable using Arena® simulation software. Our logic included the flexibility for accepting any updates in the model parameters with no need for rebuilding the model. This was realized while keeping the modeling as simple as possible to be easily assessed and validated.

Future work for this research involves conducting design of experiments with the model results for real input data to understand different relationships between model outputs and different model parameters and inputs. This would help solving different problems and improving the general production performance of the actual system.

References

Fowler, J., Monch, L., and Ponsignon, T. ¨ Discrete-event simulation for semiconductor wafer fabrication facilities: A tutorial. *International Journal of Industrial Engineering*: Theory Applications and Practice 22, 5 (2015), 661–682.

Freitag, M., and Hildebrandt, T. Automatic design of scheduling rules for complex manufacturing systems by multi-objective simulation-based optimization. *CIRP Annals 65*, 1 (2016), 433 – 436.

Law, A. M., *Simulation Modeling and Analysis*, 5th Edition, McGraw-Hill Education, New York,2015

Lee, J., Bagheri, B., and Kao, H.-A. A cyber-physical systems architecture for industry 4.0-based manufacturing systems. Manufacturing Letters 3 (2015), 18 – 23.

Lee, J., Kao, H. A., and Yang, S. Service innovation and smart analytics for industry 4.0 and big data environment. Procedia CIRP 16 (2014), 3 – 8. Product Services Systems and Value Creation. Proceedings of the 6th CIRP Conference on Industrial Product-Service Systems.

Negri, E., Fumagalli, L., and Macchi, M. A review of the roles of digital twin in cps-based production systems. Procedia Manufacturing 11 (2017), 939 – 948. *27th International Conference on Flexible Automation and Intelligent Manufacturing*, FAIM2017, 27-30 June 2017, Modena, Italy.

Vachlek, J., Bartalsk, L., Rovn, O., imiov, D., Morh, M., and Lokk, M. The digital twin of an industrial production line within the industry 4.0 concept. In *2017 21st International Conference on Process Control (PC)* (June 2017), pp. 258–262.

Said, A. B., Shahzad, M.-K., Zamai, E., Hubac, S., and Tollenaere, M. Towards proactive maintenance actions scheduling in the semiconductor industry (si) using bayesian approach. *IFAC-Papers OnLine 49*, 12 (2016), 544 – 549. 8th IFAC Conference on Manufacturing Modelling, Management and Control MIM 2016.

Zhang, Y., Ren, S., Liu, Y., and Si, S. A big data analytics architecture for cleaner manufacturing and maintenance processes of complex products. *Journal of Cleaner Production* 142 (2017), 626 – 641. Special Volume on Improving natural resource management and human health to ensure sustainable societal development based upon insights gained from working within Big Data Environment.

Biographies

Ahmed Sakr is a PhD student in Mathematics and Industrial Engineering at École Polytechnique de Montréal in Canada. He holds a MSc. and a B.Sc. in Mechanical design and production Engineering from Cairo University in Egypt. His research interests and specialties include: Machine learning and artificial intelligence application in operating systems, Manufacturing and service centers modeling, simulation, and analysis, Decision Making for Industrial Systems for optimal performance and waste reduction, Analysis of variance and Design of experiments, Mechanical vibrations and dynamics of mechanical systems. He has developed passion for interdisciplinary tactics to improve operating systems performance. His doctoral research involves simulation and improvement of complex product manufacturing systems performance through the realization of the Big data analytics and IoT technologies. He completed his MSc. and BSc. at Cairo university in renewable energy and vehicle dynamics respectively.

Soumaya Yacout is a Professor in the Department of Mathematics and Industrial Engineering at École Polytechnique de Montréal in Canada. She holds a D.Sc. in operations research from Georges Washington University in U.S.A.; and a B.Sc., and a M.Sc. in Industrial Engineering from Cairo University in Egypt. She designed and taught courses on quality engineering, reliability, and maintenance for undergraduate, graduate, and professional engineers. Her research interests include Condition Based Maintenance, and distributed decision-making for product quality. She has publications in journals including Quality Engineering, International Journal of Production Research, Computers and Industrial Engineering, The European Journal of Operational Research, and JESA. She is a senior member of the American Society for Quality; and a member of the Institute of Industrial Engineering, and the Canadian Operational Research Society.

Samuel Bassetto is an Associate Professor in the Department of Mathematics and Industrial Engineering of Polytechnique Montréal, Creative, dynamic and passionate. Over the years, he has directed more than 150 graduate students, CIMAR-LAB to structure researches on continuous improvement and system engineering. He graduated from ParisTech in 2001, on mechanical engineering while in the meantime got his MSc from University of Nancy in the field of Artificial Intelligence. He earned his PhD in 2005, from ParisTech, in the field of Industrial Engineering and Systems Modeling, Pierre-Bezier and GDR MACS, ADRIQ and NSERC prices. His investigations are now focused on the integration of AI in system engineering and habit development.