Artificial Data Generation for Smart Manufacturing Systems: Discrete Event Simulation, Product Traceability, and Process Mining

Alexandre Bellargus S. Costa[1] **, Juliano Y. Nishura**[1] **, Paulo Victor Lopes** [1,2] **, Filipe Alves Neto Verri** [1] **, Anders Skoogh** [2]

[1]Dept. of Computer Science – Instituto Tecnológico de Aeronáutica (ITA)
São José dos Campos – SP – Brazil

[2]Dept. of Industrial and Materials Sci. – Chalmers University of Tech.
Gotemburgo, Suécia.

alexandrebellargus@gmail.com, julianoyoshiro@gmail.com

victorf.lopesbr@gmail.com, verri@ita.br, anders.skoogh@chalmers.se

Abstract. *Industry 4.0 is revolutionizing the industrial sector with advanced technologies. In this scenario, process mining - data mining for business and industrial processes - is essential for optimizing operations by taking advantage of real-time data on performance, behaviour and trends. However, obtaining professional data and detailed models to apply process mining techniques is challenging. CLEMATIS is a discrete event simulation (DES) model that aims to generate data for this purpose, however, it was not meeting the technical requirements of process mining. This study therefore improves CLEMATIS to make it compatible with process mining techniques. The methodology involves the establishment of requirements, the traceability of production line resources via tokenization, the development of a visual simulation tool, and the construction of compatible event logs for commercial and academic use. The results shows the model's effectiveness in applying process mining techniques in real time, meeting the needs of academic research in this area.*

1. Introduction

The significant increase in complexity in industrial production lines, along with their constant mutations, has created challenges in the analytical modeling of these processes, making it a costly task [Van der Aalst et al. 2004]. As a result, there has been a growing demand for the application of discrete event simulation (DES) methods to production lines in order to meet the challenges posed by these dynamic and complex changes. In this scenario, CLEMATIS, a model for generating artificial production lines, has emerged as a disruptive and interesting tool aimed at obtaining artificial data to encourage further studies on this topic [Lopes et al. 2024].

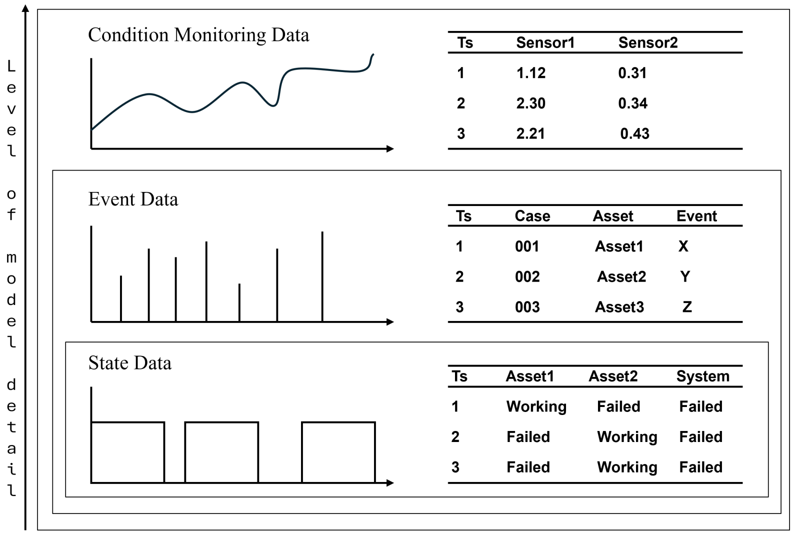
A DES model is a computer representation of a system in which events occur at specific points in time and have an impact on the state of the system [Fishman 2001]. In this type of simulation, time is divided into discrete intervals and events occur at specific times. Each event can alter the state of the system, such as changing variables, modifying queues, updating resources, among others.

In the context of analyzing production lines, the DES model provides a huge theoretical advance in a scenario where it is difficult to acquire professional data available for study as industrial productions lines generate business sensitive data. The synthetic data generation (SDG) allows advancements in many areas, including generation of data for AI/ML models. CLEMATIS is therefore a tool that, through DES, generates data from artificial production lines.

However, in order for a data-driven simulation model to be reliable, the level of detail that this data offers about the system must be analyzed. In the context of process mining, there are levels of detail that event records fall into: state data, event data and monitoring condition data [Friederich et al. 2021].

One understand these levels from the Fig. 1 taken from the original article by Jonas Friederich, et. al. [Friederich et al. 2021]. But in summary they are defined as:

1. State data: Records the different states of the assets and system of an Intelligent Manufacturing System (IMS), such as running times, idleness and failure. This data is a low-level source of information, but requires less effort to provide.
2. Event data: Records discrete events generated by the assets and the system, marking the start and end of activities relevant to the simulation. This data provides valuable information about the manufacturing system, but requires more effort to obtain detailed data from the assets.
3. Condition monitoring data: Records relevant data on the health of the IMS, from sensors embedded in the assets or installed in critical locations. This data increases the level of detail of reliability and simulation models, allowing for deeper insights and the generation of detailed failure models.



Ts	Sensor1	Sensor2
1	1.12	0.31
2	2.30	0.34
3	2.21	0.43

Ts	Case	Asset	Event
1	001	Asset1	X
2	002	Asset2	Y
3	003	Asset3	Z

Ts	Asset1	Asset2	System
1	Working	Failed	Failed
2	Failed	Working	Failed
3	Failed	Working	Failed

Figura 1. Requirements for data-driven reliability modeling [Friederich et al. 2021].

State data records the different states of the assets and system of an Intelligent Manufacturing System. Event data records discrete events generated by the assets and the system. In turn, condition monitoring data records relevant information from sensors embedded in the assets. Taking this information into account, we can see the original CLEMATIS output in machine status data below in Tab. 1.

Tabela 1. Example of CLEMATIS original output.

Time	Vertex	State	Step
1	0	working	0
1	13	blocked	0
1	24	working	0

Based on this analysis, CLEMATIS has a level of detail in state data that is below what is necessary to apply all the discovery techniques we have at our disposal.

1.1. Research Challenge and Objectives

Thus, our main objective is to increase this level of detail to *event data* so that it is possible to generate artificial production line data compatible with the main known discovery techniques.

So our objective is to support the field of industrial smart systems through the generation of datasets that represent production lines. The specific objectives are:

a) Propose a *tokenization* strategy for an existing production line simulator called CLEMATIS, to generate event logs in an efficient way;
b) Development of a visual tool to support the scheduling and planning activities through the representation of dynamic behaviour of the simulated system;
c) Application of process mining techniques to showcase the synthetic data processing, focused on processes and production lines; and
d) Conformance checking to evaluate the system capabilities.

Therefore, our research challenge involves generating high-quality datasets for applying data mining techniques and consequently developing smart systems.

To address this problem, we divided the methodology of this work into four main parts, using the technologies mentioned in the introduction and process mining techniques.

First, we assembled the fundamental requirements to increase the detail level of the previously structured model, as discussed in the introduction. Next, we structured the methods needed to implement the traceability of production line resources using the *tokenization* technique, which will be discussed in the following subsection. We then explored the steps to build the event logs of the artificial production line and assessed the compatibility of this data for commercial and academic uses. Finally, we developed an extrinsic visual analysis tool developed to run in parallel with CLEMATIS to facilitate the visualization of the data generated by the model.

2. Theoretical Background

Real data from industrial devices is often sensitive, proprietary, or costly to collect, which constrains the use of AI/ML techniques to analyze discrete material flow processes. In Industry 4.0, SDG can enable AI and ML applications. These SDG techniques ensure data accessibility and authenticity, supporting tasks to optimize manufacturing processes and evaluate system performance without the need for real data. The next subsections address the challenges and theoretical background to contextualize the generation of event logs and traceability data to train AI/ML models, and how process mining can be considered as a powerful data mining tool for this purpose.

2.1. Synthetic data generation for AI/ML in Industry 4.0

Industrial devices can generate and store real data, but this data is often sensitive, proprietary, or requires a high allocation of resources to be collected [Anderson et al. 2014]. To represent real-world systems properly, the balance between data accessibility and authenticity needs to be ensured [El Emam et al. 2020]. The SDG techniques are an alternative capable of bridging the gap between real data availability and the demand for data sets to train some AI and ML models [Libes et al. 2017] The utility of SDG can even extend to data generation, augmentation, or even knowledge transference from similar systems through the use of pre-trained meta models [Piga et al. 2024].

Especially in complex production systems, SDG can address challenges of high product variance, component variety, and different production routes, increasing the information available to improve the efficiency of processes in industry [Nguyen et al. 2022]. SDG can also simulate various states of manufacturing parts, facilitating more accurate and efficient AI/ML models applications [Manettas et al. 2021]. Another strategy for SDG use is to combine synthetic and real data to extend the coverage of scenarios within the training dataset, addressing class imbalance issues and improving the model's ability to generalize from small data sets to real-world scenarios [Gutierrez et al. 2021]. The ability to generate coherent synthetic datasets signifies broader implications to industrial data collection challenges, thus advancing the deployment of AI in manufacturing environments [Hodapp et al. 2020].

subsectionTraceability Systems and Product Traceability

Traceability is a trending terminology that increased in visibility since its first use in international industrial standards [22]. Traceability is an umbrella term that refers to the practice of identifying an object or work item and obtaining all the information about it at every stage of its life cycle [Schuitemaker and Xu 2020]. Traceability is usually achieved by using a unique mark or label on the object, recording the data and movements from the beginning to the end of the production system.

The traceability systems are composed of the principles, practices, and standards that support product traceability [20]. Initially, traceability systems relied on paper-based methods until the invention of the bar code in the 1950s, which enabled the beginning of the digitized traceability era [25]. Nowadays, Industry 4.0 traceability systems are computerized, utilizing ICT systems instead of traditional manual or paper-based methods. Nevertheless, the practical implementation of traceability remains a significant challenge for many companies [11]. This challenge constrains the collection of traceability data in scale and consequently the use of AI/ML algorithms to discrete material flow systems.

The benefits of traceability systems have been already showcased in industrial case studies [58] and research papers [28]. These benefits can extrapolate quality assurance and claims reduction, for example in inventory management, stock optimization, and waste recycling [32]. The transfer potential from theory to industries depends on the user interests and availability of technologies [28]. For example, the availability of datasets and structured ways of generating data can enable AI/ML adoption, serving as a leading technology in this context.

2.2. Process Mining and Data Mining (discovery, conformance and enhancement)

Advancements in process mining now allow the discovery, analysis, and improvement of business processes using event data recorded in event logs. This method reveals the true nature of processes, often uncovering gaps in understanding and suggesting solutions, much like an X-ray [Van Der Aalst 2012]. Process mining discovers, monitors, and enhances real processes by extracting knowledge from event logs, which are abundant in modern information systems. Traditional data mining methods are not process-centric, leading business process management to rely on handmade models. Process mining bridges this gap by providing a more accurate, data-driven approach .

Process discovery involves learning models from event logs, often simplifying attributes to activity names. Beyond discovery, process mining includes conformance checking and enhancement. Conformance checking compares models to actual behavior to identify discrepancies, ensuring that processes align with intended workflows. Enhancement focuses on optimizing processes by analyzing event logs from three perspectives: process, organizational, and case .

The process perspective examines control-flow and activity order, aiming to characterize all possible paths. The organizational perspective analyzes the originator field to classify roles and build social networks. The case perspective looks at case properties, such as paths and data element values, to gain deeper insights into specific instances like replenishment orders. These analyses help organizations optimize their workflows and improve overall process performance, making process mining a comprehensive approach to managing and enhancing business processes [Aalst 2005].

3. Methodology

Product traceability is used to determine the physical location of the item. Process traceability, on the other hand, aims to identify the type, sequence, and variables of the processes that affect the product. Indeed, in the context of artificial production lines, such as CLEMATIS [Lopes et al. 2024], traceability lies in the act of tracking the exact location of all the resources inserted into the production cycle. In the next subsections, we describe our traceability implementation which modifies the data structure used in the simulation model so that the product ID and time during each resource passage are also taken into account.

3.1. Requirements for Event Logs Generation

In order for the Event Log to move to the second layer of the level of detail of industrial monitoring systems, the records generated need to be traceable and the exposure of this traceability needs to follow a sampling pattern that fits the main process mining methods. To conform to this standard, the exported data needs to meet database construction requirements.

The requirements for this process are:

1) Each resource is represented by a different id;
2) Each resource goes through all the stages of the production process;
3) The resources available at the entrance to the production line are finite;
4) Each machine has a buffer for storing resources;

5) Each machine has a production time represented by a probability distribution;
6) Each entry in the event log is represented by a passage of resources from one machine to another at a given time;
7) Data is saved in .csv or .xes[1] format.

3.2. Generating Event Logs

In the manufacturing industry, process analysis receives attention from companies because process optimization is directly linked to a reduction in production costs and overall time reductions [Yang et al. 2014]. Thus, the proposed modification to CLEMATIS aims to meet this demand and increase the scope of possible uses for this tool.

During the empirical analysis of process discovery algorithms, generally three requirements regarding the test data must be met. Firstly, one must have full control over the control flow characteristics of the event data generated. A second requirement is randomness to avoid incorrect generalizations based on non-random event data. Finally, the final event logs and reference models must be in standard format[2] to ensure their compatibility with tools that implement process discovery algorithms and evaluation metrics [Jouck and Depaire 2016].

Originally, CLEMATIS did not meet any of the requirements presented, as we can see in the Tab. 2, making it difficult to analyze this data using the discovery techniques implemented in the context of process mining.

Tabela 2. Example of the original CLEMATIS output.

time	vertex	state	state_id	buffer_occup.	prod._step
1	0	working	2	0.0	0
1	9	working	2	0.0	0
1	12	working	2	0.0	0
1	13	working	2	0.0	0
1	24	working	2	0.0	0

The first requirement for implementing process discovery is the traceability of objects on the production line. In order to verify the traceability of the products, we proposed the experiment of calculating the *cycle time* of the resources that enter the artificial production line, i.e. the analysis of the times and routes that each resource takes to pass through the production line.

For the second requirement, it was proposed that the production time of the machines should follow a probability distribution, with the intention of generalizing the data generation. The probability function chosen was the *Poisson*[Fernández and Williams 2010] function, as it models the frequency of occurrences of operations over time well, suiting the needs of the problem[Letkowski 2012].

Finally, in order to store the data in a more comprehensive format, the *.csv* and *.xes* formats were chosen as the saving mechanisms for the event logs generated. This

[1] .xes is an extension used to standardize the sending of event data, https://xes-standard.org/https://xes-standard.org/.

[2] The standard format for process mining is usually .CSV or .XES, which are the formats supported by most programs and libraries.

change opens up many options so that the data generated can be used on other platforms such as the main compliance analysis libraries PM4PY and SIMPY.

In general, the structural update in CLEMATIS logic can be described by the following pseudo-code:

```
while producted < production:
    update_time()
    for node in reversed(sorted_nodes_list):
        # if an "in" node is not with
        # the buffer full, fill it
        node.fill()
        # check if any of the elements
        # feeded by node has space to receive
        o_production = node.check_nexts()
        if not node.failed():
            node.produce(o_production)
            next_node = node.get_next()
            pass_tokens(node, next_node)
            producted += o_porduction
```

The logic implemented consists of going through all the layers, from the output layer to the input layer, checking whether the producers in that layer can produce something, and if they can, who they can pass that resource on to, statistically calculating the probability of that node working.

The resource is passed on using the tokenization method presented previously, in which each resource is marked by a token and its passage through the network is analyzed.

Finally, each resource that leaves the output layer is added to global production and all the passes that the token for that product has made through the production chain are added to the event log.

To analyze the improvements made, it is proposed to compare the original model outputs with the current output and apply 2 process discovery techniques to verify that the changes really had the expected impact on the model.

3.3. Visual Simulation Tool

For use in industry, the record of events that *CLEMATIS* artificially generates, with the improvements presented, already has a very comprehensive use and is in line with the reality of international industry, in terms of traceability capacity. However, for academic purposes, the tool as a whole was rather technical and not very didactic. As a result, a visual tool was also developed to complement *CLEMATIS*.

This tool was developed in *Python* with the help of the *Pygame* library, which extends the graphics framework of the language used. The pseudo-code of the implemented tool is shown below:

1) Read the Event Log;
2) Make an abstraction of the shape of the graph from the entries of transitions between machines;
3) Scroll through the data according to the *timestamp* and render the status of each machine;

3.4. Design of experiments and Process Mining

In order to check whether the results of the CLEMATIS improvements have really increased the level of detail of the event log, it was proposed to carry out experiments using process discovery algorithms, such as *Alpha Miner* and *Directly-Follows Graph*, and compliance analysis algorithms, such as *Token-Based Replay*.

This analysis is valid due to the fact that the level of detail of the event logs must be at least at the second level of detail for these process mining techniques to be applied.

In each section of the results and use cases, we'll delve into each of the algorithms presented and explain how they work and the motivation for using them.

4. Results and Use Case

With the changes made, it was possible to procedurally generate *event logs* like the one in Tab. 3.

Tabela 3. Updated CLEMATIS output example.

case	activity	time_stamp	time_stamp_out	product
1	16	2023-09-24 09:31:35	2023-09-24 09:53:35	1
1	6	2023-09-24 09:53:35	2023-09-24 10:32:35	1
1	2	2023-09-24 10:32:35	2023-09-24 11:10:35	1
1	10	2023-09-24 11:10:35	2023-09-24 11:35:35	1
1	4	2023-09-24 11:35:35	2023-09-24 12:01:35	1
2	16	2023-09-24 09:31:35	2023-09-24 10:05:35	2

The data obtained is visually at the event data level of detail, as we can see the activity, entry time and exit time for each case. But to prove that the model really does meet this level of detail, process discovery was carried out using the two main process mining methods: *Alpha Miner* and *Directly-Follows Graph*, both using the PM4PY library in Python. This discovery was made using as parameters a network of 20 nodes, 5 steps and a random seed of value 2 so that the data obtained can be reproduced.

4.1. Graphical Tool

The algorithm developed allows users to have a broader and deeper view of how the *CLEMATIS* simulation works, as seen in the Fig. 2.

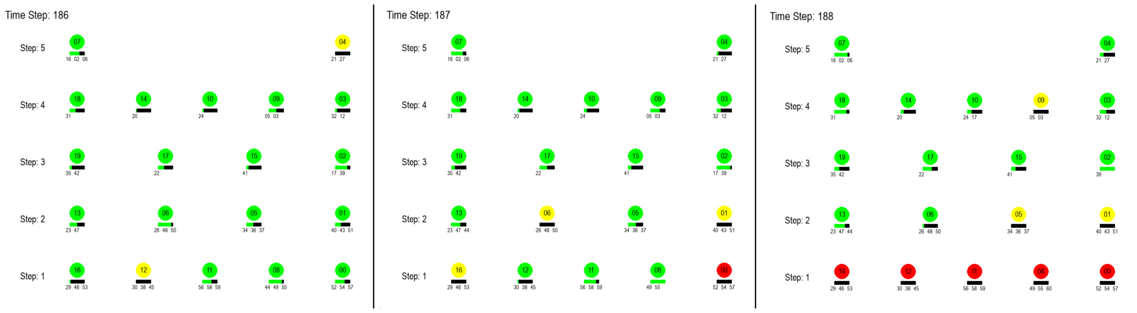


Figura 2. Visual tool example

With this tool it is possible to visualize for each production time the current status of each machine (green: working, red: blocked, yellow: failed to produce, grey: no resources), which resources are on each machine, and the production progress of each machine. It also gives a general idea of the distribution of the production line graph.

4.2. *Alpha Miner*

Alpha Miner is one of the most widely used algorithms for discovering process models from event logs. It was introduced by Wil van der Aalst in his research work on process discovery, one of the guiding works of the whole process mining area. This algorithm initially works by generating a Petri net, which is a graphical representation that describes the interactions between the activities of a process, as well as the conditions that must be met for the process to proceed. It consists of places, transitions, directed arcs and *tokens*.

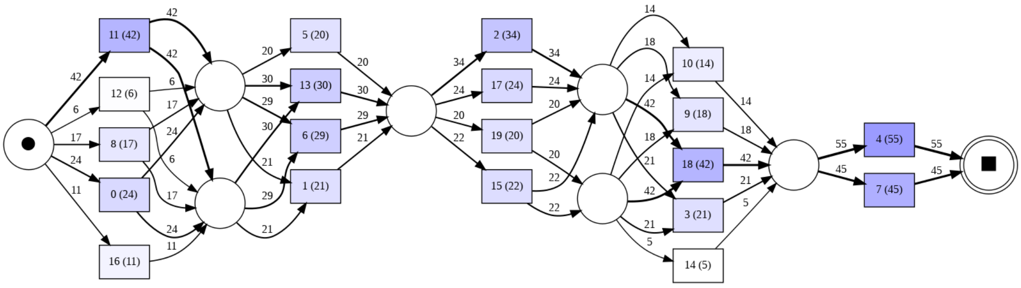


Figura 3. Graphical representation of process discovery using *Alpha Miner*.

In order to use the algorithm, the *event log* must have a level of detail in event data. Thus, when we apply the algorithm to the generated *Event Log*, we get the result shown in Figure 3, an uncovered Petri net with the transition edges showing the number of products that pass through them throughout the process. This shows that the proposed modifications have achieved their aim of improving the initial model.

4.3. *Directly-Follows Graph* (DFG)

Another important advance done in the improved model was that the execution frequency of each process on each node was constant, but to get closer to reality, a stochastic frequency was implemented, based on Poison variables. To evaluate this change, the DFG can be used, which is a process discovery method that evaluates the frequency of process transitions.

Using the Event Log generated, the DFG result shown in Fig. 4 was obtained, showing an uneven distribution of production times, reflecting the action of bottlenecks and demonstrating that the changes make the model closer to real data.

4.4. *Token-Based Replay*

For conformance analysis, the token-based replay was performed on the event log shown in the Tab. 3 for the case 1. Token based replay is a classic approach of evaluating the process discovery technique by its generated Petri-net. Through the placement of tokens after the recent activity in the neighbouring node and consumption of necessary tokens to execute the following activity, it is possible, by the end of simulation, to know the places where tokens are remaining or missing, which indicates the transitions where these

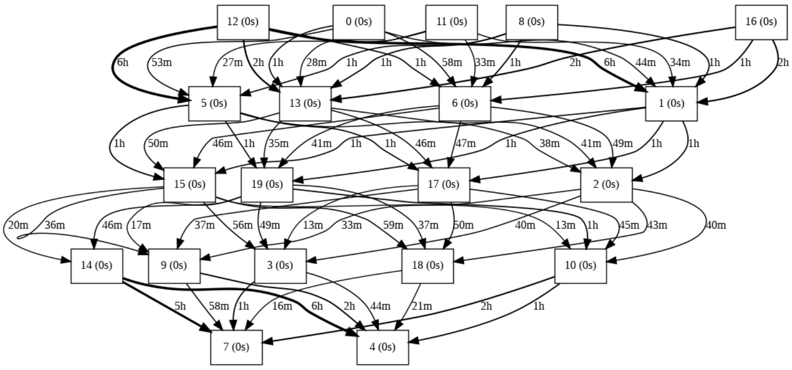


Figura 4. Graphical representation of process discovery using the DFG.

problems occurred. However, in this case study, the process evaluation is done over the event log generated by CLEMATIS rather than the process discovery technique, which is *Alpha Miner*.

Tabela 4. Token-Based Replay for Case 1.

Evaluation	Result
Trace Fitness	0.875
Trace is fit	False
Missing Tokens	2
Consumed Tokens	8
Remaining Tokens	0
Produced Tokens	6
Transitions with problems	16-6, 2-10

The results of the Tab. 4 is obtained through the token-based replay of the Case 1, using the Petri-net from Fig. 3 discovered by *Alpha Miner*. This is a sole example from 100 traces contained in event log. Initially, the results returned by this particular example shows the fitness of the event-log with the discovered Petri-net with no remarkable deviations. For the average result of 100 examples, the Tab. 5 summarizes the information regarding fitness, remaining and missing tokens:

Tabela 5. Token-Based Replay for all 100 cases.

Evaluation	Average Result
Trace Fitness	0.939
Trace is fit	0.25
Missing Tokens	0.69
Consumed Tokens	7.75
Remaining Tokens	0.25
Produced Tokens	7.31

The average fitness for the traces is 0.939, which express that the event log generator is capable of producing data which are sound and robust upon process discovery. The trace is only fit once the fitness equals 1, which are 25 % of the cases. Finally, the average quantity of missing and remaining tokens for each process case is lesser than 1, indicating that the application of process discovery techniques on the generated event log yields coherent results. These observations reinforce the replicability of event log generation by CLEMATIS and the adequacy of the model in generating data which are close to real world data.

5. Conclusion

The enhancements implemented in the CLEMATIS model significantly improve its applicability for both commercial and academic purposes. By increasing the granularity of data generated and integrating advanced process mining techniques, CLEMATIS now offers a more robust tool for analyzing production lines in the context of Industry 4.0. The successful incorporation of event data into the model allows for detailed and accurate process mining, enabling the identification of bottlenecks and optimization opportunities in manufacturing processes. Furthermore, the development of a visual simulation tool enhances user interaction, making the model more accessible and insightful, especially for educational purposes. Overall, this study not only enhances the technical capabilities of CLEMATIS but also broadens its scope of application, making it a valuable asset for the ongoing advancement of smart manufacturing systems.

Referências

Aalst, W. v. d. (2005). Business alignment: using process mining as a tool for delta analysis and conformance testing. *Requirements engineering*, 10:198–211.

Anderson, J. W., Kennedy, K. E., Ngo, L. B., Luckow, A., and Apon, A. W. (2014). Synthetic data generation for the internet of things. In *2014 IEEE International Conference on Big Data (Big Data)*, pages 171–176. IEEE.

El Emam, K., Mosquera, L., and Hoptroff, R. (2020). *Practical synthetic data generation: balancing privacy and the broad availability of data*. O'Reilly Media.

Fernández, M. and Williams, S. (2010). Closed-form expression for the poisson-binomial probability density function. *IEEE Transactions on Aerospace and Electronic Systems*, 46(2):803–817.

Fishman, G. S. (2001). *Discrete-event simulation: modeling, programming, and analysis*, volume 537. Springer.

Friederich, J., Jepsen, S. C., Lazarova-Molnar, S., and Worm, T. (2021). Requirements for data-driven reliability modeling and simulation of smart manufacturing systems. In *2021 Winter Simulation Conference (WSC)*, pages 1–12. IEEE.

Gutierrez, P., Luschkova, M., Cordier, A., Shukor, M., Schappert, M., and Dahmen, T. (2021). Synthetic training data generation for deep learning based quality inspection. In *Fifteenth International Conference on Quality Control by Artificial Vision*, volume 11794, pages 9–16. SPIE.

Hodapp, J., Schiemann, M., Arcidiacono, C. S., Reichenbach, M., and Bilous, V. (2020). Advances in automated generation of convolutional neural networks from synthetic data in industrial environments. In *HICSS*, pages 1–7.

Jouck, T. and Depaire, B. (2016). Ptandloggenerator: A generator for artificial event data. *BPM (Demos)*, 1789:23–27.

Letkowski, J. (2012). Applications of the poisson probability distribution. In *Proc. Acad. Business Res. Inst. Conf*, pages 1–11.

Libes, D., Lechevalier, D., and Jain, S. (2017). Issues in synthetic data generation for advanced manufacturing. In *2017 IEEE International Conference on Big Data (Big Data)*, pages 1746–1754. IEEE.

Lopes, P. V., Silveira, L., Guimaraes Aquino, R. D., Ribeiro, C. H., Skoogh, A., and Verri, F. A. N. (2024). Synthetic data generation for digital twins: enabling production systems analysis in the absence of data. *International Journal of Computer Integrated Manufacturing*, pages 1–18.

Manettas, C., Nikolakis, N., and Alexopoulos, K. (2021). Synthetic datasets for deep learning in computer-vision assisted tasks in manufacturing. *Procedia CIRP*, 103:237–242.

Nguyen, H. G., Habiboglu, R., and Franke, J. (2022). Enabling deep learning using synthetic data: A case study for the automotive wiring harness manufacturing. *Procedia CIRP*, 107:1263–1268.

Piga, D., Rufolo, M., Maroni, G., Mejari, M., and Forgione, M. (2024). Synthetic data generation for system identification: leveraging knowledge transfer from similar systems. *arXiv preprint arXiv:2403.05164*.

Schuitemaker, R. and Xu, X. (2020). Product traceability in manufacturing: A technical review. *Procedia CIRP*, 93:700–705.

Van Der Aalst, W. (2012). Process mining. *Communications of the ACM*, 55(8):76–83.

Van der Aalst, W., Weijters, T., and Maruster, L. (2004). Workflow mining: Discovering process models from event logs. *IEEE transactions on knowledge and data engineering*, 16(9):1128–1142.

Yang, H., Park, M., Cho, M., Song, M., and Kim, S. (2014). A system architecture for manufacturing process analysis based on big data and process mining techniques. In *2014 IEEE International Conference on Big Data (Big Data)*, pages 1024–1029. IEEE.