Streamlining Data Preparation for Discrete Event Simulations in Automobile Manufacturing: A Python-based Automation Tool

I. Saran Kumar
Student, Department of Mechanical Engineering
National Institute of Technology Calicut, India
ika_m210541me@nitc.ac.in

V. Madhusudanan Pillai
Professor, Department of Mechanical Engineering
National Institute of Technology Calicut, India
vmp@nitc.ac.in

Abstract

Discrete event simulations are becoming increasingly important in the management of complex manufacturing systems, however, a significant issue with current methods is the collection and processing of data from various physical systems. This data is often of poor quality and incomplete, making it difficult to produce accurate results. To address this problem, a Python-based automation tool was created to collect, analyse, and store data from Manufacturing Execution Systems (MES) using a multi-step data preparation algorithm and a dedicated simulation database. This tool is more efficient than using tools such as Microsoft Excel and includes a user-friendly interface for data entry and visualization. This automation tool is expected to improve the quality and accuracy of simulation results, while reducing the time and effort required for data preparation in the automobile manufacturing production line.

Keywords:
Discrete Event Simulation, Data Preparation, Data Processing, Manufacturing Execution Systems, Python-based Automation tool.

1. Introduction

Manufacturing is crucial to a society's prosperity and progress. To swiftly capitalize on emerging market trends, operations must constantly be kept as adaptable as possible. One of the essential elements of competitive success in industry is the requirement for quick adaptability to changes in industry trends or consumer patterns. Unfortunately, rather than making greater use of the equipment that already exists, this modification involves large investments. Obviously, reducing wastes like excessive production, delay, movements, defects, etc. is crucial for improving competitiveness. Continuous improvement of production flows, which is at the heart of Lean Manufacturing, will result in a reduction in the time it takes to complete an activity from start to end as well as overall expenses.

Discrete Event Simulation (DES) has shown to be a useful tool for such advancements. The use of DES reduces the a forementioned wastes by effectively planning, developing, and optimizing material flows in manufacturing. Nevertheless, despite its potential, the industrial sector has had trouble using simulation as a decision-support tool (McNally and Heavey 2004). The lengthy time commitment of simulation studies is one of the key drawbacks of using DES. This results in less thorough study and, as a result, production systems that ignore variation and disruptions and are built for perfect conditions. The input data procedure may be the cause of the lengthy time consumption. Typically, 10–40% of a DES project's total time is required for the input data procedure (Skoogh and Johansson 2008). This is the outcome of the human labor-intensive process of converting raw data into simulation input and the poor quality of the data (Robertson & Perera 2002).

Long data collecting stages are often suitable for independent simulation projects that are carried out over the course of many months (Robinson 2004). The use of DES in manufacturing, however, has significantly changed during the past ten years, moving from design of production systems to everyday operations. Operations and maintenance planning and scheduling are now frequent application areas, and real-time control is anticipated to be the next major sector (Negahban and Smith 2014).

There are additional restrictions on research into input data automation and simulation interoperability. This study begins when the essential information has already been gathered, located, and recognized. It therefore relies on the presumption that the data that have been obtained are already of good quality. A variety of inherently problematic aspects of the data gathering process are also a part of the multifaceted challenge of input data management to DES. Many authors have discussed these problems and drawn attention to issues with various aspects of the quality of the simulation data, such as accuracy, timeliness, and reputation (Balci et al. 2000). Some studies have also described issues with these aspects, such as missing data, restricted access to data sources, and poor data collection quality (Kuhnt and Wenzel 2010). However, actual studies demonstrating these issues with the quality of simulation data from the perspective of practitioners are still lacking.

The use of computer languages, notably Python, to automate the data preparation process for DES in the automotive industry has recently attracted a lot of attention. Python offers a number of benefits, including simplicity, adaptability, and the availability of a large number of libraries and data processing tools and visualization (Rossum and Boer 1991). In this study, we provide a Python-based automation solution for simplifying data preparation for DES in the automotive industry. The tool makes use of Python's strengths to streamline the data preparation procedure and minimize human labor, improving the simulation's effectiveness and accuracy. The tool has several features, including data preparation, visualization, and verification.

The contribution of this study is that it introduces a brand-new Python-based tool for speeding the data preparation process, illustrating its usefulness with a case study in a real-world vehicle manufacturing scenario based on the practitioner's perspective. The remainder of the paper is structured as follows: In Section 2, we go over the relevant literature on the DES data preparation process in the automobile industry. In Section 3, we describe the data collection process. We outline the methodology of the Python-based automation tool in Section 4. We discuss the results and findings in Section 5. Section 6 serves as the paper's conclusion and offers suggestions for the study's future steps.

Objectives

The purpose of this study is to present a method for collecting cleaned data for throughput simulation in the automotive sector by interfacing with Manufacturing Executive System (MES) to retrieve the necessary data based on user input and perform a complete purification process. The goals considered for the study are as follows:

- To create a tool that can get data from a database, incorporate data processing, and offer visualizations.
- To verify the output files against the output of the macros.

2. Literature Review

A possible automation of the data gathering process would be highly advantageous, according to (Robertson and Perera 2002), who recognized that the manual nature of the activity made data collection a very time-consuming operation. They provided descriptions of four potential approaches to entering the necessary data into DES projects. The manual stages in the first two techniques are mostly found in the gathering of simulation data, whereas the automated procedures in the latter two approaches. The third method relies on a DES model coupled to an intermediate database that is connected to Corporate Business System (CBS) applications (such as ERP, MRP, and CAD systems). The amount of human work and mistakes can be greatly reduced if the intermediate database interfaces with the CBS. The model directly obtains data from the CBS through an interface in the fourth technique. The tools are mostly produced in accordance with the third technique by looking at the literature.

The Ford Assembly Simulation Tool (FAST) is one example of an application that uses shop floor data from an Excel spreadsheet to create a WITNESS simulation model of a powertrain assembly line (Winnell and Ladbrook 2003). To create and import simulation data files into the Lanner WITNESS software, a collection of VBA macros are integrated with an Excel file. The FAST-transmitted file contains all the necessary characteristics to conduct simulation experiments. However, there are drawbacks to using spreadsheets, including their size, computing capacity, inability to create complicated models, and difficulties automating repetitive processes.

(Skoogh and Johansson 2008) developed a systematic approach for more time-efficient and accurate input data management for simulation projects based on 15 semi-structured interviews with simulation practitioners, with an emphasis on integrating all operations within input data management in an effective manner. Through the study of Survey results from 86 firm's experiences acquired during Winter Simulation Conference, (Skoogh, Perera et al. 2012) gave an update of the industrial practice in input data management in order to identify and explain a prospective advancement. (Skoogh, Johansson et al. 2012) introduced the Generic Data Management Tool (GDM-Tool) to enable automated data entry and analysis management. The GDM-Tool sends an interoperable file to a simulator in accordance with the Core Manufacturing Simulation Data (CMSD) requirements in addition to managing data. The GDM-Tool pulls information from several databases with diverse internal data structures, allowing for the reuse of database connections and pre-configured information from previously used simulation input data. A series of tiny plug-ins are linked together for execution, in order for the software to function. (Byrne et al. 2013) presented a software adapter that connects with data sources and/or fills data gaps using a data generator as part of their effort towards automated data collecting concentrating primarily on SMEs (Small and Medium-sized Enterprises). According to this study, SMEs will benefit from DES with less complexity, lower costs, and improved validity and accuracy of results thanks to the cloud-based solution.

It is evident from a review of the in-scope publications that automation of the input data process is the main area of scientific difficulty. For simulation modeler's, automation of the input data process would be a priceless asset. The creation of software that enables the automation of the input data management phases (data extraction, data processing, and data interface) will aid in the increased use of DES by both manufacturers and researchers (Barlas and Heavey, 2016b) . To solve the issue of significant time consumption and shorten the duration of simulation projects, (Barlas and Heavey 2016a) created a tool that includes capabilities that span the three stages of input data management. This tool uses the Python Rpy2 module.

(Bokrantz et al. 2018) contributed to better data quality in discrete event simulation (DES) in the manufacturing sector by offering 22 guidelines based on 6 cases and empirical evidence from semi-structured interviews, direct observations, records, and theoretical knowledge. The guidelines were backed up by empirical data from the cases. They highlighted the significant participation of simulation analysts in the data generation process as a crucial component of obtaining high-quality production data for simulation. In reality, every business is compelled to address big data quality issues given the realization of digitalized production in the future and the move towards employing DES for real-time management. As a result, providing high-quality data should be a prime concern in every manufacturing organization now and in the future.

Additionally, Python is a popular choice for data preparation in DES because of how frequently it is used in data science and machine learning. The present literature on the use of Python for automating data preparation in DES for vehicle production is minimal but promising. Using a real-time indoor localization system, a manufacturing simulation with real-time data parameterization and management of the real system through adaptations was introduced by (Mieth et al. 2019).

The literature study provides clear guidelines and methods for creating DES input data. Although there has been little study on using Python to create automation tools for DES input data, research on other input data management tools provides an overview of current developments and future directions. Another research gap is the tool's use of a specific database. This study expands on this assessment of the literature and suggests a Python-based automation tool for input data management in DES.

3. Data Collection

The Data used in the study is taken from a MES of an automobile manufacturer, the source of the data cannot be disclosed due to confidentiality. Simulation models continuously acquire real-time data from the physical system databases and subsequently the physical system takes the benefits from the outcomes of the simulation experiments. For data collection & cleansing process the simulation practitioners are using Excel macros which is time taking activity as the macro must run for every asset for different time periods and data quality is to be verified manually. So, a large amount of productive time is lost in preparing the data.

The database of the MES, consists event log data for every machine located at every manufacturing unit. So, an Application Programming Interface (API) based on python language is developed to fetch the required data from the server, then a unique cleaning algorithm prepared based on simulation input data requirements is applied. The result of this tool is a file suitable for simulation input data for throughput and bottleneck prediction which is stored in the simulation database. The working of Input Data Management (IDM) tool can be seen in Figure 1.

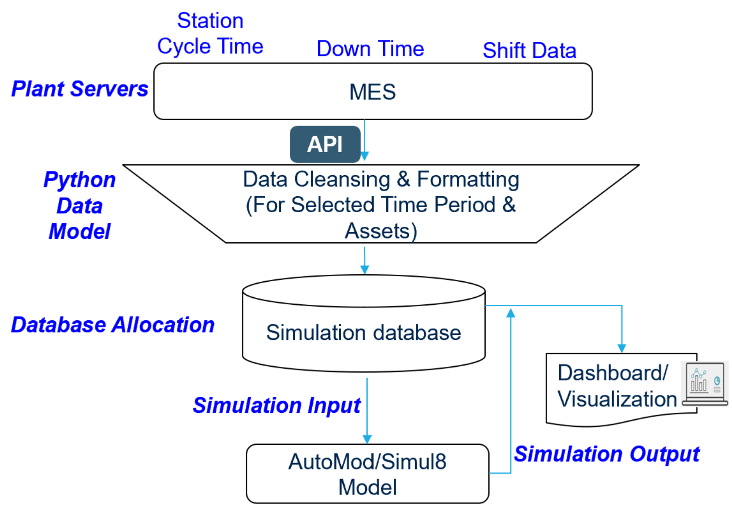


Figure 1. Flow chart of Input Data Management Tool

3.1 Fetching the raw data

A front end or user interface is developed to collect data from the user. The Figure 2 shows different data input sections like MES server name, Area key, Node ID or Asset details and date range. The selected data belongs to ABC server of Area One with three assets 185, 186 and 187 and date range from 1st July 2022 to 30th September 2022. The selected inputs will be used by API to establish connection with the server of that specific plant and pull the data into the simulation database. The same inputs of plant information and date range are used to fetch the shift pattern data.

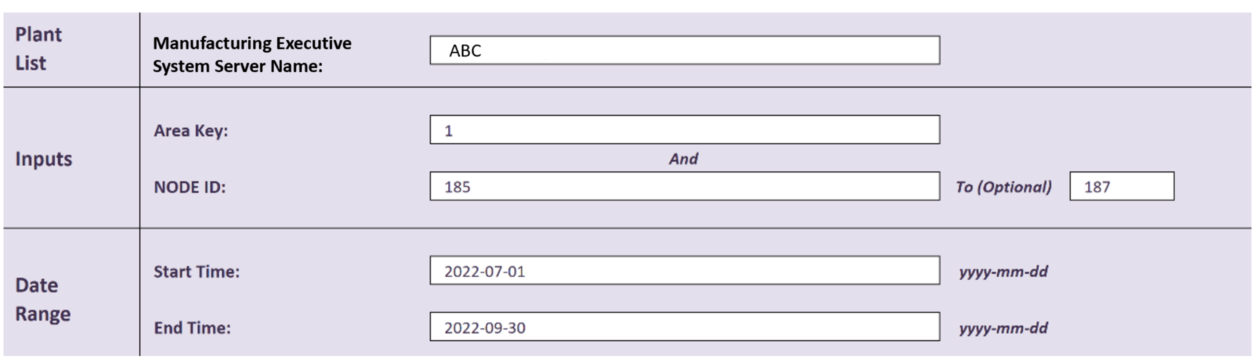


Figure 2. User interface for input collection

	AssetName	Description	StartTime	EndTime	Duration	Area	Station
0	OP20_1	Down	2022-07-01 06:22:38	2022-07-01 07:19:36	3418.0	1	185
1	OP20_1	Shutdown	2022-07-01 06:22:38	2022-07-01 06:22:39	1.0	1	185
2	OP20_1	Waiting Attention	2022-07-01 06:22:39	2022-07-01 06:22:42	3.0	1	185
3	OP20_1	error drives status class 1 __0_0_37	2022-07-01 06:22:39	2022-07-01 07:19:36	3417.0	1	185
4	OP20_1	Repair	2022-07-01 06:22:42	2022-07-01 07:19:36	3414.0	1	185
...	...	...	...	...	...	...	...
95119	OP20_3	Cycling	2022-09-29 21:37:04	2022-09-29 21:37:23	19.0	1	187
95120	OP20_3	Cycling Primary	2022-09-29 21:37:04	2022-09-29 21:37:23	19.0	1	187
95121	OP20_3	Shutdown	2022-09-29 21:37:23	2022-09-29 21:41:59	276.0	1	187
95122	OP20_3	Down	2022-09-29 21:37:23	2022-09-29 21:41:59	276.0	1	187
95123	OP20_3	No Comm	2022-09-29 21:41:59	2022-09-30 05:59:40	29861.0	1	187

95124 rows × 7 columns

Figure 3. Machine event log data

The Figure 3, talks briefly about the assets, various events and their corresponding start and end time with duration. A total of 95,124 rows pertaining to three assets with each asset consisting of three months of data. The Figure 4, is about various shifts, their corresponding start and end time with duration and whether it is productive or not. A total of 1238 rows of data for the three months period.

	ShiftName	PeriodName	StartTime	EndTime	Duration	IsProductive	Area
0	Shift 1 - Tue to Fri	NonProd	2022-07-01 06:00:00	2022-07-01 07:00:00	60	False	1
1	Shift 1 - Tue to Fri	Hour2	2022-07-01 07:00:00	2022-07-01 08:00:00	60	True	1
2	Shift 1 - Tue to Fri	Hour3	2022-07-01 08:00:00	2022-07-01 09:00:00	60	True	1
3	Shift 1 - Tue to Fri	Hour4	2022-07-01 09:00:00	2022-07-01 09:45:00	45	True	1
4	Shift 1 - Tue to Fri	Lunch	2022-07-01 09:45:00	2022-07-01 10:15:00	30	False	1
...	...	...	...	...	...	...	...
1233	Shift 2 - Afternoon shift	Lunch	2022-09-29 17:45:00	2022-09-29 18:15:00	30	False	1
1234	Shift 2 - Afternoon shift	Hour5	2022-09-29 18:15:00	2022-09-29 19:00:00	45	True	1
1235	Shift 2 - Afternoon shift	Hour6	2022-09-29 19:00:00	2022-09-29 20:00:00	60	True	1
1236	Shift 2 - Afternoon shift	Hour7	2022-09-29 20:00:00	2022-09-29 21:00:00	60	True	1
1237	Shift 2 - Afternoon shift	Hour8	2022-09-29 21:00:00	2022-09-29 22:00:00	60	True	1

1238 rows × 7 columns

Figure 4. Shift pattern data

4. Methodology

The cleaning algorithm is prepared in python language using the libraries Numpy, Pandas, Matplotlib, Date-Time and Seaborn. The Figure 5, describes the steps involved in preparing the data for throughput simulation. The cleaning algorithm is developed based on the discussions with the simulation experts and through analysis of the machine event log data. The outcome of the automation tool should contain only continuous events with no duplicates.

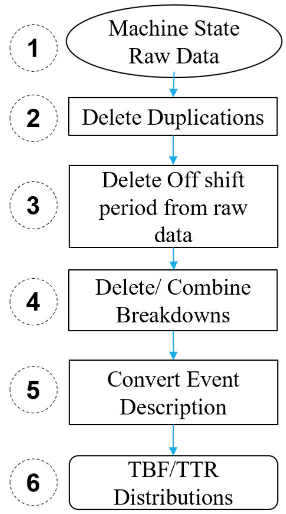


Figure 5. Flow chart of data preparation

First Step is collection of the machine event log and shift pattern data from the MES database. Then a row-by-row comparison of each entry is done to determine its type. As no production is carried out during the off-shift period, such entries will be removed. Further similar type of events will be combined and converted into three kinds of states. In detail, the analysis can be seen in the figures below. The final step is to extract Time Between Failure(TBF) and Time To Repair(TTR) distributions from the cleaned data.

The Figure 6, is about the count of status of data entry of the first asset obtained through data analysis of row-by-row comparison of start time and end time with previous and next rows. Out of 28574 rows about 13283 rows are duplicate entries and some have overlaps with the unique data. So, 13856 rows are unique and will undergo further cleaning. This non-duplicate data is compared with the shift pattern and wherever it is not productive and such timelines are discarded.

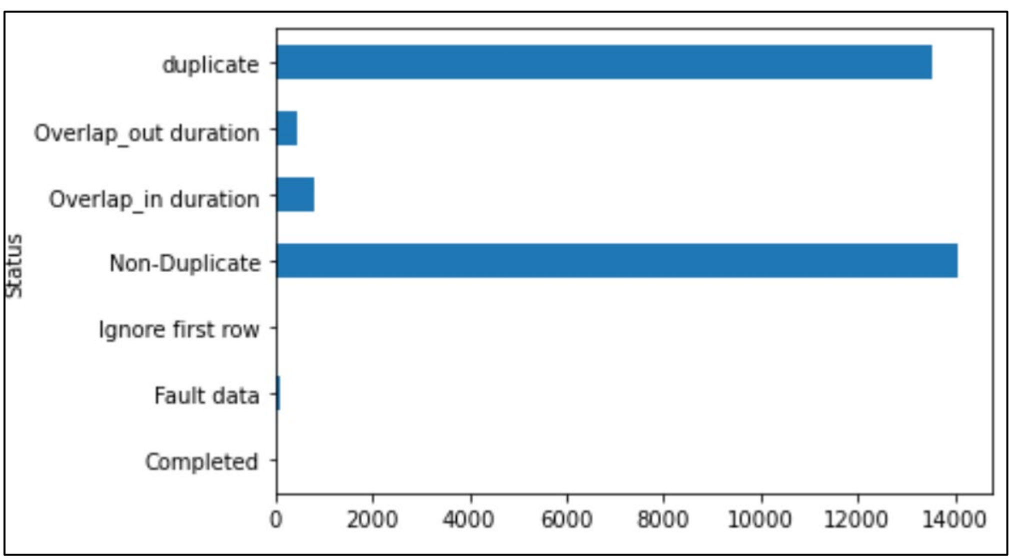


Figure 6. Data analysis of event log data

The fourth step is combining data of consecutive similar events. Next is converting events into three types as cycling, idle and down. Since each of these three events have other nomenclature in Event log data. The final step is to combine events in between two successive down states which is termed as TBF and all the down states will be considered as TTR.

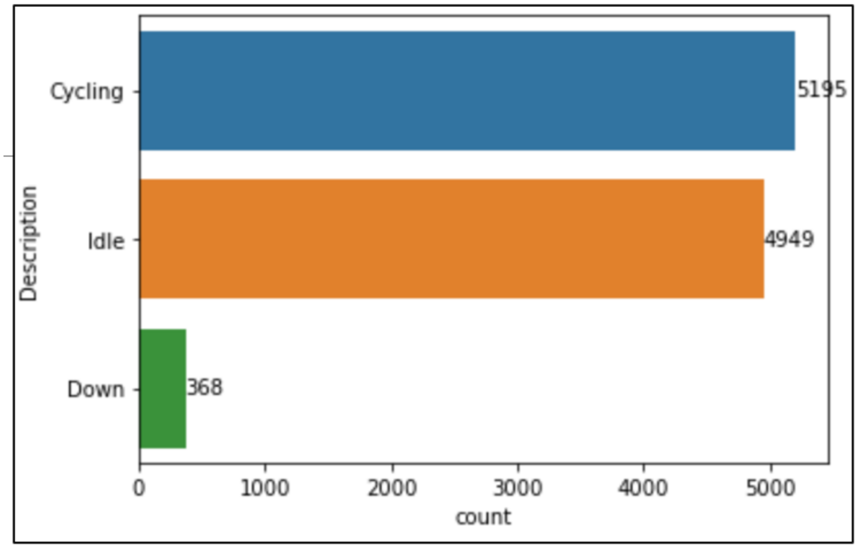
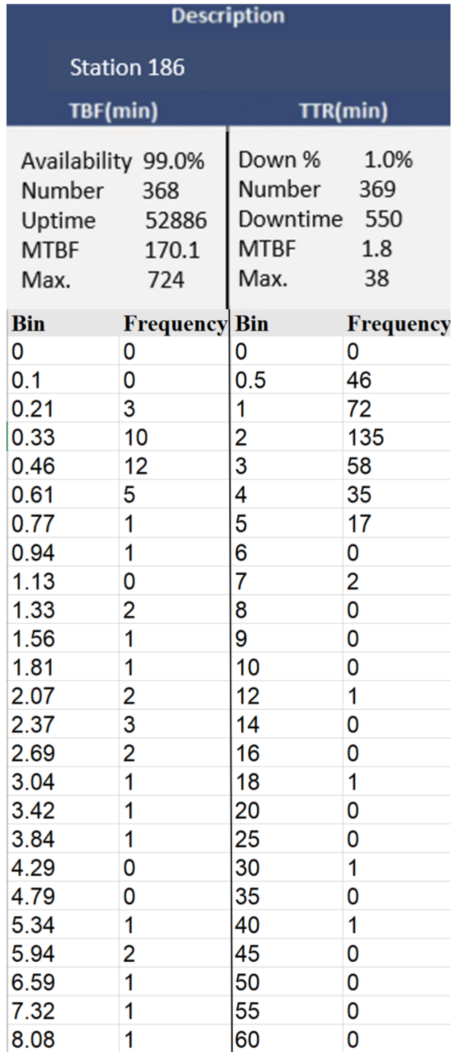


Figure 7. Count of events after complete data processing

The Figure.7, is about the count of three states after six steps of data processing of asset one. Here idle is a loading event which has count little less than cycling state and there are 368 down states.

4.1 Input data for throughput simulation

The below Figure 8, is the input data for the station 186 which will be used in simulation to generate breakdown pattern. These are obtained from the cleaned data where Time To Repair (TTR) durations in minutes corresponding to down states and Time Between Failure (TBF) durations in minutes corresponding to time between two successive down states. So, each and every station has this breakdown pattern which will be used as an input in simulation to run the model for several weeks and calculate throughput, bottleneck and other statistics regarding the line. This tables are the output of the IDM tool which is stored in the simulation database.



Description			
Station 186			
TBF(min)		TTR(min)	
Availability	99.0%	Down %	1.0%
Number	368	Number	369
Uptime	52886	Downtime	550
MTBF	170.1	MTBF	1.8
Max.	724	Max.	38
Bin	**Frequency**	**Bin**	**Frequency**
0	0	0	0
0.1	0	0.5	46
0.21	3	1	72
0.33	10	2	135
0.46	12	3	58
0.61	5	4	35
0.77	1	5	17
0.94	1	6	0
1.13	0	7	2
1.33	2	8	0
1.56	1	9	0
1.81	1	10	0
2.07	2	12	1
2.37	3	14	0
2.69	2	16	0
3.04	1	18	1
3.42	1	20	0
3.84	1	25	0
4.29	0	30	1
4.79	0	35	0
5.34	1	40	1
5.94	2	45	0
6.59	1	50	0
7.32	1	55	0
8.08	1	60	0

Figure 8. Screenshot of input data to simulation

5. Results and Discussion

The Table 1, shows the results of the summary of IDM tool data cleaning of 12 Assets for a time period of 3 months. The python based automation tool was able to complete this in 30 minutes which involves user input, data fetching, data cleaning and output file generation. Here Production time is the time during which the Assets are in cycling or loading state. The Down time is the non-productive time of the Asset. During the three months period the number of such Down events are counted for each asset and are shown in the Number of Downtime Occurrences column. Mean Time Between Failure (MTBF) and Mean Time To Repair (MTTR) are the averages corresponding to Productive time and Down time respectively for each Asset. In machine breakdown analysis, availability is a metric that measures the amount of time a machine or equipment is available for production or operation during a specified period. It is the ratio of the total time that a machine is operational to the total time it should be operational. Availability is expressed as a percentage and is a key performance indicator (KPI) for machine reliability and uptime.

Table 1. MTBF-MTTR Summary of python-based Input Data Management tool.

Asset Name	Productive time (min)	Down Time (min)	Number of Downtime Occurrences	MTBF (min)	MTTR (min)	Availability
A	29797.3	187.7	320	93.1	0.6	99.4%
B	29914.13	70.9	109	274.4	0.7	99.8%
C	29915.7	69.3	72	415.5	1.0	99.8%
D	29969.18	15.82	37	810.0	0.4	99.9%
E	29965.75	19.25	30	998.9	0.6	99.9%
F	29978.57	6.4	12	2498.2	0.5	100.0%
G	29975.78	9.22	21	1427.4	0.4	100.0%
H	29983.53	1.47	3	9994.5	0.5	100.0%
I	29967.17	17.8	112	267.6	0.2	99.9%
J	29955.35	29.7	153	195.8	0.2	99.9%
K	29563.32	421.68	458	64.5	0.9	98.6%
L	29953.2	31.8	172	174.1	0.2	99.9%

The Table 2, shows the results of the summary of Excel Macro tool used by the simulation practitioners for data cleaning of 12 Assets for a time period of 3 months. The tool was able to complete this in 300 minutes which involves user input, data fetching, data cleaning and output file generation.

These Occurrences difference effect the MTBF and MTTR values of each Asset. Also, there is negligible difference in the results of Availability column and the highest recorded is by the Asset-C that is 0.05%. This Validates that the IDM tool provides accurate results than the Macro tool with provision for interpretation and examination at step of data cleaning. The IDM tool only takes 10% of Excel Macro tool time to execute the same task with more precision and accuracy. Thereby helping simulation engineers to complete their projects in advance.

Table 2. MTBF-MTTR Summary of Excel Macro tool.

Asset Name	Productive time (min)	Down Time (min)	Number of Downtime Occurrences	MTBF (min)	MTTR (min)	Availability
A	29790.9	194.1	308	96.7	0.6	99.4%
B	29909.8	75.2	107	279.5	0.7	99.7%
C	29901.1	83.9	76	393.4	1.1	99.7%
D	29967.5	17.6	38	788.6	0.5	99.9%
E	29959.6	25.4	31	966.4	0.8	99.9%
F	29980.5	4.5	12	2498.4	0.4	100.0%
G	29975.4	9.7	20	1498.8	0.5	100.0%
H	29983.6	1.4	3	9994.5	0.5	100.0%
I	29966.4	18.7	109	274.9	0.2	99.9%
J	29953.5	31.5	146	205.2	0.2	99.9%
K	29558.7	426.3	449	65.8	0.9	98.6%
L	29944.9	40.1	167	179.3	0.2	99.9%

The Table 3, shows the differences between the results of the summary from IDM and Excel Macro tools of the 12 Assets. There are minimal differences in Occurrences as the IDM tool was able to clean the data in a better way without omitting or considering any events that are excluded in the off-shift periods, which sometimes the Macro tool have picked up.

Table 3. Differences in Summary from IDM and Excel Macro tools.

Asset Name	Occurrences Difference	MTBF Difference	MTTR Difference	Availability Difference
A	4%	-3.61	-0.04	0.02%
B	2%	-5.09	-0.05	0.01%
C	-5%	22.06	-0.14	0.05%
D	-3%	21.36	-0.03	0.01%
E	-3%	32.42	-0.18	0.02%
F	0%	-0.16	0.16	-0.01%
G	5%	-71.35	-0.04	0.00%
H	0%	-0.01	0.01	0.00%
I	3%	-7.36	-0.01	0.00%
J	5%	-9.37	-0.02	0.01%
K	2%	-1.28	-0.03	0.02%
L	3%	-5.16	-0.06	0.03%

6. Conclusions

We created an automation tool for IDM in DES that is based on Python which has a User Interface for data entry as well as a separate database for holding output files from the tool and the simulation software. Thereby establishing a workspace for simulation engineers that includes complete automation of the generation of input data for throughput simulation and the provision of data analysis visualizations with a single database housing all of their project-related documents. This tool significantly shortens the deadlines for simulation projects by giving useful input data in a matter of minutes, even when the tool is used for hundreds of Assets. In contrast, the Excel-based Macro tool generates the output over the course of hours or even days, and it occasionally gets stuck.

The results demonstrate that the Python IDM tool is significantly more effective at delivering precise output rapidly. The newly created cleaning algorithm is exclusive and solely applicable to the data creation for throughput simulation. Depending on the sort of simulation input data required and the type of data given for cleaning, this task can be expanded by altering the cleaning algorithm, as a future direction of research.

References:

Balci, O., Ormsby, W. F., & Carr, J. T. and Saadi, S.D., December. Planning for verification, validation, and accreditation of modeling and simulation applications. *Proceedings - Winter Simulation Conference,* vol. 1, pp. 829-839, 2000.

Barlas, P., & Heavey, C., Ke tool: An open source software for automated input data in discrete event simulation projects. *Proceedings - Winter Simulation Conference*, pp.472–483, 2016.

Barlas, P., & Heavey, C. 2016b. Automation of input data to discrete event simulation for manufacturing: A review. *International Journal of Modeling, Simulation, and Scientific Computing*, vol.*7*,no.1.

Bokrantz, J., Skoogh, A., Lämkull, D., Hanna, A., & Perera, T., Data quality problems in discrete event simulation of manufacturing operations. *Simulation*, vol.*94*,no.11, pp.1009–1025,2018.

Byrne, J., Byrne, P. J., Carvalho E Ferreira, D., & Ivers, A. M., Towards a cloud based SME data adapter for simulation modelling. *Proceedings of the 2013 Winter Simulation Conference - Simulation: Making Decisions in a Complex World, WSC 2013*, pp.147–158,2013.

Kuhnt, S., & Wenzel, Information acquisition for modelling and simulation of logistics networks. *Journal of Simulation*, vol.*4*,no.2, pp.109–115,2010.

McNally, P., & Heavey, C., Developing simulation as a desktop resource. *International Journal of Computer Integrated Manufacturing*, vol.*17*,no.5, pp.435–450,2004.

Mieth, C., Meyer, A., & Henke, M. 2019. Framework for the usage of data from real-time indoor localization systems to derive inputs for manufacturing simulation. *Procedia CIRP*, vol.*81*, pp.868–873.

Negahban, A., & Smith, J. S., Simulation for manufacturing system design and operation: Literature review and analysis. *Journal of Manufacturing Systems*, vol.*33*,no.2, pp.241–261,2014.

Robertson, N., & Perera, T., Automated data collection for simulation? *Simulation Practice and Theory*, vol.*9*,no.6–8, pp.349–364,2002.

Robinson, S., Simulation: The Practice of Model Development and Use. Inglaterra: John Willey and Sons. *Inc.*

Rossum, G. van, & Boer, J. de. 1991. Interactively Testing Remote Servers Using the Python Programming Language. In *CWI Quarterly* (pp. 283–303), 2004.
Skoogh, A., & Johansson, B., A methodology for input data management in discrete event simulation projects. *Proceedings - Winter Simulation Conference,* pp.1727–1735,2008.
Skoogh, A., Johansson, B., & Stahre, J. 2012. Automated input data management: Evaluation of a concept for reduced time consumption in discrete event simulation. *Simulation*, vol.*88*,no.11, pp.1279–1293.
Skoogh, A., Perera, T., & Johansson, B., Input data management in simulation - Industrial practices and future trends. *Simulation Modelling Practice and Theory*, vol.*29*, pp.181–192, 2012.
Winnell, Andrew, ladbrook, J., Towards composable simulation: Supporting the design of engine assembly lines. In *17th European Simulation Multiconference ESM,* pp. 9-11,2003.

Biographies

I. Saran Kumar is presently pursuing his final year of post-graduation in Industrial Engineering and Management from National Institute of Technology Calicut. Earned his graduation in Mechanical Engineering from RGUKT RKValley in 2020.

V. Madhusudanan Pillai, Professor, Department of Mechanical Engineering, National Institute of Technology Calicut, Kerala, India. With more than 30 years of teaching experience and over 150 publications in international journals and conferences, and editorship of a book, he has developed several laboratory exercises and software packages in the area of manufacturing management and supply chain management. His research interest includes modelling of problems in supply chain operation simulation, e-learning tools for supply chain operation simulation, sustainable supply chain management, cellular manufacturing systems, material requirements planning, scheduling, facility layout planning, inventory control, lean manufacturing, manpower planning – annualised hours, ergonomics and machine learning and blockchain applications in operations management. He supervised 10 Ph.D. theses, over 100 Master students and he is currently advising four Ph.D. scholars and three Master students on their research.