Proceedings of the 2014 Winter Simulation Conference
A. Tolk, S. Y. Diallo, I. O. Ryzhov, L. Yilmaz, S. Buckley, and J. A. Miller, eds.

BIG DATA IN DAILY MANUFACTURING OPERATIONS

Tim Wilschut
Ivo J.B.F. Adan

Department of Mechanical Engineering
Technical University Eindhoven
PO Box 513
Eindhoven, 5600 MB, THE NETHERLANDS

Joep Stokkermans

NXP Semiconductors
Industrial Technology & Engineering Center
Nijmegen, 6534 AE, THE NETHERLANDS

ABSTRACT

Big data analytics is at the brink of changing the landscape in NXP Semiconductors Back End manufacturing operations. Numerous IT tools, implemented over the last decade, collect gigabytes of data daily, though the potential value of this data still remains to be explored. In this paper, the software tool called Heads Up is presented. Heads Up intelligently scans, filters, and explores the data with use of simulation. The software provides real-time relevant information, which is of high value in daily, as well as long term, production management. The software tool has been introduced at the NXP high volume manufacturing plant GuangDong China, where it is about to shift the paradigm on manufacturing operations.

1 INTRODUCTION

"Knowledge provides power in many manufacturing contexts, enabling and facilitating the preservation of valuable heritage, new learning, solving intricate problems, creating core competencies and initiating new situations for both individuals and organizations now and in the future" (Choudhary, Harding, and Lin 2007). This quote captures the strategy set out by the NXP semiconductors Industrial Engineering and Technology Center (ITEC). In recent years numerous IT tools were implemented, which collect huge amounts of data at production facilities. This data is extensively analyzed during new product introductions or trouble shooting. However, the value potential of this data in steering every day production operations is yet to be exploited.

This paper presents a Greenfield project, which is the joint effort of NXP ITEC and the Manufacturing Networks research group of the Technical University Eindhoven, is presented. The goal of this project is to test and integrate the use of data mining tools in every day production operations, which is in line with the grand challenges in manufacturing (Fowler and Rose 2004). APG, NXPs Assembly Plant in GuangDong China, was selected as the pilot site. The APG factory produces millions of products per day, on a few thousand pieces of equipment, and it collects over 26 Gigabytes of data per day, which is equivalent to 100 paper Oxford dictionaries. Manually, scanning and filtering this data is too labour intensive. This has so far prohibited the use of this data in day-to-day decision making.

Due to the shear scale of APG operations it is difficult to get a grasp on "what is happening" inside the factory. Also due to economic migrations and China's industrial development policy, factories in China have a high people turn over, which limits opportunities of knowledge build-up and retention. This gives rise to consider new, more suitable production system concepts, like quantitative analysis tools to measure production efficiency and to point out hot spots. Moreover, these tools should facilitate day-to-day operational decision making, following rational thinking patterns. Based on smart metrics, these tools must give clues to basic operational questions like:

978-1-4799-7486-3/14/$31.00 ©2014 IEEE

- How is production going?
- Where is what attention required?
- Which action has the priority?

As the software tool Heads-Up must support basic user routines, it is being developed in close cooperation with the shop floor staff. In order to be of practical value, questions like above must be answered within minutes. Reporting functions like the first two questions are best answered by filtering production data on dedicated metrics. To address the third question, more specific intelligence must be added to the data. For priority setting in factory maintenance, a *fluid simulation model* is proposed to estimate productivity profits of repairs.

The software developed in this project is custom designed for the high volume production floor of APG. This floor houses a few dozen of production lines consisting of machines which are sequentially connected. Manually monitoring and processing manufacturing data of these few hundred machines is labor intensive and inefficient. Therefore, this floor was selected to implement an intelligent data mining tool.

The outline of this paper is as follows. First, the data collection system will be discussed in Section 2, after which the general concepts behind the software will be explained in Section 3. Next, from Section 4 to Section 5 the fluid flow simulation model, which is the main focus of this paper, is thoroughly explained. Finally, in Section 7 and Section 8 respectively, the preliminary results and future scope will be discussed.

2 DATA COLLECTION

Since 2001, NXP develops equipment status monitoring tools for the analysis of equipment performance. Equipment Status Monitoring is part of ITECs Advanced Warning And data Collection System environment, AWACS. It collects state events of the equipment with corresponding time stamp data. Equipment can be in one of the three aggregated states, Production, Down or Standby. In Figure 1, a context diagram shows the states and their respective sub-states. The state Production indicates that the machine is up and producing. The state Standby denotes that the machine could be producing, but it is not. The reason for being standby is indicated by one of the four sub-states. Sub-state wait-input means that the upstream buffer is empty and the machine is starved. In sub-state wait-output, the downstream buffer is full and the machine is blocked. The state Down indicates that the machine cannot produce. The reason for being down is indicated by one of the eight sub-states. The sub-state Error is also an aggregated state, containing hundreds of error types that can occur.

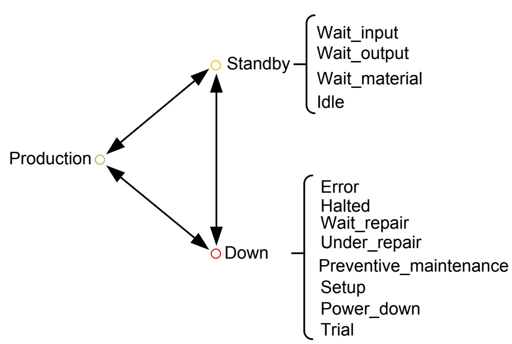

Figure 1: Machine states: Production, which indicates that the machine is producing; Standby, which indicates that the machine could be producing, but it is not; and Down, which indicates that the machine cannot produce. Both Standby state and Down state have sub-states, providing more detailed state information.

The time stamp data is used to construct error Pareto reports and to calculate production efficiency metrics, such as, e.g., Overall Equipment Efficiency (OEE)[1]. Via a web application, data of every piece of equipment within a factory is available.

In AWACS the full state information is gathered in a chronological event list. This event list is graphically displayed at the bottom line of Figure 2. A green line indicates that the machine is in the Production state; the dashed blue line indicates that the machine is Standby; a dashed red line indicates the Error state and the yellow line that the machine is Halted. On the shop floor, errors may not be immediately resolved, but typically after several attempts, or they may immediately induce other related errors. Hence, in the event list, patterns where an Error state is followed by a sequence of short Production, Halted and Error states often occur. In the simulation model, we treat the machine as *not* being in Production in between these closely following down states, but we *cluster these down times* into a single (long) down period. Such "effective" down periods are related to the concept of effective process times (Etman et al. 2011, Jansen et al. 2012). To explain the clustering and the calculation of up and down time realizations of a piece of equipment, we introduce the following notations:

1. Up Period (Up_i): Period of time during which no Error or Halted states occur.
2. Up Time (Upt_i): Sum of the durations of all Production states within an Up Period (so excluding durations of Standby states possibly occurring during this Up period).
3. Down Period (D_i): Period of time during which no Production state appears, which lasts longer than 60 seconds. This threshold value of 60 seconds is also applied in (Martono, Redlarski, and Kulikov 2010).
4. Down Time (Dt_i): Sum of the durations of all Error and Halted states within a Down period.

So down periods cluster multiple error states into a single long down time realization if these error states closely follow each other. The proper choice of the threshold value for what should be considered as "close" depends on the application at hand. In Martono, Redlarski, and Kulikov (2010), it is demonstrated that the simulated production throughput for the 60 second threshold value closely matches the throughput data. The clustering procedure is illustrated in Figure 2, where one can identify three up periods and two down periods. In clustering error states, we blame the error that is responsible for most of the down time during the cluster, as the one inducing the down period. By processing the entire list of events, a chronological list of up times and (clustered)) down times is obtained for each piece of equipment. These up and down times provide crucial input to the fluid simulation model, as will be explained in Section 4.

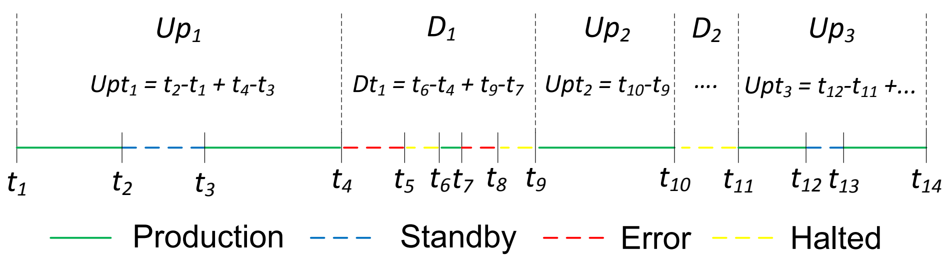

Figure 2: Schematic overview of up and down time calculation.

3 THE HEADS UP SOFTWARE

The main objective of this project is to use data analysis and simulation into day-to-day factory decision making. The software is developed around three key principles. First, different management levels require

[1]OEE Standardization SEMI E10-0301: Specification for definition and measurement of equipment reliability, availability, and maintainability; SEMI E79-0200: Standard for definition and measurement of equipment productivity.

different information. For example, a shop floor manager is not interested in detailed machine information, but a maintenance supervisor is. Second, the amount of data should be just fit for use to support the dedicated task, which is significantly less than the total data available. Hence, the software should intelligently scan and filter the data and only present those parts of the data, which are of interest to the user. Third, the user should have enough influence to customize the output. This is highly important for acceptance of the software by factory employees. Therefore, the tool should be flexible, adaptable and contain multiple levels of abstraction.

Around these key-principles the software tool is developed with a three level Graphical User Interface (GUI) between AWACS and the user. The tool is illustrated in Figure 3.

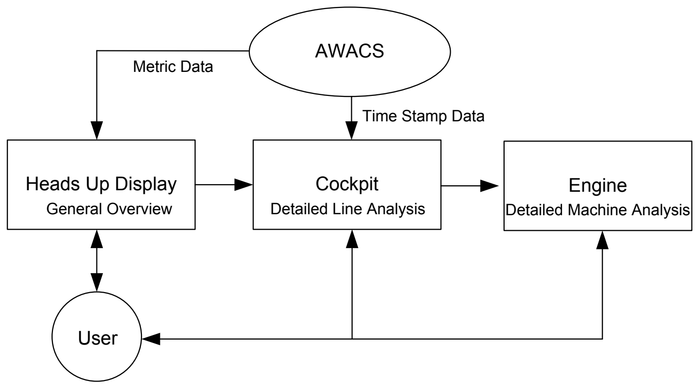

Figure 3: Three layered graphical user interface. The Heads Up Display provides production floor information. The Cockpit presents detailed line information. The Engine displays machine information.

The first layer, called the Heads Up Display (HUD), provides a general overview of the complete production floor. The target user group for HUD consists of production managers, quality managers and maintenance managers. The main goal of HUD is to provide quick insights and clues to “How is production doing?” and “Where is attention needed?”. In Figure 4 the HUD is shown. Production lines not performing well automatically turn red based on the Overall Equipment Efficiency (OEE) and counts of errors, which are critical to product quality. The OEE is defined as the realised throughput divided by the theoretically maximum throughput. The user can manually alter the OEE target, i.e., the OEE value below which a production line is considered to be insufficiently productive, and the error types, which need to be monitored by the system. Furthermore, hints are provided, which might help to understand why a production line is not performing well.

The second level of the GUI consists of the Cockpit view, which is shown in Figure 5. This view is designed as a high level line analysis tool for production and maintenance supervisors. It provides clues to the questions “What happened on this production line?” and “What should I do first?” State charts, Gant charts and trend graphs provide insight in the recent history of the equipment selected. For the second question, discrete event simulation is used. In the GUI a fluid flow simulation model is integrated. It looks for the errors that are responsible for most of the down time on that particular piece of equipment and then estimates the amount of throughput gain when these errors are solved. Hence, it helps the user to determine the error that is most critical to the throughput performance of the line.

The third level of the software consists of the Engine view displayed in Figure 6. This view displays equipment specific specialist information and focusses on maintenance engineers and technicians as a user group. This view also clearly indicates the effect of error clustering, which has been discussed in Section 2.

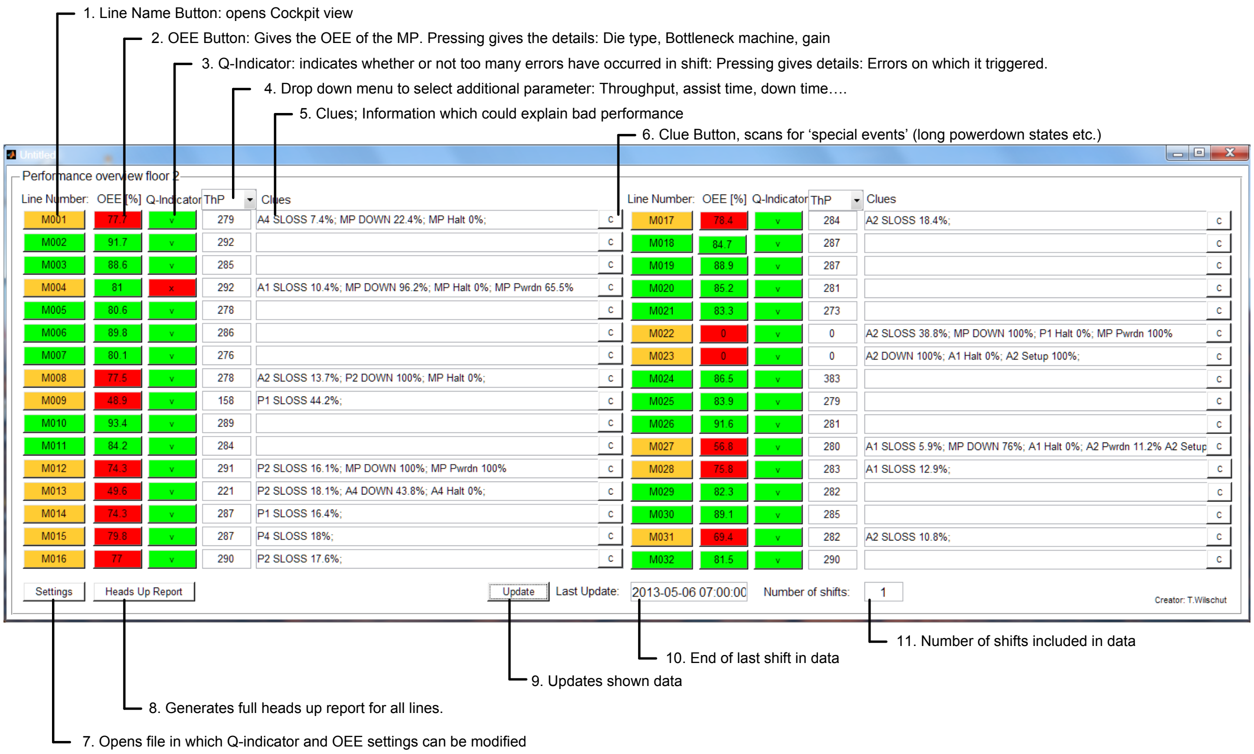

Figure 4: The Heads Up Display.

4 FLUID FLOW SIMULATION OF ASSEMBLY LINES

As highlighted above, a fluid flow simulator is embedded in the software to locate errors that are most critical to the line output. Of particular interest is the question: "What can be gained in terms of throughput if certain errors are solved?".

At APG, high volume semiconductor products are assembled on flow lines in a roll-to-roll process. The flow line consists in 9 machines with small buffers in between. At very high throughput, the buffer capacity is just enough to absorb several minutes of production.

The fluid flow simulator models the discrete flow of products through the assembly line as a continuous fluid flow. This modeling assumption is viable in this environment, because of the high rates: the products literally flow through the equipment. More granularity is not required for throughput calculations. Its power lays in simulation efficiency and speed to run "what-if"-scenarios is far less time than discrete product simulations. The simulation model carefully describes the interactions between the machines in the flow line, and the effect on overall line performance of system characteristics such as buffer capacities, machine production rates, and mean and variability in machine up and down times. As such, the simulator provides a powerful tool to support decision making in flow line configuration. Below we describe the (generic) fluid flow model of the production line in more detail.

The production line consists of N machines, labeled $1,\dots,N$. Note that we have $N=9$ in case of the assembly lines at APG. These machines are separated by finite buffers, labeled $i=2,..,N$. Buffer i with capacity K_i is in between machine $i-1$ and i. The flow through the machines is continuous, i.e., it is a fluid. Machine i produces at a maximum rate of s_i units per time unit, but it adjusts its production rate to the rate of machine $i-1$ when it depletes upstream buffer i, and to the rate of machine $i+1$ when it fills up downstream buffer $i+1$. Each machine suffers from breakdowns. When a breakdown occurs, the machine is taken into repair. During repair, the machine is not able to produce (i.e., it is down), possibly causing starvation of downstream machines and blocking of upstream machines. Successive up and down times

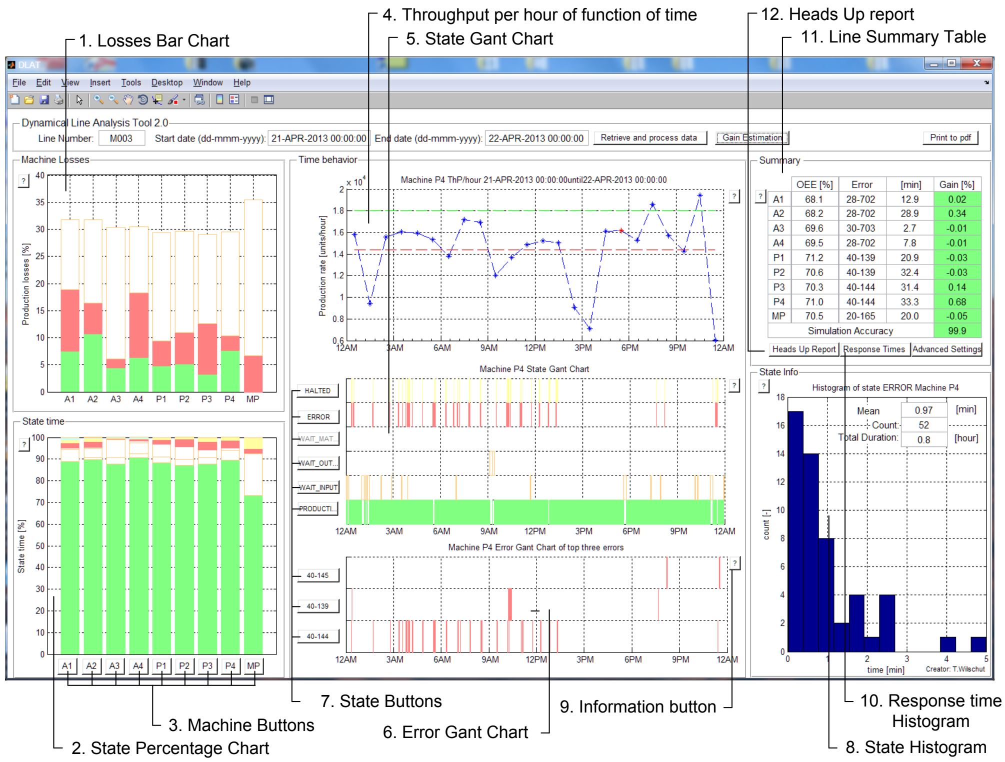

Figure 5: The Cockpit.

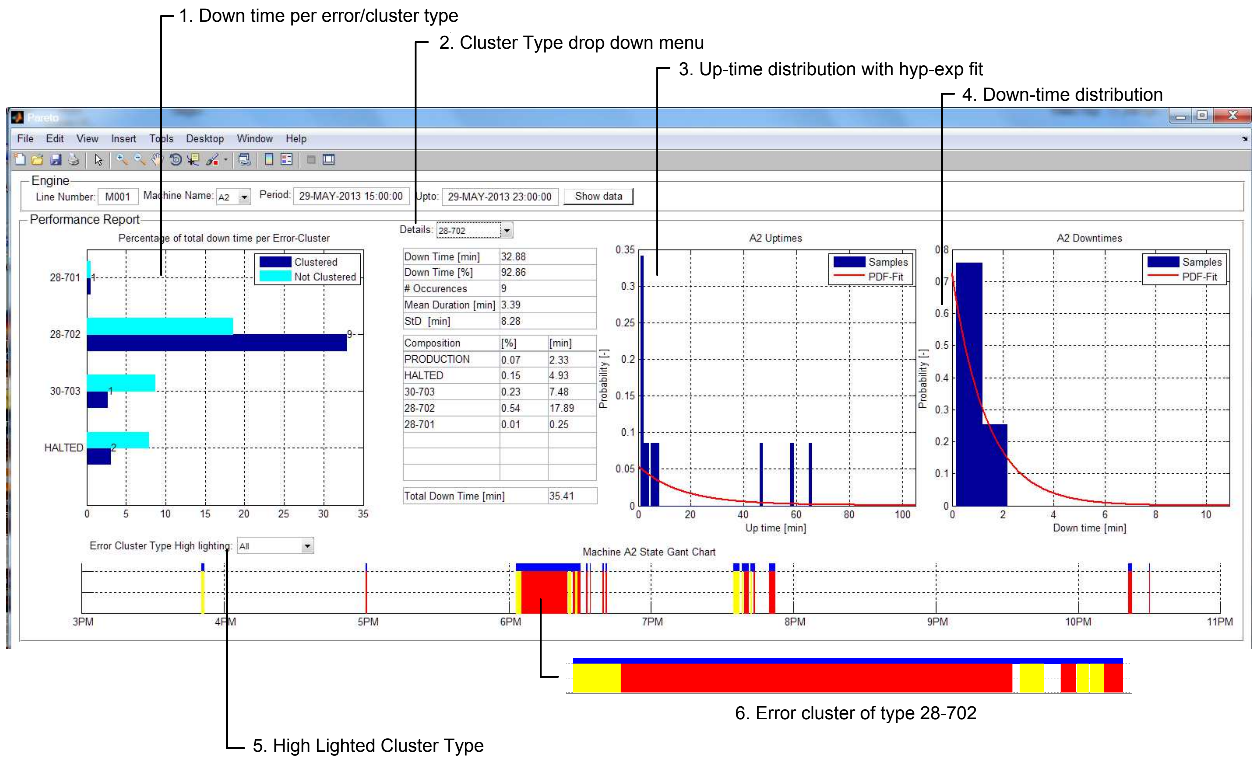

Figure 6: The Engine.

are assumed to be independent and generally distributed. Furthermore, the duration of the up times are assumed to be independent of the machine production rate and it is assumed that a machine cannot break down while it is not producing because of starvation or blocking, referred to as *operational dependent failures* (Buzacott and Shanthikumar 1993). Hence, during starvation or blocking, the up time of machine i is frozen and resumed at the point where it was interrupted when machine i continues to produce again. In the following subsections the behavior of the buffers and machines are described in more detail.

In this project we estimate the performance of the fluid flow model by means of discrete-event simulation, since this technique is flexible, efficient, transparent to users and easy to implement. It should be noted, however, that in the literature, efficient analytical approximation techniques are also available, see, e.g., Levantesi, Matta, and Tolio (2003), Bierbooms, Adan, and van Vuuren (2013). These approximations are typically based on decomposition techniques (Dallery and Gershwin 1992, Gershwin 1987) and sophisticated matrix-analytic methods (Latouche and Ramaswami 1999, da Silva Soares and Latouche 2006).

4.1 Buffer Behavior

Finite buffers have a large impact on the assembly line throughput. Therefore it is important to carefully incorporate their behavior into the simulation model. The state of buffer i is characterized by its content, denoted by B_i. Three aggregate states can be distinguished: Empty ($B_i = 0$), Neutral ($0 < B_i < K_i$) and Full ($B_i = K_i$). In case buffer i is Empty, $\dot{B}_i = 0$ and machine i has to slow down to the production rate of machine $i-1$, and vice versa if the buffer is full. In case the buffer is in state Neutral, it does not influence the productions rates of the surrounding machines, and the fill rate of buffer i is given by $\dot{B}_i = v_{i-1} - v_i$ where v_i denotes the *current* production rate of machine i (which can be less than the maximum rate s_i). Figure 7 graphically displays these relations. In the simulation model, transportation times of products through buffers are not included, implying that products produced at machine i are instantly available to machine $i+1$. This assumption is not relevant to the throughput, but it suggests that the simulated flow time will be less than the real flow time.

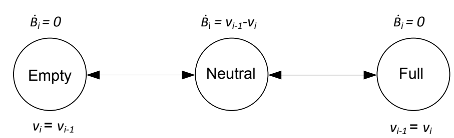

Figure 7: Schematic overview of the aggregate buffer states and their relation to the current production rate v_i.

4.2 Equipment Behavior

The event state model of the equipment was explained in Section 2. Machines can be in three states: Production (or Up), Down and Standby. The state Standby has two sub-states Wait-input and Wait-output, which stand for starvation and blocking, respectively. When machine i starts in the Production state, an up time is sampled. Subsequently, machine i stays in that Production state and produces at its current production rate v_i (which can be less than its maximum rate s_i) until the up time elapses or until a buffer event forces the machine to switch to Wait-input or Wait-output. An empty upstream buffer forces machine i to slow down (and possibly other down stream machines when more buffers are empty) or to switch to the Wait-input state (in which case the production rate v_i reduces to zero). A full downstream buffer forces machine i to slow down (and possibly other up stream machines when more buffers are full) or to switch to Wait-output. Note that when a machine slows down, it stays in the Production state. If a machine switches to the Wait-input or Wait-output state, its residual up time is frozen and it stays in that state until production is possible again. As the machine returns to the Production state, the machine continues to produce for the residual up time. When the up time elapses, the machine switches to the Error state, a down time is sampled and the machine stays in the Error state for that period of time. Next, it switches back to the Production state and the cycle is repeated.

4.3 Up and Down Time Distributions

To correctly simulate the production line behavior, the machine up times and down times are required. The calculation of up and down time realizations has been explained in Section 2. These realizations are plotted in a normalized histogram in Figure 8. In the simulation model, we might randomly sample up and down times from these up and down time realizations (thus also assuming that up and down times are independent). However, if the data collection period is short, then the number of up and down time realizations is rather small and possibly, not representative. Therefore, we decided to apply the following moment fitting approach. The empirical probability distributions of up and down times are approximated by fitting a probability density function (PDF) shown as the red line in Figure 8. The method adopted in this project is to fit phase-type distributions (Law and Kelton 2000) to the sample mean and sample variance obtained from the data set. Commonly used phase-type distributions are mixed Erlang distributions, or 2-phase Hyper-exponential distributions (Tijms 1988), depending on whether the sample coefficient of variation is less or greater than 1.

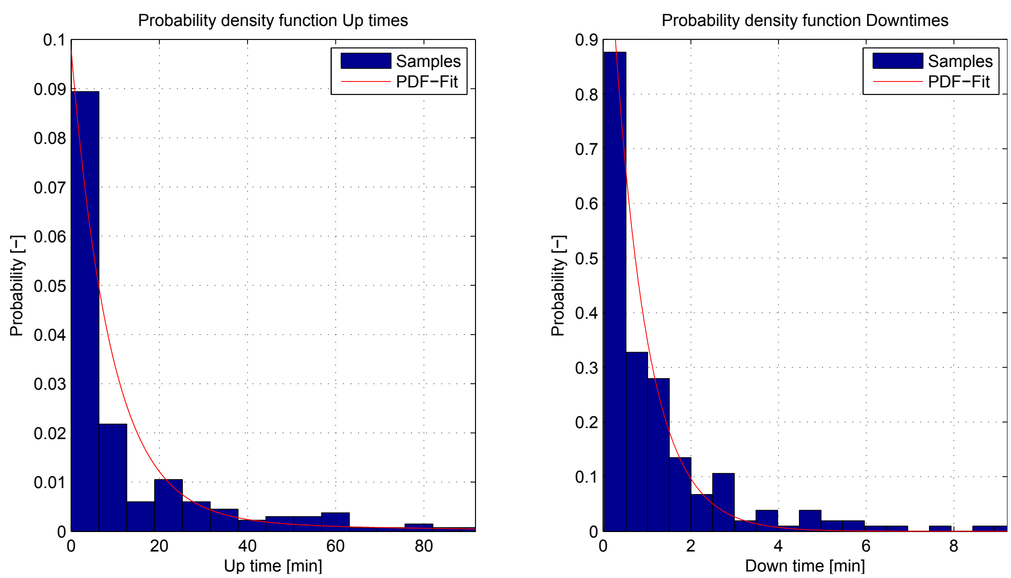

Figure 8: Normalized histograms of Up and Down times with 2-phase Hyper-exponential PDF fit.

In the simulation model we then sample up and down times from the fitted phase-type distributions, instead of the empirical distributions.

4.4 Production Rates

Besides up and down time distributions, the simulation model also needs the production rate s_i of each machine as input. The production rate of each machine is estimated by the average production rate during the sequence of up times obtained from the event list (i.e., the total production of the machine during the event list, divided by the sum of all up times of the event list).

5 VALIDATION

In this section we compare the simulation results to historical data. The performance measure of interest is the throughput. To validate the simulation model, we have done the following experiments. First 10 data periods of, for example, 3 shifts are selected. Then, for each data period, we calculate the simulation input parameters and set the simulation model to run for a total simulation time of 250000 hours. The first 5000 hours are used as warm-up period (and thus ignored), since the buffers are set to be initially empty. The resulting estimate of the throughput is then compared to the actual throughput measured over the corresponding data period. This procedure is followed for data periods of 1, 3, 6, 12 and 18 shifts. In Table 1, we list both the average and maximum percentage absolute error of the simulation estimate over the selected 10 data periods.

Table 1: Performance comparison between the simulated throughput and actual throughput.

	Absolute Error [%]				
#Shifts	1	3	6	12	18
Data Period 1	5.43	0.01	4.74	1.43	3.78
2	9.60	7.88	5.14	0.82	18.34
3	1.81	0.53	3.26	8.29	0.34
4	10.2	1.07	1.99	3.59	1.56
5	6.58	3.33	0.73	4.71	3.29
6	14.8	6.07	4.38	2.55	3.52
7	8.27	0.83	2.49	3.97	3.46
8	13.5	2.49	1.59	3.03	6.28
9	7.90	2.99	2.66	3.25	5.92
10	11.3	3.24	0.15	8.45	2.58
Average Error	8.93	2.84	2.72	4.01	4.91
Maximum Eror	14.8	7.88	5.14	8.45	18.3

Table 1 shows that on average, the simulation model accurately predicts the throughput. Simulations based on a single shift, however, are less accurate due to the fact that (i) the throughput is "more random" over short periods, and (ii) there are only few up and down time realizations available for fitting phase-type distributions. As a result, the fitted distributions may not reliably represent the machine up and down behavior. For longer data periods, it is seen that on average the simulated predictions are quite accurate. Though, there are some outliers with simulations errors three to four times as large as the average error. During these data periods, exceptional situations such as unusually long Powerdown or Halted states are observed. The simulation model cannot correctly cope with such situations. From a user perspective, this is not problematic: It is immediately evident that such situations need attention. The simulation model is intended to help the maintenance crew to focus their attention on the critical errors, in case there are no such eminent problems.

More validation results can be found in Kesuma and Martono (2010), Bierbooms (2012). Fluid flow simulation models have also been developed and validated for high throughput packaging lines at Heineken, where bottles and cans are filled with beer and processed, see Christian (2009). In this context, the simulator is used (i) to support decision making in line configuration, and (ii) for priority setting in maintenance operations.

An alternative to the proposed stochastic simulation method based on distribution fitting is a *deterministic data re-enactment method*. Instead of fitting phase-type distributions, one may simply use the chronological list of up and down time events as input to the simulation model. That is, the state events that occurred in real-life on a each machine, are, in chronological order, re-enacted in the simulation model.

6 GAIN CALCULATION

The main reason for implementing the simulator into the tool is to be able to do a "what-if" analysis. The main question is: "What is the output gain if a certain error type is solved?". To answer this question, the chronological list of up and down times is modified by assuming that it is possible to solve all errors of this specific type. For example, if this error type induces the down period D_1 in Figure 9, then all events occurring within this down period are removed, i.e., the entire error cluster is removed. This results in the event list graphically displayed by the top line in Figure 9. Consequently, a new up period Up_1' is created which has a longer up time realization. So, solving errors does not only eliminate down periods, it also makes up periods and up times longer. In this way we proceed through the whole event list and then obtain new up and down time distributions by re-fitting phase-type distributions on the modified up and down realizations.

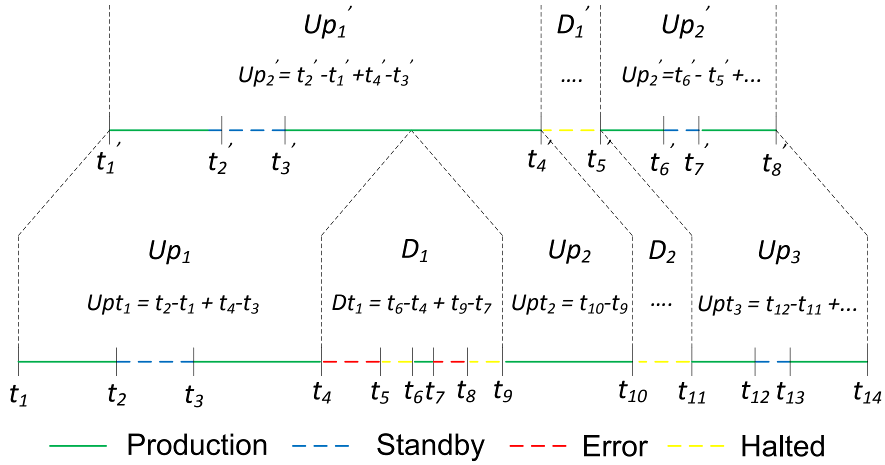

Figure 9: Clustering of errors: all states within the period t_4 until t_9 are considered to belong to the same down period D_1. The error that causes most of the downtime within $D1$ is said to be the cause of $D1$. In case of "what-if" analysis, all down periods related to a certain error type are removed and the pre- and proceeding up times are merged.

The gain is calculated by comparing the simulated throughput based on the *modified* up and down times to the simulation throughput based on the *original* up and down times. Table 2 shows the possible throughput gain for three different data sets of a single day (three shifts). Each data set is collected on the same production line.

The gain calculation process proceeds as follows. For each machine in the line, the error is determined that is responsible for most of the down time on that machine. Next, nine simulation runs are done, one run for each machine on which the dominant error is removed. The resulting throughput is then compared to the simulated throughput based on the original data in order to estimate the possible gain. Table 2 indicates which error type on which machine should be removed to obtain maximum gain in throughput. Clearly, the critical error and critical machine varies over the different days. One could also look at the effect of removing multiple error types on multiple machines, though those results are not presented here.

Table 2: Throughput gain calculation.

Date	17-04-13		19-04-13		28-04-13	
Machine	Error type	Gain [%]	Error type	Gain [%]	Error type	Gain [%]
1	28-702	0.11	40-702	0.02	28-552	0.00
2	30-709	0.18	40-702	0.06	30-703	0.00
3	30-34	0.07	40-703	0.05	30-703	0.14
4	28-702	0.22	40-703	0.04	30-703	0.00
5	40-144	0.14	40-145	0.04	40-351	0.00
6	40-145	0.05	40-356	0.61	40-135	1.91
7	40-359	0.04	40-135	1.10	40-144	1.93
8	40-139	1.74	40-356	0.88	40-144	0.00
9	20-165	0.06	20-177	0.16	20-177	0.00

7 CONCLUSIONS

The Heads Up software is a data mining tool with an integrated fluid flow simulation model to do "What-if" analysis. The fluid flow simulation model developed by Kesuma et al. (Kesuma and Martono 2010) can accurately simulate the production line behavior of NXP assembly lines. The combination of data mining and increased intelligence by use of simulation has proven to be an effective and valuable tool within daily manufacturing operations. It enables the factory maintenance crew to better focus their attention and set task priorities. Preliminary results show that the NXP assembly plant in GuangDong China has gained of few percent in Overall Equipment Efficiency since the implementation of the software.

8 FUTURE WORK

The next step in the development of the Heads Up tool is to allow users to create "mini-companies", that is, provide the users with the ability to create specialized views which focus on a small section of the factory or on a single product type. By doing so each factory employer can create a view in which all relevant information with respect to their responsibilities is shown.

The Heads Up software has been primarily designed as decision support system, similar to the work of Zhou et al. (2005). However, in the future, it may be extended with data mining algorithms for job shop scheduling (Kwak and Yih 2004), quality control (Rokach and Maimon 2006), fault diagnostics (Lee and Ng 2006) and condition based maintenance (Hou, Liu, and Lin 2003).

Furthermore, for more general data mining applications, there are possibilities for product traceability enhancement by integration of batch assembly data with shop floor control systems; searching for correlations between product deficiencies and machines errors by integration of batch test data and assembly data; and enhancement of the SAP Production Maintenance module by adding root-cause information to errors.

REFERENCES

Bierbooms, R. 2012. *Performance analysis of production lines*. Department of Mathematics and Computer Science, Eindhoven University of Technology.

Bierbooms, R., I. Adan, and M. van Vuuren. 2013. "Approximate performance analysis of production lines with continuous material flows and finite buffers". *Stochastic Models* 29 (1): 1–30.

Buzacott, J., and J. Shanthikumar. 1993. *Stochastic Models of Manufacturing Systems*. Prentice Hall.

Choudhary, A., J. Harding, and H. Lin. 2007. "Engineering moderator to universal knowledge moderator for moderating collaborative projects". *Global Journal of e-Business & Knowledge Management* 3 (1): 5–12.

Christian, E. 2009. *Packaging Line Performance Analysis and Optimization in a Simulation-Based Environment*. Department of Mathematics and Computer Science, Eindhoven University of Technology.

da Silva Soares, A., and G. Latouche. 2006. "Matrix-analytic methods for fluid queues with finite buffers". *Performance Evaluation* 63 (4): 295–314.

Dallery, Y., and S. Gershwin. 1992. "Manufacturing flow line systems: a review of models and analytical results". *Queueing Systems* 12 (1-2): 3–94.

Etman, L., C. Veeger, E. Lefeber, I. Adan, and J. Rooda. 2011. "Aggregate modelling of semiconductor equipment using effective process times". In *Proceedings of the 2011 Winter Simulation Conference*, edited by S. Jain, R. R. Creasey, J. Himmelspach, K. P. White, and M. C. Fu, 1795–1807. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Fowler, J., and O. Rose. 2004. "Grand challenges in modeling and simulation of complex manufacturing systems". *SIMULATION* 80 (9): 469–476.

Gershwin, S. 1987. "An efficient decomposition algorithm for the approximate evaluation of tandem queues with finite storage space and blocking". *Operations Research* 35 (2): 529–537.

Hou, T., W. Liu, and L. Lin. 2003. "Intelligent remote monitoring and diagnosis of manufacturing process using an integrate approach of neural networks and rough sets.". *Journal of Intelligent Manufacturing* 14 (2): 239–253.

Jansen, F., L. Etman, J. Rooda, and I. Adan. 2012. "Aggregate simulation modeling of an MRI department using effective process times". In *Proceedings of the 2012 Winter Simulation Conference*, edited by C. Laroque, J. Himmelspach, R. Pasupathy, O. Rose, and A. M. Uhrmacher, 919–930. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Kesuma, G., and R. Martono. 2010. *Dynamic Line Modeling*. Department of Mathematics and Computer Science, Eindhoven University of Technology.

Kwak, C., and Y. Yih. 2004. "Data mining approach to production control in the computer integrated testing cell.". *IEEE Transactions on Robotics and Automation* 20(1):107–116.

Latouche, G., and V. Ramaswami. 1999. *Introduction to matrix analytic methods in stochastic modeling*, Volume 5. ASA-SIAM Series on Statistics and Applied Probability.

Law, A., and W. Kelton. 2000. *Simulation modeling and analysis*. McGraw-Hill New York.

Lee, S., and Y. Ng. 2006. "Hybrid case-based reasoning for on-line product fault diagnosis.". *International Journal of Advanced Manufacturing Technology* 27 (7-8): 823–840.

Levantesi, R., A. Matta, and T. Tolio. 2003. "Performance evaluation of continuous production lines with machines having different processing times and multiple failure modes". *Performance Evaluation* 51 (2): 247–268.

Martono, R., L. Redlarski, and P. Kulikov. 2010. *Clustering Effect for errors in a Dynamic Line*. Department of Mathematics and Computer Science, Eindhoven University of Technology.

Rokach, L., and O. Maimon. 2006. "Data mining for improving the quality of manufacturing: A feature set decomposition approach.". *Journal of Intelligent Manufacturing* 17 (3): 285–299.

Tijms, H. 1988. *Stochastic modelling and analysis: A computational approach*. John Wiley & Sons, New York.

Zhou, J., X. Li, A. Andernroomer, H. Zeng, K. Goh, Y. Wong, and G. Hong. 2005. "Intelligent prediction monitoring system for predictive maintenance in manufacturing". In *Proceedings of the 32nd Annual Conference of the IEEE Industrial Electronics Society*, 2314–2319.

AUTHOR BIOGRAPHIES

TIM WILSCHUT is a Ph.D. candidate of the Manufacturing Networks group in the Department of Mechanical Engineering at the Eindhoven University of Technology. He holds a M.S. in Mechanical Engineering from the Eindhoven University of technology. His email address is t.wilschut@student.tue.nl.

IVO ADAN is Full Professor of the Manufacturing Networks group in the Department of Mechanical Engineering at the Eindhoven University of Technology. He holds a M.S. and Ph.D. in Mathematics from the Eindhoven University Technology. His current research interests include the analysis of multi-dimensional Markov processes and queueing models, and the performance evaluation of production and warehousing systems. His email address is iadan@tue.nl.

JOEP STOKKERMANS is a development manager at NXP semiconductors. He holds a M.S. in mechatronics from the Delft University of Technology. His email address is joep.stokkermans@nxp.com.