Proceedings of the 2007 Winter Simulation Conference
S. G. Henderson, B. Biller, M.-H. Hsieh, J. Shortle, J. D. Tew, and R. R. Barton, eds.

SENSITIVITY ANALYSIS ON CAUSAL EVENTS OF WIP BUBBLES BY A LOG-DRIVEN SIMULATOR

Ryo Hirade
Rudy Raymond
Hiroyuki Okano

IBM Research, Tokyo Research Laboratory
1623-14 Shimo-tsuruma, Yamato
Kanagawa, 242-8502, JAPAN

ABSTRACT

Fluctuations of work-in-progress (WIP) levels cause variability of cycle time and often lead to productivity losses in semiconductor wafer fabrication plants. To identify sources of such variability, we are developing a root cause analysis tool with history logs of operational events, such as high WIP or equipment downtime, as inputs to automatically find the chains of events that create the variability. In the root cause analysis, one of the key steps is to aggregate the observed events into groups that are likely in cause-effect relationships. For operational events that involve time lags in cause-effect relationships, grouping the events requires identification of the time windows of causality based on discrete event simulations. This paper describes a design and implementation of a simulator for this purpose. The simulator does not assume any statistical or mathematical model, and thus is simple to maintain.

1 INTRODUCTION

Rapid development of new products, quick delivery, and high productivity are essential for semiconductor wafer fabrication plants (fabs) to win against severe market competition. Throughput, cycle time, and yields are some of the metrics of productivity. With regards to cycle time, the productivity can be improved by reducing the cycle time variability. The reduction of cycle time variability can be done by controlling lot releases, rescheduling equipment maintenance, and so on, with relatively small costs. However, finding efficient actions to reduce variability is not easy due to the high complexity of manufacturing processes involving hundreds of pieces of equipment. Our approach here is to focus on particular sources of cycle time variability, namely, *work-in-progress* (WIP) bubbles and holes, and to develop an intelligent tool that finds chains of related causal factors causing the variability. (WIP bubbles and holes mean high and low numbers of lots, respectively.) Prior study shows that the cycle time varies greatly as the number of WIP bubbles becomes higher (Hassoun, Rabinowitz, and Lachs 2005). Intuitively, the following Little's Law states that at a given throughput level, cycle time becomes longer when WIP bubbles occur (Little 1961):

$$WIP = Cycle\ time \times Throughput.$$

WIP levels are affected by transient capacity losses. Table 1 shows 22 capacity loss factors in a fab reported by Robinson, Fowler, and Neacy (2003) based on surveys and interviewing experts. It is believed that WIP bubbles are triggered by a combination of the above factors in which a causal factor, though not directly inducing WIP bubbles, might trigger other events and thus be considered as the root cause. Thus, mitigating the root causes can efficiently minimize the variability. In an actual fab, it often takes hours for production control engineers to analyze root causes of WIP bubbles. Our ultimate goal is to develop a *root cause analysis* (RCA) tool that will help reduce this time.

One of the main steps in RCA is to create a *causal factor graph*. Figure 1 is an example of such a graph as created by hand. In our application, each node in a causal factor graph is represented by an *operational event*, such as a WIP level, the number of high priority lots, scheduled/unscheduled downtime of equipment, and so on. Nodes are linked if they are likely to have cause-effect relationships. Our RCA tool is intended to have a function to automatically create such causal factor graphs from a fab's history logs.

This paper describes a technique to identify which operational events in a fab affect other events during certain *time windows*. Operational events affect each other due to cause-effect relationships. Given the realization of such events in history logs, for events that change or occur simultaneously, we can identify groups of events by computing probabilities that they occur together in fixed time buckets. However, the accuracy of the statistical approach is in general low due to the interactions between events, including those which

Table 1: Capacity loss factors in semiconductor manufacturing as reported by Robinson, Fowler, and Neacy (2003)

Time constraints between steps	Preventive maintenance	Factory shutdown	Product mix	Yield
Dispatching / sequencing	Hot / engineering lots	Tool dedication	Shift plans	Setup
Lack of tool redundancy	Operator availability	Batching policy	Inspection	
Unscheduled maintenance	WIP control strategy	Reentrant flow	Lot size	
Operator cross-training	End-of-shift effect	Order release	Rework	

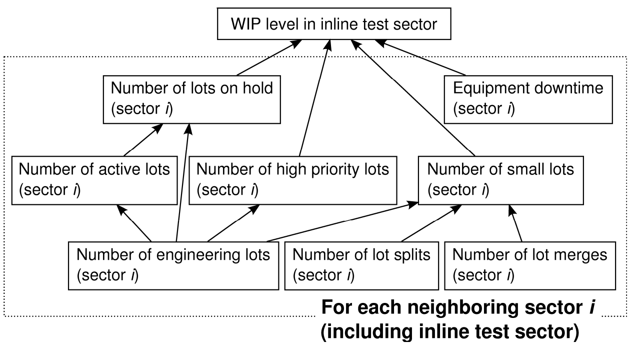

Figure 1: An example of a causal factor graph.

might be independent of the target events. For events that are affected by others after some period of time and that return to normal states after some time (*time-lag events*), in order to understand which events occur together, we need to know the time window of the causality by specifying the start time when the influence of one event begins and the end time when its influence on other events vanishes. Such a time window may not be obvious from a priori knowledge of the target plant system, or even from the statistical method because of the noises caused by interactions among many events.

Our approach for identifying correlated events and for finding the time windows of causality is to use *discrete event simulation* (DES). The merits of using DES against other methods based on statistical approaches are that

- it can perform sensitivity analysis while avoiding interactions from unrelated events, and
- it can measure the time lags between events.

DES in general requires a precise model of the target system which needs to be carefully recalibrated until it can reproduce what happens in reality. We propose in this paper a *log-driven* simulator which can traverse the history logs of events to exactly reproduce what happened in reality. The proposed simulator can also simulate the history logs along with additional events, such as a new unscheduled downtime or a lot hold.

The aims of using a log-driven simulator are to understand which pairs of events correlate with each other based on sensitivity analysis, in order to identify the time windows in which time-lag events are correlated, and to output groups of observed data (event values) for the correlated events. When correlated events are identified as in Figure

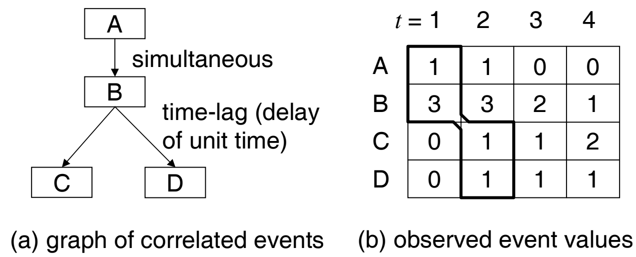

Figure 2: Examples of the graph of correlated events and corresponding output of event values.

2 (a) and the event values are observed as in Figure 2 (b), the event values for events A, B, C and D are grouped into $(1,3,1,1)$ and $(0,2,2,1)$. The bold bordered enclosed area in Figure 2 (b) represents a group of event values that are correlated. Note that the enclosed area covers event values at different time.

This paper is organized as follows. Section 2 provides an overview of root cause analysis. Section 3 describes the design and implementation of our log-driven simulator which does not assume any statistical or mathematical model, and thus is relatively simple to maintain. Section 4 gives a numerical study of the log-driven simulator. We summarize our paper with a brief conclusion in Section 5.

2 ROOT CAUSE ANALYSIS

2.1 Overview

In general, RCA is defined as a structured procedure to guide an analyst in finding the underlying root causes of events or failures, such as WIP bubbles or holes in fabs. Many arguments exist about the definition of root causes, however, they can simply be taken as answers to explain why certain events happened instead of how it happened or what happened. Root causes are underlying causes that can be reasonably identified, effectively controlled, and mitigated (Rooney and Heuvel 2004).

RCA consists of the following main steps:

1. collect data to list all of the events related to the events of interest,
2. verify the relationships among the events and generate a graph of correlated events,
3. create a causal factor graph,
4. identify the root causes,

5. list priority of actions for mitigating the effect of these root causes.

There are many tools to perform the above steps in RCA. Five-why and Pareto analyses are some of them which are popular in manufacturing. In the most basic Five-why analysis, questions are asked about events, starting from the events of interest or failures, iteratively until the analyst thinks he has found the underlying root causes. For example, in Figure 1, the first why is asked about the events of *inline test sector had high WIP*. Assume that the answer is *the number of lots on hold is high in the previous sector*. Then the next why is also applied to this answer and so on until the analyst finds that a *high number of engineering lots* is one of the possible root causes. (Of course, he can also proceed to ask why for this event but let us assume that the analyst stops at this answer.) It can be seen immediately from this example that the Five-why analysis has some weak points. It heavily depends on the analyst's capability to ask the right questions in addition to finding the appropriate answers. Moreover, different analysts might end up finding different root causes simply because there are multiple causes, especially in fabs whose operations are very complex. In such case, the Pareto analysis can be used to treat the most frequent candidates as the root causes. However, many analysts are concerned that the events that appear most frequently might just be immediate symptoms rather than root causes.

To address such weaknesses, Weidl, Madsen, and Israelson (2005) proposed a graphical model, a method combining statistics and graph theory, for RCA and decision support, which is applicable to industrial processes. However, it still depends on the judgment of experts to infer the relationships between events (i.e., drawing the edges in Figure 1).

On the other hand, we aim at providing a fully automated and intelligent tool for supporting the RCA procedures, including an automated procedure to create causal chains of events leading to root causes. The purpose of this paper is to explain the approaches in Steps 1 and 2. These are essential for identifying the root causes and listing priority of actions by, e.g., data mining or machine learning techniques in the following steps. The next subsection describes our method to extract correlated events from logs in *manufacturing execution system* (MES). SiView (IBM Corporation) is a major MES used in many fabs, including the IBM East Fishkill plant.

2.2 Extraction of Operational Events

In this subsection, we describe our approach to extract the operational events: lot-related events such as WIP levels and equipment-related events such as unscheduled downtime. An operational event can be a candidate as a node in a causal factor graph. Robinson, Fowler, and Neacy (2003) listed unscheduled downtime as one of the five biggest loss factors in fabs. Therefore, it is reasonable to use such events in our experiments. The procedures to extract those events from the history logs are outlined below.

2.2.1 Extraction of Events from Lot Operation History Logs

The events recorded in logs, or the *raw events*, are often too detailed and do not correspond to operational events in fabs. First, we extract the operational events by collecting the history of lot operations in the logs. We extract 4-tuple instances of data expressed as $(timespan/timestamp, sector, attribute, value)$. Here, a *sector* is a group of pieces of equipment categorized by wafer fabrication process such as lithography, inline test, and furnace. In the IBM East Fishkill plant, there are 15 sectors according to the types of equipment in SiView which we refer to sectors 1–15. Table 2 shows the attributes of statistics we are using in the experiments. We also studied other attributes, such as rework counts and the numbers of released lots, but due to limitation of space we omit them. Note that for all of this data, the value of the attribute is numerical. Next, we discretize the values of those instances by setting thresholds and classifying them (i.e., replacing their numerical values) with three labels: *High*, *Normal*, and *Low* (or respectively, *Bubble*, *Normal*, and *Hole* for WIP level).

2.2.2 Extraction of Events from Equipment Status Change History Logs

SEMI E10, or *E10*, is the specification for definition and measurement of equipment reliability, availability, and maintainability (SEMI 2004). E10 defines the categories of equipment conditions that can be used to track, compare, and evaluate the performance and other metrics of the equipment within a factory or among different factories. According to E10, the equipment status consists of 6 states: *Engineering*, *Nonscheduled*, *Productive*, *Scheduled downtime*, *Standby*, *Unscheduled downtime*. We label the status of equipment in the fab according to states in E10 and, similarly as before, obtain 4-tuple instances of data whose attribute is called "Equipment status".

3 LOG-DRIVEN DISCRETE EVENT SIMULATOR

3.1 Our Approach

Due to the large numbers of pieces of equipments, lots, routes, and dispatching rules in fabs, building a detailed simulation model is almost impossible. It is possible to get a more accurate result, however the simulator tends to be expensive to maintain and implement. Thus, it is important to select an appropriate design of the simulator to build

Table 2: Typical statistics categories.

Attribute	Description
WIP level	Number of lots/wafers waiting in a sector at a certain time
Frequency	Number of lots/wafers arrived at a queue of a sector in a certain period
Throughput	Number of lots/wafers completed at a sector in a certain period
Processing time	Average processing time (per lot/wafer) of a sector
Hold lot	Number of lots/wafers holding in a certain period
High priority lot	Number of waiting lots/wafers whose priority is high in a certain period
Small lot	Number of waiting lots/wafers whose number of wafers is low in a certain period

a reliable graph of correlated events as inexpensively as possible.

There are two main approaches for analyzing the performance of the fabs that can be used to model and infer a graph of correlated events: those based on analytical approaches (Kumar and Kumar 2001, Connors, Feigin, and Yao 1996) and those based on simulation approaches (Schmidt, Rose, and Weigang 2006, Qi, Tang, and Sivakumar 2002). Analytical approaches use a stationary assumption and therefore are good for modeling the long term WIP levels. However, they may be unsuitable for modeling the short term WIP levels due to frequent changes of product or priority mix or human intervention. For this reason, many researchers are studying the performance by using simulation approaches (Schmidt, Rose, and Weigang 2006) that can show the most effective strategies for lot releases, dispatching, and scheduling equipment maintenance (Qi, Tang, and Sivakumar 2002).

Since our objective is to analyze the root causes of WIP bubbles which occur on the short term, we prefer a simulation approach. In this paper, we restrict our discussion to finding relationships between equipment downtime and throughput in a sector or among different sectors of a fab. Our discrete event simulator is based on actual logs in an MES as already mentioned, and therefore is called a log-driven simulator. Compared to analytical approaches, a log-driven simulator assumes no mathematical models or parameter settings. Thus, it is more robust against changes in product mix, priority, and dispatching rules and is less expensive to maintain. However, it also imposes some limitations as discussed in Section 5.

3.2 Simulator Design

We describe our simulator design. Figure 3 gives an overall picture of the design. It consists mainly of three components: *data collection*, *simulation*, and *statistics generation*. Both data collection and statistics generation support simulations, but they can be used without running a simulation.

3.2.1 Data Collection

This component executes SQL queries and saves the results as CSV files. The simulator manages a *fab object*, which contains attributes such as equipment status and lot attributes,

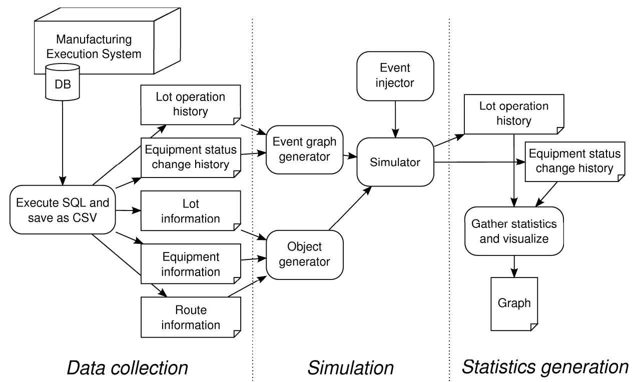

Figure 3: The log-driven simulator design.

and a priority queue to execute the simulation events and update the object. Thus, it requires the information about the lot, equipment, and route for a fab object and history logs about the lot operations and equipment status changes for the simulation events.

3.2.2 Simulation

This component consists of four sub-components: *object generator*, *event graph generator*, *event injector*, and *simulator*. The simulator needs a simulation event graph and a fab object.

The object generator creates a fab object from lot, equipment, and route information. The lot of an object has a unique ID, a type (*Production*, *Engineering*, *Monitor*), a priority, a route ID, a number of wafers, and so on. The equipment of an object has a unique ID, a sector, a capacity, a batch size, and so on. The route of an object has a unique ID and a sequence of step IDs. The step has a unique ID, the order in the route, and a list of equipment that can process.

The event graph generator creates a simulation event graph from lot operation history logs and equipment status change history logs. Figure 4 shows an example of a simulation event graph. A simulation event, which is equal to a record in the history logs, is defined as a node in the graph. If two events are dependent, they are connected by a directed edge. Each edge has a weight defined by the time difference between the nodes at the each end of the edge.

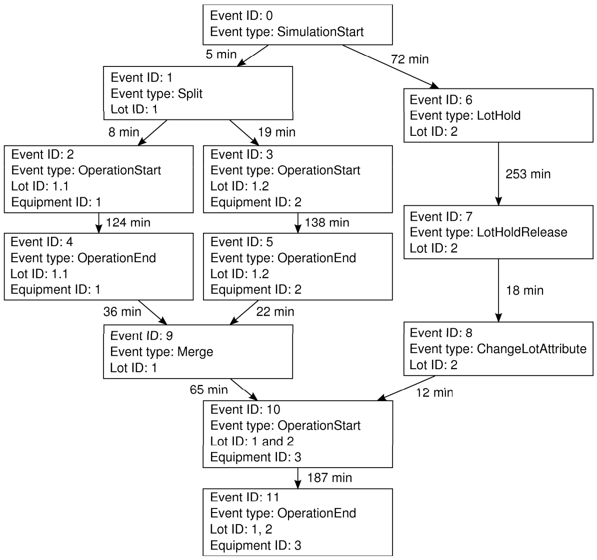

Figure 4: An example of a simulation event graph.

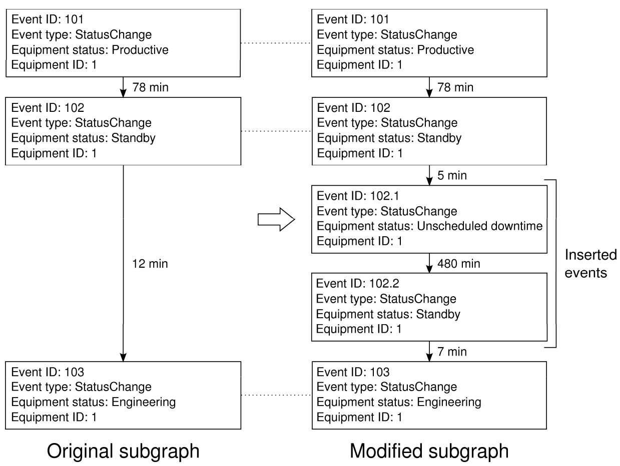

Figure 5: Original and modified graph inserted two events: unscheduled down and standby.

The event injector can insert, remove or modify simulation events and related edges of the simulation event graph. Figure 5 shows an example of inserting new events. In order to investigate the effects of 480 minutes of unscheduled downtime of a piece of equipment, we insert two simulation events, one *Unscheduled downtime* event (event ID 102.1) and one *Standby* event (event ID 102.2). In this case, the weight of the edge is 480 (minutes). Since the status of the equipment with ID 1 in the fab object is unscheduled downtime from the event with ID 102.1 to the event with ID 102.2, then all of the *OperationStart* events about the equipment with ID 1 cannot be fired between them and are on hold in front of the queue. When the event with ID 102.2 is fired and the status of the equipment with ID 1 becomes standby, the events on hold related to the equipment with ID 1 are released.

The simulator reproduces the history logs based on the simulation event graph modified by the event injector. If no events are injected, the simulator just replays the history logs. The simulator manages a timestamp-based priority queue. A simulator event has a timestamp attribute. The root node of the graph is put into the priority queue. This is the *SimulationStart* node in Figure 4. After execution, the simulator eliminates the outcoming edges from the node and put the nodes which have no incoming edges into the priority queue. When a node is executed, the simulator changes the current capacity or status of the equipment according to the simulation event type. This operation is repeated until the queue is empty.

3.2.3 Statistics Generation

This component compiles the statistics from the history logs. It can read not only the results from the simulator but also the original history logs in the MES. The types of statistics are shown in Table 2.

4 SENSITIVITY ANALYSIS

4.1 Outline

A graph of correlated events is constructed by linking its nodes (representing operational events), that are in cause-effect relationships. If the nodes are simply linked when their occurrence times are close to one another, the resulting graph of correlated events is almost complete. This will obviously lower the performance of data mining tools in extracting the root causes.

This section describes our proposed method of extracting correlated events by *sensitivity analysis* (SA). SA is a set of techniques for studying the sensitivity between the changes of the inputs and the changes of the outputs as summarized in Saltelli, Chan, and Scott (2000). In this paper, we used the correlation coefficient as the measure of sensitivity. The correlation coefficient is the linear correlation coefficient computed from the input-output pair of points (x_i, y_i) where $i = 1, \ldots, N$. The correlation coefficient r_{xy} between x and y is defined as

$$r_{xy} = \frac{\sum_{i=1}^{N}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{N}(x_i - \bar{x})^2}\sqrt{\sum_{i=1}^{N}(y_i - \bar{y})^2}}$$

where $\bar{x} = \sum_{i=1}^{N} x_i/N$ and $\bar{y} = \sum_{i=1}^{N} y_i/N$. If r_{xy} is close to 1 or -1, we say there is a *strong positive* or *strong negative correlation*, respectively, between the input x and the output y. On the other hand, if r_{xy} is close to 0, we say there is only a *small correlation*.

Operational events are generated at regular time intervals from history logs, with time t_k $(k = 1, \ldots, n)$. Then it is

Table 3: Correlation coefficients between equipment downtime and throughput in a sector.

Sector	Correlation coefficient
1	-0.288
2	-0.672
4	-0.594
5	-0.938
6	-0.774
7	-0.962
8	-0.662
11	-0.845
13	-0.807
15	-0.730

clear that

$$\begin{aligned} L_i(t_k) &> L_i(t_{k-1}) \qquad \text{if } \lambda_i(t_k) > \mu_i(t_k), \text{ and} \\ L_i(t_k) &< L_i(t_{k-1}) \qquad \text{if } \lambda_i(t_k) < \mu_i(t_k), \end{aligned}$$

where $\lambda_i(t_k)$ is the frequency in sector i from time t_{k-1} to t_k, $\mu_i(t_k)$ is the throughput in sector i from time t_{k-1} to t_k, and $L_i(t_k)$ is the WIP level in sector i at time t_k. We first study the sensitivity between unscheduled downtime of equipment and throughput in a sector. Note that the unscheduled downtime is ranked as one of the most significant factors causing capacity loss in fabs (Robinson, Fowler, and Neacy 2003). In addition, we also study the relationships between the throughput in sector i and the frequency in other sectors j $(j \neq i)$.

4.2 Relationships between Unscheduled Downtime and Throughput in a Sector

We selected a piece of equipment for each sector i $(i = 1,\ldots,15)$ and changed its status from *Standby* to *Unscheduled downtime* for k hours $(k = 1,\ldots,20)$. Then, we computed the correlation coefficient between downtime and observed average throughput. The results are shown in Table 3. Note that Sectors 3, 9, 10, 12, and 14 are not listed because their correlation coefficients are small, i.e., their throughputs were not affected because alternative equipment had sufficient capacity to cover for the down equipment. For the other sectors, there are negative correlations between downtime and throughput. From these results, we can confirm that when the capacity is saturated and equipment is down for a long period in a sector, then the throughput tends to decrease.

4.3 Relationships between Throughput of a Sector and Frequency of Other Sectors

In the previous experiment, we studied the relationships of equipment downtime and throughput in the same sector. In contrast, the objective of the experiment in this subsection is to study the relationships of events among different sectors. When the throughput of a sector decreases, it is expected that the frequency of the neighboring sectors also decreases.

Table 4: Correlation coefficients between average throughput $\bar{\lambda}_3$ of Sector 3 and average frequency $\bar{\mu}_j$ of sector j $(j \neq 3)$.

Sector	Correlation coefficient
1	0.03
2	0.60
4	0.91
5	-0.50
6	0.71
7	0.50
8	-0.10
9	0.46
10	-0.02
11	-0.13
12	0.31
13	0.63
14	-0.04
15	0.75

Our experiment is as follows. For each sector i, the number of pieces of down equipment is increased incrementally. We then compute the correlation coefficients between the average throughput of sector i and the average frequency of sector j such that $j \neq i$ for all pairs (i, j). If the correlation coefficient is strong positive, we say sector i *influences* sector j. Due to limitations of space, in Table 4 we only present the results of an experiment for the throughput of Sector 3 and the frequency of the other sectors. According to the route definition in the MES, almost all of the next steps of Sector 3 are in Sector 15. However the capacity of Sector 15 is not saturated because of its low processing time, so Sector 3 influences Sector 4 rather than Sector 15. Since Sector 15 is a neighbor of various sectors, we got useful information about the actual effects from Sector 3.

4.4 Time Lags between Equipment Downtime and Throughput Declines

The experiments in Subsection 4.2 showed the relationships between unscheduled downtime and throughput in a sector. As mentioned before, there may be a time lag between them. Here, we study the average elapsed times from the occurrences of equipment downtime until the change point in the throughput. We injected 8 hours of unscheduled down simulation events into a piece of equipment in Sector 6 at randomly chosen times. The results are shown in Table 5. The average elapsed time can be used for grouping events. In particular, the value of the throughput from the time when the equipment goes down (or from the standby time) plus the

Table 5: Average elapsed time from the occurrenes of equipment downtime until the declines of throughput.

Sector	Average elapsed time [mins]	Variance
3	10.0	39.1

average elapsed time can be considered as a single value of grouped events. This can also be used to measure the time lags between the occurrences of unscheduled downtimes and the declines of throughput.

5 CONCLUSIONS

We have discussed the design of a discrete event simulator for extracting a graph of correlated events from raw events in the history logs of SiView. Using the simulator, we proposed a method of extracting the relationships among the operational events and showed some preliminary results of experiments to obtain:

1. the relationships between equipment downtime and the throughput in a sector,
2. the relationships between the decrease of the throughput in a sector and the decrease of the frequency in other sectors,
3. a method to measure time lags between the occurrences of unscheduled downtime and the declines of throughput.

Our simulator does not assume any statistical or mathematical model and uses only the history logs in the MES. Nevertheless, it allows us to generate a graph of correlated events by creating links between events while considering the time windows and removing links between uncorrelated events. We believe that these are unique features of our simulator. Our future work will seek to verify that the resulting graphs of correlated events can be used to enhance the performance of data mining tools in root cause analysis.

While the simulator is simple to maintain, it still needs additional modeling and implementation for cases when information not included in the logs is needed. For example, to simulate lot split events, we need to decide how many wafers are included in each split lot, the routing path, processing times, and so on. Currently, our simulator cannot simulate these types of events. Adding additional capabilities while maintaining the low cost of maintenance is another challenge for future work.

ACKNOWLEDGMENTS

Authors would like to thank Parth Dave and Dr. Steven Moskowitz from IBM East Fishkill for valuable suggestions on the subject matter and Dr. Ching-Hua Chen-Ritzo and Dr. Robert Baseman from IBM T. J. Watson Research Center for technical discussions. Authors also thank Shannon Jacobs and anonymous reviewers for proofreading and suggestions to improve the paper.

REFERENCES

Connors, D. P., G. E. Feigin, and D. D. Yao. 1996. A queueing network model for semiconductor manufacturing. *IEEE Transactions on Semiconductor Manufacturing* 9 (3): 412–427.

Hassoun, M., G. Rabinowitz, and S. Lachs. 2005, September. Classification and cost estimation of WIP bubbles in a fab. In *Proceedings of IEEE International Symposium on Semiconductor Manufacturing*, 323–326.

IBM Corporation. IBM SiView Standard. Available via <www-06.ibm.com/jp/iisc/english/po/siview/index.html>.

Kumar, S., and P. R. Kumar. 2001. Queueing network models in the design and analysis of semiconductor wafer fabs. *IEEE Transactions on Robotics and Automation* 17:548–561.

Little, J. D. C. 1961. A proof for the queueing formula $L = \lambda W$. *Operations Research* 9 (3): 383–387.

Qi, C., T. K. Tang, and A. I. Sivakumar. 2002, December. Simulation based cause and effect analysis of cycle time and WIP in semiconductor wafer fabrication. In *Proceedings of the 2002 Winter Simulation Conference*, Volume 2, 1423–1430. San Diego, California, USA.

Robinson, J., J. Fowler, and E. Neacy. 2003. Capacity loss factors in semiconductor manufacturing. Working paper.

Rooney, J. J., and L. N. V. Heuvel. 2004, July. Root cause analysis for beginners. *Quality Progress*:45–53.

Saltelli, A., K. Chan, and E. M. Scott. 2000. *Sensitivity analysis*. John Wiley and Sons Inc.

Schmidt, K., O. Rose, and J. Weigang. 2006, December. Modeling semiconductor tools for small lotsize fab simulations. In *Proceedings of the 2006 Winter Simulation Conference*, 1811–1816. Monterey, California, USA.

SEMI 2004. *SEMI E10-0304E: Specification for definition and measurement of equipment reliability, availability, and maintainability (RAM)*.

Weidl, G., A. L. Madsen, and S. Israelson. 2005. Object-oriented bayesian networks for condition monitoring, root cause analysis and decision support on operation of complex continuos processes: Methodology & applications. *Technical Report 2005-1*. 36 pages, IST-University of Stuttgart.

AUTHOR BIOGRAPHIES

RYO HIRADE is a researcher at Tokyo Research Laboratory of IBM Japan. Her research interests are in the area of performance analysis. She received B.S. and M.S. degrees in Computer Science from Tokyo Institute of Technology, Japan. Her email address is <rhirade@jp.ibm.com>.

RUDY RAYMOND HARRY PUTRA is a researcher at Tokyo Research Laboratory of IBM Japan. He received B.S. (2001), M.S. (2003), and Ph.D. (2006) degrees from Kyoto University. His email address is <raymond@jp.ibm.com>.

HIROYUKI OKANO is a researcher of Tokyo Research Laboratory of IBM Japan. He received B.S. (1988) and M.S. (1990) degrees in Information Science from the Tokyo University of Agriculture and Technology. He joined Tokyo Research Laboratory, IBM Japan, in 1990, and researched on user interface and system software. In 1995, he joined the operations research group, and started to research on combinatorial optimization. His research interests include the traveling salesman problem and the vehicle routing problem. His email address is <okanoh@jp.ibm.com>.