SAND2000-0779C

Automated Analysis of Failure Event Data

Dr. Corey Hennessy, Avistar, Inc.; Fred Freerks, Applied Materials; Dr. James E. Campbell and Bruce M. Thompson, Center for System Reliability at Sandia National Laboratories

Keywords:

Abstract

Our paper focuses on fully automated analysis of failure event data in the concept and early development stage of a semiconductor-manufacturing tool. In addition to presenting a wide range of statistical and machine-specific performance information, algorithms have been developed to examine reliability growth and to identify major contributors to unreliability. These capabilities are being implemented in a new software package called Reliadigm.

When coupled with additional input regarding repair times and parts availability, the analysis software also provides spare parts inventory optimization based on genetic optimization methods. The type of question to be answered is: If this tool were placed with a customer for beta testing, what would be the optimal spares kit to meet equipment reliability goals for the lowest cost? The new algorithms are implemented in Windows® software and are easy to apply.

This paper presents a preliminary analysis of failure event data from three IDEA machines currently in development. The paper also includes an optimal spare parts kit analysis.

Introduction

Early identification of reliability issues has become more important as customer's specifications call out MTBF, MTBI, MTTR, and availability requirements as part of the binding performance metrics that will be used to evaluate our tools. These requirements can be tied to "stiff" financial penalties that translate directly to system margins and gross profit. Tools that can be used to identify and help correct reliability issues are therefore becoming more important. One of the available tools is the Reliadigm reliability analysis software. Reliadigm is part of the Reliadigm Reliability Suite, tools and technologies originally developed by Sandia National Laboratories. Reliadigm is a highly configurable software tool that can provide a wide array of reliability analysis results from raw failure event data. In addition to user-definable reliability metrics, the software performs sensitivity and variability analysis. Reliadigm also includes an optimization capability that can be applied to spare parts inventories or to reliability trade-off studies. The capability of the software to aid in MTBF and MTTR analysis makes it an ideal candidate as a software analysis tool, particularly when combined with the optimization capabilities.

To test the capability of the software, a reliability database (in the form of failure event records) based on product development was used. So as not to reveal the true product under development, a fictitious product is

DISCLAIMER

called IDEA (Implant Depth Estimation & Analysis) tool was created for this analysis. IDEA is a concept. The modules described do not exist and any resemblance between this and any existing tool is strictly accidental. However, the reliability data used in this analysis is based on actual development projects and does have a basis in fact. The data has been modified to hide the true identity of the projects. System up times and wafer counts are similarly based on development projects.

Results and Discussion

Data Collection

The failure event data used in the analysis is a highly modified (to disguise the original projects) series of system failure records taken from three similar systems. The records are based on actual failure events recorded during early product development. The records are entered into a Microsoft Access® two-part database that records wafer transfers, machine states, and failure data. The time-frame of the data is from January through December 1999. The data was not specifically taken for this analysis and some data for the individual failure records is based on best estimates. The majority of the records indicate software, software induced, or communication between modules as failure modes. This is not unexpected during early product development. The failure data collection and system run information was performed using a Reliability Database Main Table. The Table has pull-down menus based on the product structure and failure modes as identified by the FMECA. The table simplifies the data entry into the Access database. The database can be queried to provide the input for the analysis.

Failure Event Data

Failure event data were available for three IDEA machines. For two of the machines, the data sets covered about a year while the third data set covered only about a one-month time period. The dates are provided in Table 1.

	Machine 1	Machine 2	Machine 3
Start Date	01/09/1999	01/18/1999	11/15/1999
End Date	12/17/1999	01/04/2000	12/16/1999
Number of Failure Events	372	191	36
Total Hours	8,232	8,448	768
Productive Hours	1,563	641	111
Non-Scheduled Hours	5,373	7,398	590
Unscheduled Downtime	1,296	409	67

Table 1. Machine Data Sets

The raw event data was provided in a spreadsheet format and included, for each failure event, a machine identifier, a failure location and failure mode identifier (module, submodule, failure mode and failure code), a failure event date, and repair time. Also included were daily and cumulative values of tractor time, wafer cycles, and power on time. For this analysis, machines were assumed to be productive when tractor time was reported and to be in "standby" mode (i.e., nonscheduled time) at all other times when not being repaired. The machines were assumed to be in one of these three states at all times. The data sets contained only failure events: we generated corresponding standby events in the spreadsheet to accurately account for nonscheduled time rather than simply using an average utilization fraction. As failure times were not available, we assumed that the first failure on any given day occurred at 12:00 am and that any productive time immediately preceded the corresponding failure event.

Preparing the failure event data for analysis provided some lessons both in terms of what data to record as well as how to record the information. For example, the time taken to repair failures was not recorded contemporaneously with the failure. This meant that technicians were required to estimate repair times after the fact, a time-consuming process after the event but information that would have taken seconds to record at the time. Similarly, the time of day could have been recorded for each failure to make the process of time accounting more accurate. Finally, the failure coding scheme, while hierarchical, could have benefited from having dashes separate the different levels of the hierarchy. For example, PRO-WFR-SC-TILT is more easily interpreted than PROWFRSCTILT.

The failure event data was imported into Reliadigm for analysis. The Reliadigm Import Wizard performs a sequence of checks and then provides data summarized by machine. When event data importing is complete, the software automatically (i.e., with no further user interaction) builds a reliability model and performs the statistical calculations needed. At this point, the complete range of Reliadigm analysis results is immediately available.

Reliability Analysis Results

Figure 1 shows a Reliadigm histogram of MTBF based on statistical analysis of data from all three machines. The range of values is from about 2.5 to 5 hours of operational time between failures.

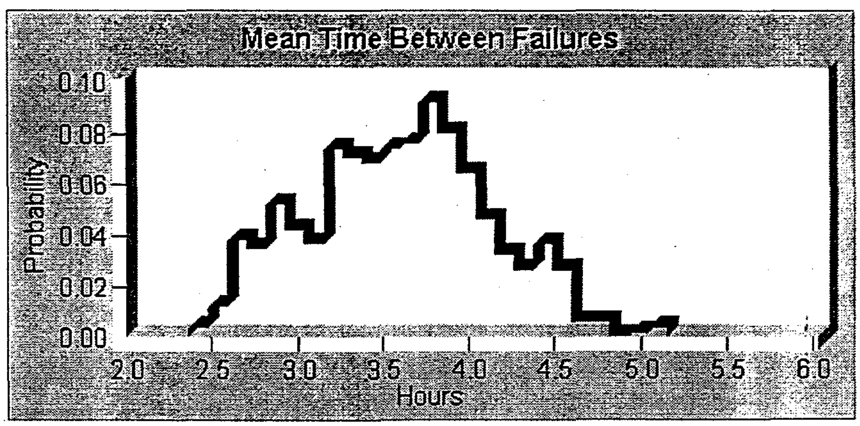


Figure 1. Histogram of MTBF for the Three-Machine Sample

Figure 2 shows a bar chart of actual MTBF values for the three machines. The MTBF values are Machine 1 (4.2 hours), Machine 2 (3.4 hours), and Machine 3 (3 hours). The range of MTBF values in the histogram is wider than the actual machine MTBF values because the statistical analysis seeks to characterize the population of machines based, in this case, on a sample size of 3.

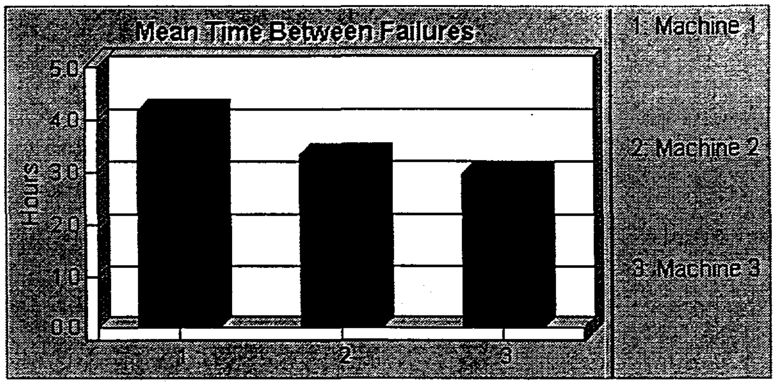


Figure 2. MTBF Values by Machine

To see which failure modes have the greatest influence on MTBF, consider the Reliadigm sensitivity plot shown in Figure 3. The Pareto plot shows that the Process Module Load Lock Software is, statistically, the largest contributor to system failure. It also shows that there is considerable variability in this result from machine to machine. This can be seen in Figure 3 by looking at the percentiles for each failure mode. The percentiles indicate that the fractional contribution of the "Proc LL Software" failure mode ranges from less than one percent to about 7 percent of all failures. In fact, "Proc LL Software" occurred three times in only 111 hours of operation on Machine 3 but was seen twice on Machine 1 (1,563 hours of operation) and twice on Machine 2 (641 hours of operation).

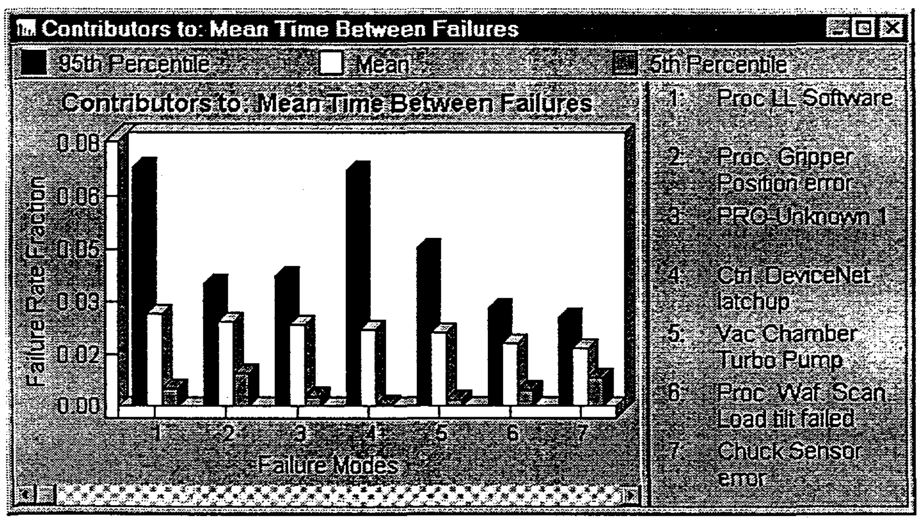
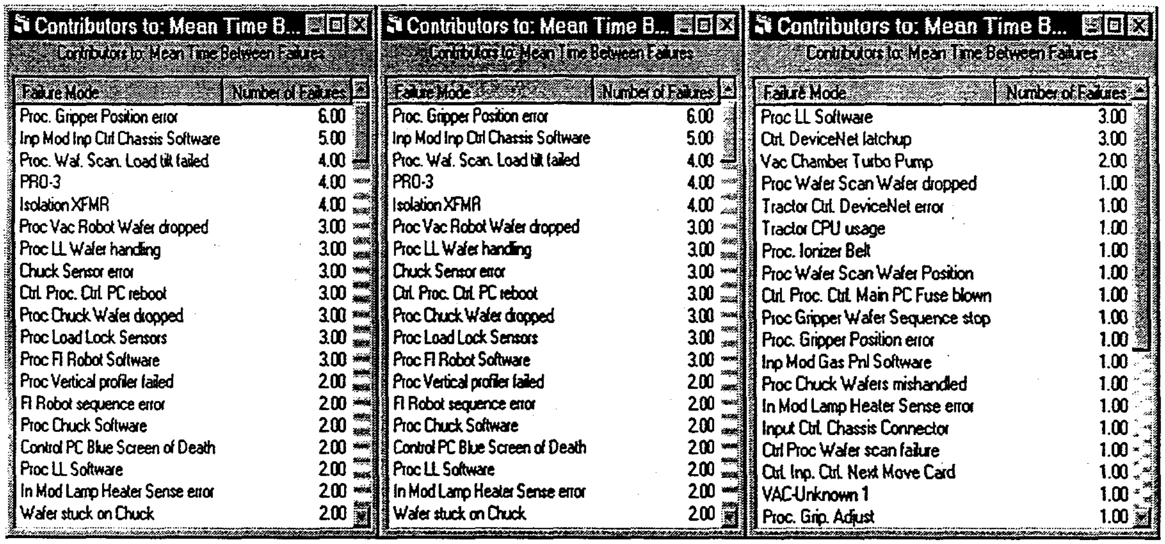


Figure 3. Pareto of Contributors to System Failure

The Reliadigm software used for this analysis also provides a wide range of equipment-specific results. For example, Figure 4 shows the most frequently occurring failure modes for each of the three machines. Notice that "Proc LL Software" is the most frequently occurring failure mode on Machine 3 but is not among the top ten most frequent failure modes on Machines 1 and 2.



Contributors to: Mean Time B...

Contributors to: Mean Time Between Failures

Failure Mode	Number of Failures
Proc. Gripper Position error	6.00
Inp Mod Inp Ctrl Chassis Software	5.00
Proc. Waf. Scan. Load tilt failed	4.00
PRO-3	4.00
Isolation XFMR	4.00
Proc Vac Robot Wafer dropped	3.00
Proc LL Wafer handling	3.00
Chuck Sensor error	3.00
Ctrl. Proc. Ctrl. PC reboot	3.00
Proc Chuck Wafer dropped	3.00
Proc Load Lock Sensors	3.00
Proc Fl Robot Software	3.00
Proc Vertical profiler failed	2.00
Fl Robot sequence error	2.00
Proc Chuck Software	2.00
Control PC Blue Screen of Death	2.00
Proc LL Software	2.00
In Mod Lamp Heater Sense error	2.00
Wafer stuck on Chuck	2.00

Contributors to: Mean Time B...

Contributors to: Mean Time Between Failures

Failure Mode	Number of Failures
Proc. Gripper Position error	6.00
Inp Mod Inp Ctrl Chassis Software	5.00
Proc. Waf. Scan. Load tilt failed	4.00
PRO-3	4.00
Isolation XFMR	4.00
Proc Vac Robot Wafer dropped	3.00
Proc LL Wafer handling	3.00
Chuck Sensor error	3.00
Ctrl. Proc. Ctrl. PC reboot	3.00
Proc Chuck Wafer dropped	3.00
Proc Load Lock Sensors	3.00
Proc Fl Robot Software	3.00
Proc Vertical profiler failed	2.00
Fl Robot sequence error	2.00
Proc Chuck Software	2.00
Control PC Blue Screen of Death	2.00
Proc LL Software	2.00
In Mod Lamp Heater Sense error	2.00
Wafer stuck on Chuck	2.00

Contributors to: Mean Time B...

Contributors to: Mean Time Between Failures

Failure Mode	Number of Failures
Proc LL Software	3.00
Ctrl. DeviceNet latchup	3.00
Vac Chamber Turbo Pump	2.00
Proc Wafer Scan Wafer dropped	1.00
Tractor Ctrl. DeviceNet error	1.00
Tractor CPU usage	1.00
Proc. Ionizer Belt	1.00
Proc Wafer Scan Wafer Position	1.00
Ctrl. Proc. Ctrl. Main PC Fuse blown	1.00
Proc Gripper Wafer Sequence stop	1.00
Proc. Gripper Position error	1.00
Inp Mod Gas Pnl Software	1.00
Proc Chuck Wafers mishandled	1.00
In Mod Lamp Heater Sense error	1.00
Input Ctrl. Chassis Connector	1.00
Ctrl Proc Wafer scan failure	1.00
Ctrl. Inp. Ctrl. Next Move Card	1.00
VAC-Unknown 1	1.00
Proc. Grip. Adjust	1.00

Figure 4. Most Frequently Occurring Failure Modes

To see if the system MTBF varied over the duration of the data, we plotted MTBF over three month time intervals. The result is shown in Figure 5. The plot indicates that MTBF decreases for both Machines 1 and 2 in the first three time intervals, but increases slightly in the fourth interval. Note that Machine 3 did not operate in the first three quarters of the test period.

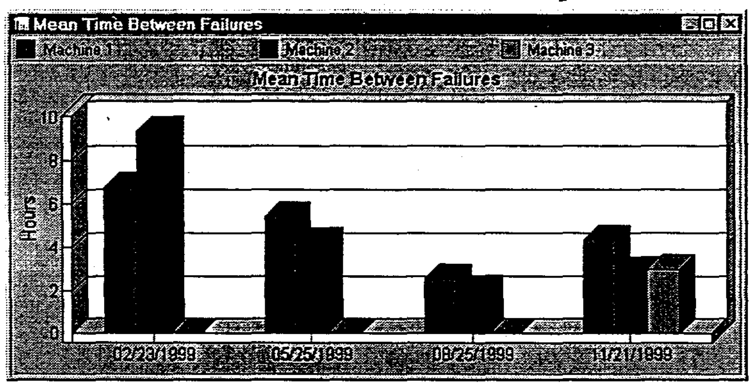


Figure 5. Time Plot of MTBF by Machine

Optimal Spares Kit Analysis

The second phase of this study involved an optimal spares kit analysis. An optimal spares kit is the set of spare parts that, for a given kit cost, minimizes downtime. For example, if a single IDEA tool were sent to a customer for beta testing, the optimal spares kit is the best possible set of spare parts, for a given budget, that could be sent with the machine to minimize downtime. The Reliadigm software has the ability to perform optimization analysis to determine the optimal spares kit to support a single machine and can also be used to determine the optimal spares inventory to support multiple machines. Only the optimal spares kit analysis is reported here.

To set up the analysis, we first defined a complete set of spares for the IDEA tool.

Although the failure event data revealed almost 300 failure modes, a large fraction of these failure modes were software-based, did not require spare parts, and had relatively short downtimes. For failures requiring a spare part, downtime with the spare on hand was estimated based on experience. Downtime without the spare was calculated by adding an urgent-shipping time of 24 hours to the downtime with spare. We also assumed that the normal time taken to restock a spare was 2 weeks. These assumptions could easily be modified based on local conditions. The purchase cost for each spare part was included in the analysis but, although Reliadigm can account for storage costs in the calculations, these were not included in this application.

Based on this problem setup, there are more that 3 x 10^{15} possible spare part kits to be considered. Reliadigm uses a powerful genetic algorithm to find the optimal spares kit for different budgets. Space does not permit us to list an optimal spares kit here. However, one way to present the results is to plot, for different values of budget, the average downtime for the optimal spares kit. Figure 6 provides such a plot for this analysis. The base case (i.e., the case where no spares are available on site) has an average downtime of 5.5 hours. As mentioned earlier, this is relatively low because so many of the failures observed were software-based and required very little time to "repair."

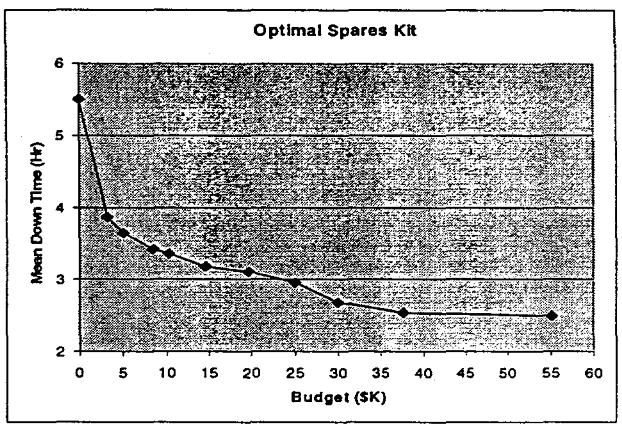


Figure 6. Downtimes for Optimal Spares Kits of Varying Cost

Figure 6 shows that, as the budget for spare parts increases, the resulting downtime decreases. It is important to note that each point plotted in Figure 6 is an optimal solution. By this we mean that, for a given budget, the downtime indicated is obtained with the optimal spares kit available for that budget. Figure 6 also shows that the greatest return on investment (i.e., the greatest downtime reduction per dollar spent on the spares kit) occurs for relatively small investments. It also becomes increasingly expensive to gain additional reductions in downtime. In this case, Figure 6 shows that very little benefit is obtained by spending more than $30K on the spares kit. However, deciding on the "right" spares kit budget will ultimately depend on how the user values downtime for the IDEA tool. More importantly, these Reliadigm results provide decision-makers with the information they need needed to make well-informed decisions.

Conclusions

As with any analysis, having the proper data is of utmost importance. In preparing the failure event data for analysis, it became quite clear that knowing what data to record as well as how to record the system information meant that knowing the analysis requirements and setting up the data log must be well understood. If Applied Materials had more precisely recorded the failure or operating record times, a more accurate performance evaluation would have resulted. Recording the time taken to repair failures at the time the repair was completed would have yielded better estimates of MTTR. Similarly, recording the exact time for each failure or system operating condition would have made time accounting much more accurate.

Modern reliability analysis techniques and software tools like Reliadigm allow manufacturers to demonstrate that Customer Specifications have been met. When repair times and parts availability are included in the data, the analysis software can provide a spare parts inventory optimization. This helps assure that average repair times meet the customer MTTR requirements and provides a basis for providing a recommended spares kit.

Recommendations

It is recommended that a standard method for recording and reporting system performance and failure data be instituted throughout a company. The selected standard would allow any reliability engineer to use raw system data to evaluate modules and sub-modules as well as fully integrated systems for reliability performance. The addition of system repair information greatly improves a manufacturer's ability to provide optimum spares for both current and future systems.

Acknowledgements

The authors would like to acknowledge the assistance of Applied Materials and Technologies for the use of their data, and Avistar, Inc. for providing the Reliadigm software.

References

Reliadigm™ User's Manual, Avistar, Inc., 2000. In preparation.

Sandia is a multiprogram laboratory operated by Sandia Corporation, a Lockheed Martin Company, for the United States Department of Energy under contract DE-AC04-94AL85000.