Reducing Variability in Equipment Availability at Intel using Systems Optimization

by

William Kwong

M.Eng. Electrical Engineering, Cornell University 1999
B.S. Electrical Engineering, Cornell University 1998

Submitted to the Sloan School of Management and the
Department of Electrical Engineering and Computer Science
In Partial Fulfillment of the Requirements for the Degrees of

Master of Business Administration and
Master of Science in Electrical Engineering and Computer Science

In conjunction with the Leaders for Manufacturing Program at the
Massachusetts Institute of Technology
June 2004

Signature of Author ______________________
Sloan School of Management
Department of Electrical Engineering and Computer Science
May 1, 2004

Certified by ______________________
Stephen Graves
Professor, Sloan School of Management
Thesis Supervisor

Certified by ______________________
Dimitris Bertsimas
Professor, Sloan School of Management
Thesis Supervisor

Certified by ______________________
Margaret C. Andrews
Executive Director of the MBA Program
Sloan School of Management

Certified by ______________________
Arthur C. Smith
Chair, Committee on Graduate Students
Electrical Engineering and Computer Science

Reducing Variability in Equipment Availability at Intel using Systems Optimization

by

William Kwong

Submitted to the Sloan School of Management and the
Department of Electrical Engineering and Computer Science
In Partial Fulfillment of the Requirements for the Degrees of

Master of Business Administration and
Master of Science in Electrical Engineering and Computer Science

Abstract

Equipment management is an important driver behind operational efficiency, since capital equipment makes up about 40% of the average semiconductor manufacturer's total assets. The main goal of this project is to reduce variability in tool availability by planning for usage-driven preventive maintenance. A method and associated tools are proposed and investigated in the context of the Thin Films area in Intel's Hudson facility. The solution we propose incorporates the following characteristics:

- Drives towards a balanced preventive maintenance (PM) schedule such that PMs are evenly distributed in time
- Enables fast recovery to a normal PM schedule after unexpected events occur on the factory floor, e.g. equipment breakdown, by re-distributing loads on each tool
- Facilitates performance tracking and accountability
- Ensures consistency in the decision-making process

We will describe the conceptual method and the implementation process, from prototype deployment to the development of a production application. Alternative solutions using case-based reasoning and rule-based systems will also be discussed. We will conclude by discussing the role of automated decision systems in manufacturing and outline key issues to be considered in choosing an optimal design.

Thesis Advisors:

Dimitris Bertsimas	Professor, Sloan School of Management
Stephen Graves	Professor, Sloan School of Management

Acknowledgements

The author gratefully acknowledges the support and resources made available to him through the MIT Leaders for Manufacturing program, a partnership between MIT and major U.S. manufacturing companies.

This work is made possible by the support of Gretchen Johnson, Greg Gerstenzang, John Sheridan, Karl Koch and Melinda Manente at Intel and my thesis advisors, Professor Dimitris Bertsimas and Professor Stephen Graves. I would like to thank Professor Al Drake and Professor Howard Shrobe in the Department of Electrical Engineering and Computer Science for their valuable advice. The Automation Group at Intel Hudson also provided critical support during the implementation phase of the internship.

Biographical Note on the Author

William Kwong graduated summa cum laude from Cornell University, Ithaca, NY in May, 1998 with a Bachelor of Science degree in Electrical Engineering. He also has a Master of Engineering degree from Cornell University. William joined IBM in July, 1999 and worked in IBM's Communications Research and Development Center in Fishkill, NY prior to MIT.

Reducing Variability in Equipment Availability at Intel using Systems Optimization

Reducing Variability in Equipment Availability at Intel using Systems Optimization

LIST OF FIGURES

CHAPTER 1

BACKGROUND AND INTRODUCTION

1.1 Background

From 2001 to 2002, the business environment for semiconductor manufacturers became increasingly challenging. Some in the industry called it "the perfect storm", as profits in the semiconductor sector dropped abruptly after a prolonged period of rapid growth. The industry downturn led to a drop in capital additions to equipment. At Intel, capital additions to machinery and equipment totaled only $2.9 billion in 2002, compared with $5.9 billion in 2001 and $5.7 billion in 2000. The total number of Intel employees dropped by 6%. Yet during this period, Intel's microprocessor sales volume actually increased. The demand for microprocessors remained strong throughout 2003. Limited capital additions to equipment over the past 2 years and rising product demand have led to a growing emphasis on capital equipment management.

Other trends in the microprocessor business have also contributed to the increasing importance of capital equipment management. The low-end segment of the microprocessor market has evolved into a commodity business, as evidenced by falling average selling prices and limited product differentiation. Within this market segment, cost control and operational efficiency have taken over product innovation as core competencies. Equipment management is an important driver behind operational efficiency, since capital equipment makes up about 40% of the average semiconductor manufacturer's total assets.

As for high-end microprocessors, time to market has always been critical. This is primarily driven by short product lifecycles and rapid price erosion following new product introduction. For the manufacturing organization, this translates into a drive to reduce cycle time – the total time required to manufacture a wafer. At Intel's Hudson facility, the total cycle time for each wafer is approximately 60 days. Excessive variations in tool availability and in-process inventory levels contribute to cycle time. Other contributors to cycle time include the human wafer transport system unique to Intel Hudson.

Major products manufactured at Intel's Hudson semiconductor manufacturing facility (fab) include the Centrino microprocessor. The current manufacturing process involves hundreds of process steps and a workforce of approximately 1,000. Historical data show large fluctuations in in-process inventory levels. Such fluctuations have been partly attributed to variations in tool availability at critical areas. Several areas in the facility, including the Thin Films area, have been identified as capacity constraints.

1.2 Variability in Equipment Availability

Each tool can either be up for production or down for maintenance. Equipment availability measures the percentage of tools that are up for production at a given time. For example, if there are ten tools in a tool group and if eight tools are up for production, the equipment availability for that tool group will be 80%.

We will now discuss sources of variability in tool availability.

1.2.1 Equipment Maintenance

Preventive Maintenance (PM) - Process excursions due to faulty equipment are costly and often difficult to detect promptly. Hence preventive maintenance is instrumental in minimizing process disruptions and cost control. However, preventive maintenance may also contribute significantly to variability in equipment availability. If PM events are not planned, multiple tools of the same type may require preventive maintenance simultaneously. This may lower equipment availability and create a bottleneck, thus building up inventory at the corresponding process step.

There are two major categories of preventive maintenance (PM): calendar-based and usage-driven. Calendar-based PMs are performed periodically (e.g. weekly, monthly) regardless of tool usage levels. In contrast, usage-driven PMs are performed when cumulative usage since the last maintenance event has reached a threshold e.g. per 1000 wafers processed. Most types of equipment in Intel's semiconductor manufacturing facilities require either or both categories of preventive maintenance.

Unscheduled Maintenance - For certain process steps, unscheduled maintenance is a major contributor to variability in equipment availability. Unscheduled maintenance is required when a process excursion occurs. Due to the complexity and tight tolerances of tools, process excursions and unscheduled maintenance are more common in semiconductor manufacturing than in other types of manufacturing processes. Other causes of unscheduled maintenance are contamination (e.g. by Copper) and hardware failure.

In summary, both scheduled and unscheduled maintenance contribute to variability in equipment availability. The impact of preventive maintenance on variability in equipment availability can be greatly reduced through careful planning.

1.2.2 Labor

The availability of labor to operate and maintain equipment has a direct impact on equipment availability. Also, technicians are required to have various levels of certifications to perform maintenance activities on equipment.

1.2.3 Product Mix

Configuring equipment for a variety of products impacts equipment availability. The Intel Hudson facility has a narrow product mix and primarily focuses on manufacturing Centrino microprocessors. However, the re-entrant process flow in semiconductor manufacturing may create challenges similar to those in a high product mix environment. Wafers are processed by the same machine multiple times and each metal layer on a given wafer may require different settings and processing times.

1.3 Preventive Maintenance and Factory Constraints

Preventive maintenance scheduling is particularly important in semiconductor manufacturing. In this section, we will discuss the characteristics of semiconductor manufacturing that have made PM events more likely to contribute to factory constraints.

Small number of tools

In a semiconductor manufacturing facility, a functional area typically only has three to ten identical pieces of equipment. Given the small number of tools, the maintenance schedule of each tool often has a material impact on overall equipment availability. The impact is particularly significant in functional areas with a high rate of tool breakage and long repair times.

High equipment cost

Most types of equipment in the fab have unit costs in the $200K-$20 million range. Adding tools and maintaining excess capacity is not a cost-effective way to reduce the impact of maintenance activities.

Fluctuations in inventory level

Frequent unplanned process disruptions and re-entrant process flows in semiconductor manufacturing often make maintaining a smooth inventory profile challenging. This is evidenced by "WIP bubbles" propagating along the production line.

1.4 Problem Overview

In the Thin Films area, usage-driven preventive maintenance was often planned in an ad-hoc manner. At the beginning of shift, area coordinators determined the target output for each functional area. In each functional area, equipment engineers and technicians decided if there was a need to perform preventive maintenance during the shift. Occasionally, they may also decide to preferentially load one tool to speed up the occurrence of a usage-driven preventive maintenance event. Given the lack of guidelines to incorporate the large number of other factors affecting usage-driven preventive maintenance, there was no standard decision-making process to manage usage-driven preventive maintenance.

Other factors contributing to preferential loading of WIP among tools include physical location on the factory floor. WIP allocation was consistently biased towards certain tools in the tool group and different levels of cumulative usage were measured over a period of several weeks.

Given that ad-hoc planning of PM events contributed significantly to equipment availability and towards factory constraints in bottleneck areas, there was a need for a systematic framework to standardize usage-driven PM planning.

1.4.1 *Prior Work*

Measuring Variability in Equipment Availability

Prior work at Intel involved the development of metrics to measure variability in tool availability, such as the A80 metric. While average availability continued to receive much focus, newer metrics recognize the importance of minimizing variability in tool availability. A80 measures the percentage of time when equipment availability is above 80% (Please refer to Section 1.2 for the definition of equipment availability). Figure 1 shows how equipment availability of a hypothetical tool group varies with time. From the graph, we see that equipment availability is above the 80% line approximately 65% of the time. Therefore, the A80 measure for this tool group is approximately 65%. Ideally, the A80 measure should be above 95%. This would mean that at the 95% confidence level, at least 80% of equipment is available at any given time.

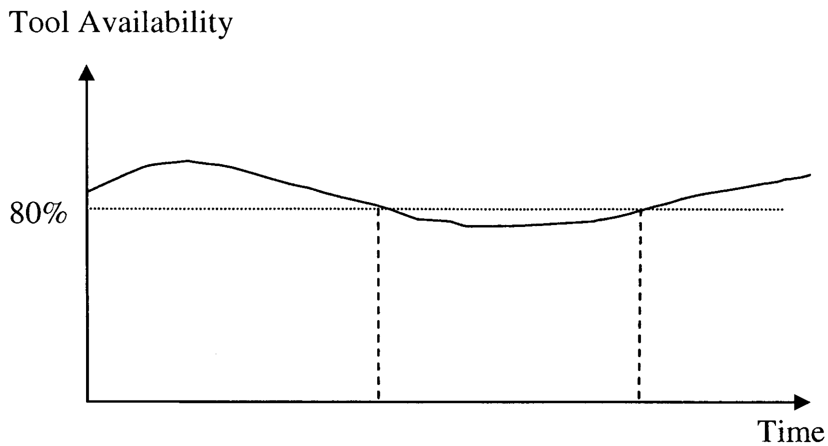


Figure 1 - Tool availability and Intel's A80 metric

Our project aims to advance the focus on variability in tool availability from the measurement stage to active management.

Scheduling Calendar-based Preventive Maintenance

Prior work on scheduling calendar-based preventive maintenance had been done by equipment engineers at Intel Hudson. Calendar-based preventive maintenance scheduling is considerably less dynamic. Because calendar-based preventive maintenance does not depend on usage levels, it is not influenced by the daily fluctuations in WIP across the facility. As a result, calendar-based PM scheduling seldom requires decisions to be made on a real-time basis. Because such scheduling tasks are not as time-sensitive, they can be deferred to the next available equipment engineer. In contrast, WIP flow constantly changes usage-driven PM schedules 24 hours a day. Hence managing usage-driven preventive maintenance requires technicians to respond in real time, sometimes without guidance from the equipment engineer.

1.4.2 *Key Challenges*

The key challenges to be addressed in a system that standardizes usage-driven preventive maintenance planning are summarized below. The system should:

- Drive towards a balanced usage-driven preventive maintenance (PM) schedule such that usage-driven PMs are evenly distributed in time when WIP flow is constant.
- Enable fast recovery to a normal PM schedule after unexpected events occur on the factory floor, e.g. equipment breakdown, by re-distributing loads on each tool.
- Facilitate performance tracking and accountability.
- Ensure consistency in the decision-making process. The system should be generic enough for deployment in multiple functional areas within the fab with minimal customization.

CHAPTER 2

CONCEPTUAL METHOD

Chapter 2 - Conceptual Method

In this chapter, we propose a model for planning usage-driven preventive maintenance by selectively distributing work-in-progress (WIP) among tools. The primary objective of this model is to reduce variability in tool availability in a functional area. Specifically, if the number of wafers processed per shift were constant, we would like consecutive preventive maintenance (PM) events in a tool group to occur as far apart in time as possible. Other issues highlighted in the "Key Challenges" section in Chapter 1 will also be addressed.

2.1 Conceptual Method Overview

Our approach consists of two steps: schedule planning and optimization. Based on current tool status and a subset of other constraints, the schedule planning step sets the ideal end-of-shift equipment usage levels and PM schedule. The optimization step detects differences between current and ideal tool usage levels and optimizes the allocation of WIP among tools accordingly.

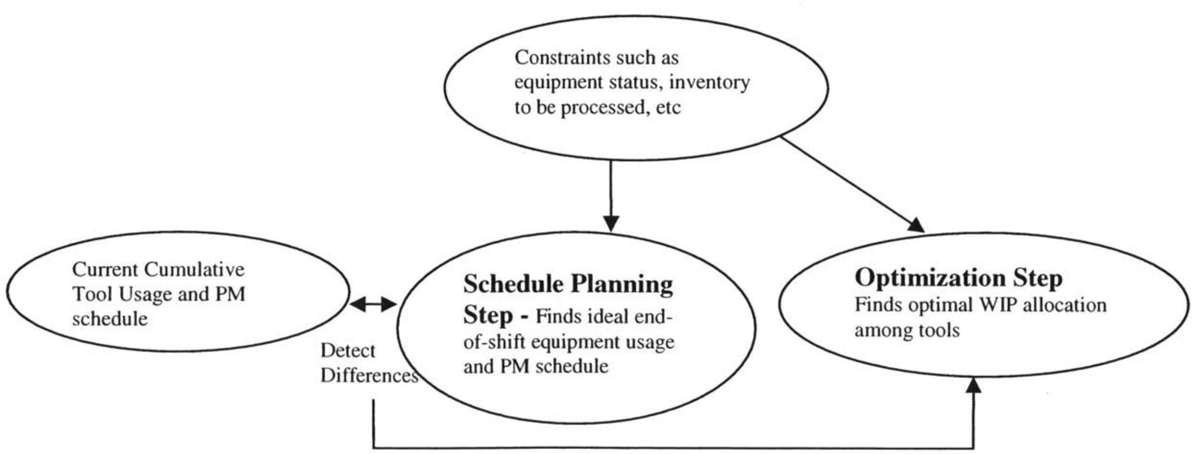


Figure 2 - Overview of Conceptual Method

2.1.1 Model Inputs

1. Total number of tools in the tool group and the current status for each tool, where the current status for each tool can either be "UP FOR PRODUCTION" or "DOWN".

2. For every machine, the component types requiring preventive maintenance and the maximum usage threshold for each component.

3. For each component, cumulative usage levels since the last PM.

4. Total active inventory and the number of wafers processed per shift at process steps covered by the tool group, where active inventory is defined by total inventory minus inventory-on-hold.

5. Active inventory and inventory turns per shift at upstream process steps.

2.1.2 Model Outputs

1. The recommended number of wafers to distribute to each machine during the next 12-hour period.

2. A score representing the difference between current and ideal usage levels.

2.1.3 Definition of Variables

We consider a tool group X which processes each wafer multiple times. For instance, we will consider a tool group that processes the wafer once for each metal layer. Note that a typical semiconductor process may have 4 to 7 layers of metal.

For re-entrant process P, let P_i represent a process step i. Each process step i is associated with a metal layer j. For metal layer j, the active inventory at process step P_i is represented by $Inv_{i,j}$. The sum of the expected queue time and processing time at process step P_i in metal layer j is represented by $\tau_{i,j}$. The tool group X performs a process step for each metal layer. The tool group consists of N identical tools: $X_1,...,X_N$. Our goal is to plan for WIP distribution among tools in tool group X for the next T hours, where T typically represents the number of hours in a shift.

2.2 Schedule Planning Step

2.2.1 Estimating Output Quantity Q_X

To determine ideal end-of-period usage levels for tool group X, we first estimate the expected output quantity Q_x, measured in wafers or kilowatt-hours (kwh), to be processed by tool group X over time period T. We sum the inventory currently at tool group X and at upstream processes as a first-order approximation for expected output, Q'_x. We assume that there are no disruptions along upstream process steps leading up to tool group X. The four steps involved in estimating output quantity Q_X are detailed below.

Step 1 - We determine for each metal layer j the number of upstream processes, k_j, to be included in the estimation of expected output Q_x. This depends on the historical average processing and queuing times at each upstream process step.

Definition of $\tau_{i,j}$

$\tau_{i,j}$ is defined as the sum of the processing and queuing time at process step i on metal layer j. Note that for our calculations, we are not concerned with how $\tau_{i,j}$ break downs into its two subcomponents because only the sum of the processing and queuing time is needed to find k_j.

For process P_i on metal layer j, we find k_j such that the following constraint is satisfied:

$$\sum_{i=I-k_j}^{I} \tau_{i,j} \approx T$$

where I denotes the process step performed by the tool group X on metal layer j.

For example, suppose process step 5 is performed on tool group X for metal layer 1, and assume that we want to determine the number of upstream process steps, k_1, to be included in the estimation of Q_x at process step P_5. If T=12 hrs, $\tau_{5,1}$=6 hrs, $\tau_{4,1}$=2 hrs and $\tau_{3,1}$=4 hrs, we would include 2 upstream process steps to satisfy the constraint

$\sum_{i=I-k_j}^{I} \tau_{i,j} \approx T$, hence k_1 equals 2. Note that k can be different for each metal layer j because processing times for wafers may vary from one metal layer to the next.

Step 2 - For metal layer j, we sum the inventory at process step P_I and at upstream processes. The number of upstream processes to be included in the summation is determined by k_j calculated in Step 1:

$$Q'_{x,j} = \sum_{i=I-k}^{I} Inv_{i,j}$$

Step 3 - The first order approximation of the expected output is found by summing across all metal layers:

$$Q'_X = \sum_{j=1}^{m} Q'_{x,j}$$

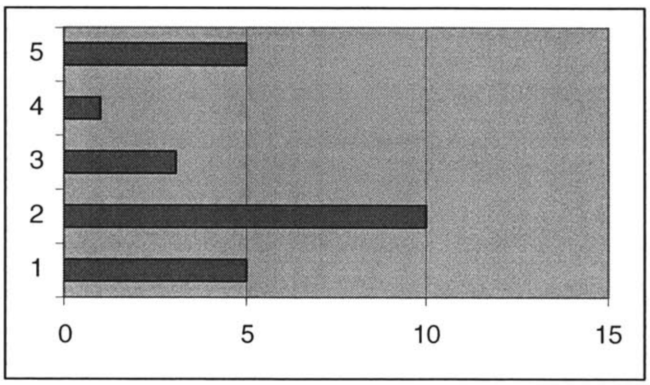
Step 4 - We apply the capacity constraint of tool group X to this estimate. The expected number of wafers to be processed by tool group X becomes:

$$Q_X = \min(Q'_X, C)$$

where C is the capacity constraint of tool X (measured in wafers or kilowatt hours) during time period T. Note that this estimate assumes that there are no capacity constraints imposed by tools upstream of tool group X.

Example:

Tool group X performs Process P_5 (For simplicity, we assume this process only applies to metal 1). Find Q'_x based on the following metal 1 inventory profile:



Process P_i

Inventory in Lots

Figure 3 - Estimating Expected Output based on Inventory Profile

For metal layer 1:
Process step P_5 requires 6 hours of queuing and processing time (i.e. $\tau_{5,1}$=6 hrs).
Process step P_4 requires 2 hours of queuing and processing time (i.e. $\tau_{4,1}$=2 hrs).
Process step P_3 requires 4 hours of queuing and processing time (i.e. $\tau_{3,1}$=4 hrs).
For T = 12 hours, $Q'_x = Inv_{5,1} + Inv_{4,1} + Inv_{3,1} = 5+1+3$ wafer lots = 9 wafer lots

2.2.2 Finding Historical Tool Usage Levels

Each tool has multiple components requiring preventive maintenance. An example of a component is the device that fixes the position of a wafer inside a processing chamber. Components within the same tool may have different preventive maintenance schedules. For example, the type of material used by each component determines its aging characteristics. Also, varying precision levels required of each component within the same tool may also drive towards different preventive maintenance schedules.

For each tool, the binding PM constraint is governed by the component with the least time remaining until PM. For tool i, the binding component's *remaining lifetime* until the next preventive maintenance, denoted by L_i, is measured in wafers or kwh. For each tool type, we find the manufacturer recommended maximum usage between consecutive preventive maintenance events for all components and denote the minimum of these values by U_{MIN}. For the Thin Films area, this minimum value represents the time between consecutive preventive maintenance events since we perform PM on all components simultaneously every U_{MIN} wafers or kwh.

For tool i, the historical usage level h_i denotes the cumulative usage since the last PM, measured by the number of wafers processed or the number of kilowatt-hours consumed:

$h_i = U_{MIN} - L_i$

2.2.3 Ideal End-of-period Usage Levels, t_i

As discussed earlier, the objective of our model is to plan for preventive maintenance events so that they occur as far apart in time as possible. Given the expected output Q_x and the historical tool usage levels h_i, we now calculate ideal end-of-period usage levels for each tool in the tool group. The ideal end-of-period usage of each tool consists of a systematic offset and an integer multiple of the optimal spacing. For N tools ranked in ascending order from 0 to (N-1) by historical usage level h_i:

*End-of-Period Usage Level for tool i = i * optimal_spacing + systematic_offset* where:

$$optimal\_spacing = \frac{U_{MIN}}{N}$$

$$systematic\_offset = \left[\frac{(Q_x + \sum_{i=1}^{N} h_i)}{N} - \frac{N-1}{2} \cdot \frac{U_{MIN}}{N} \right]$$

Please refer to Appendix A for details on the derivation of the systematic offset. In summary, the schedule planning step provides us with ideal end-of-period usage levels taking into account a subset of all constraints. This set of end-of-period usage levels would result in a preventive maintenance schedule equally spaced in time if the number of wafers processed by the tool group is constant.

2.2.4 Maximum End-of-Period Usage Levels, p_i

If tool i has "UP" status and if no PM is due during the next T hours, the maximum output for tool i is the per shift capacity defined in the tool specifications. We approximate its maximum end-of-period usage level (p_i) by summing its historical usage level (h_i) and its per shift capacity. If a tool has "UP" status and a PM is due during the next T hours, we measure p_i by adding the number of wafers remaining until the next PM to the tool's historical usage (h_i). The maximum output (*MaxOutput*) of the tool group is given by:

$$MaxOutput = \sum_{i=1}^{N} (p_i - h_i)$$

2.3 Optimization Step

Given the ideal and maximum end-of-period usage levels from Section 2.2, we now proceed to optimize WIP allocation for tool group X subject to all relevant constraints.

2.3.1 Objective Function

To allocate WIP among i tools such that the difference between actual and ideal end-of-period usage levels is minimized, we define the objective function as follows:

$$Min\sum_{i=1}^{n}[u_i(p_i - a_i - t_i)]^2$$

where:

- u_i = weight representing relative importance of each tool
- p_i = maximum end-of-period usage level for tool i (from Section 2.2.4)
- a_i = excess capacity allocated to tool i for current shift
- t_i = Ideal end-of-period usage level for tool i (from Section 2.2.3)

Subject to constraints:

- $\sum_{i=1}^{N} a_i = MaxOutput - Q_x$

where *MaxOutput* is defined in Section 2.2.4.

One way to assign u_i, the relative importance weight of each tool, is to rank by historical usage.

Example:

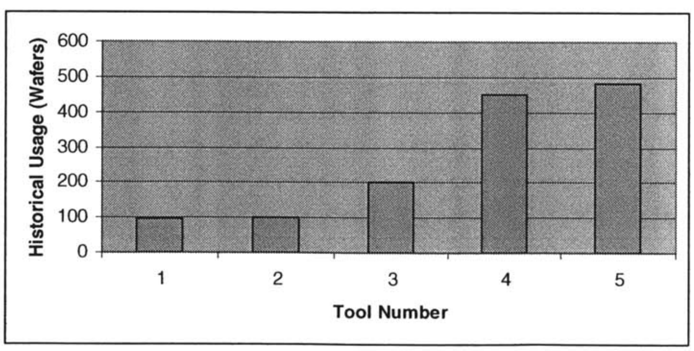


Figure 4 - Historical Usage and Relative Importance u_i

If the PM threshold for all tools is 600 wafers and WIP is equally allocated among all tools, PMs on tools 4 and 5 will be performed simultaneously, with the undesirable effect of lowering equipment availability for the tool group to 60%. If there is excess capacity during the current shift such that we can allocate WIP preferentially, we would assign maximum load to tool 5 and minimum load to tool 4.

A similar argument also applies to tools 1 and tools 2. However, priority should be given to correcting the PM schedules of tools 4 and 5, since PM events for these tools are imminent. The relative importance factor u_i in the objective function is designed to enforce such priority. The PM schedules for tools 1 and 2 can be corrected more slowly over time since they are last in line for PM. If relative tool importance is ranked by historical usage, the following assignments will be made: $u_1 = 1$; $u_2 = 2$; $u_3 = 3$; $u_4 = 4$; $u_5 = 5$. Minimization of the objective function using these weights will lead to a PM schedule that places more emphasis on tools closer to PM.

2.3.2 Key Assumptions

Our model makes the following assumptions:

1. The model assumes continuous availability of certified technicians to perform PMs at any time during a shift. In practice, not every technician is certified to perform PMs in the functional area. Also, some technicians are cross-trained and certified for multiple tool types. Hence their availability to perform PMs may be affected by tool breakages at other functional areas.

2. The model does not explicitly account for capacity constraints in upstream processes. However, capacity constraints can be accounted for if they are reflected in the classification of upstream inventory i.e. via increases in the "inventory-on-hold" category.

3. If a tool will be taken down for PM any time during a shift, our model assumes that it will remain unavailable for the remainder of the shift. This assumption simplifies capacity calculations. This is a valid assumption for the Thin Films area, where the time needed to perform PM is slightly less than the duration of a shift.

4. Other planned events impact equipment availability. These include routine equipment self tests. However, the amount of time needed to perform such routines is small relative to the duration of a shift. Hence we assume that such events have negligible impact on PM scheduling.

2.3.3 Optimization techniques

Given the objective function and constraints defined above, we use dynamic programming to identify the optimal solution.

Dynamic Programming Overview

A dynamic programming (DP) formulation involves breaking the problem down into stages with associated states for each stage. Provided that we can define a recursive relationship between consecutive stages, the optimal decision for stage i can be found if we know the solution to stage i+1. Like other recursive techniques, the final stage must also be solvable. In formulating the problem of excess capacity allocation as a DP, we follow the following steps:

1. Divide the excess capacity allocation problem into N stages – Each stage represents one tool in the tool group.

2. Define a state variable y to represent unallocated excess capacity in remaining stages.

3. Define a control variable a to represent excess capacity (measured in kwh) allocated to each stage.

4. Define the recursive step to link consecutive stages:

$$f_i(y) = \min_a \{ f_{i+1}(y - a_i) + [u_i(p_i - a_i - t_i)]^2 \}$$

f(y) represents the optimal cost function for tools i, i+1, ...N, when there are y units of excess capacity available for these tools and this capacity is optimally allocated to the tools. At any given stage, we do not require knowledge of how excess capacity was allocated in previous stages. We only need to know the total remaining excess capacity in order to derive the optimal allocation for the current stage.

Figure 5 shows an example with model inputs (tool usage level, tool status, excess capacity), objective function weights (relative importance weights u_i) and model outputs (loading level for each tool).

# of Tools		7	# Available		5				
Total Capacity		380							
Target Output		200							
				Relative Importance Weight u = 3 for Tool 707			*Relative Importance Weight u = 0 for Tool 707*		
Tool Name	**Current Usage**	**Reference**	**Status**	**Loading Levels**	**Weight u**	**End-of-Period Usage**	**Loading Levels**	**Weight u**	**End-of-Period Usage**
701	499	578	UP	**76**	6	575	**76**	6	575
702	129	178	DOWN	**0**	2	129	**0**	2	129
703	79	78	DOWN	**0**	1	79	**0**	1	79
704	607	678	UP	**76**	7	683	**71**	7	678
705	353	378	UP	**39**	4	392	**25**	4	378
706	481	478	UP	**9**	5	490	**0**	5	481
707	300	278	UP	**0**	3	300	**28**	0	328

Figure 5 - Loading Level and Relative Importance Weight u_i

The "Reference" column represents t_i, the ideal end-of-period usage level for tool i (from Section 2.2.3). The maximum loading level for each tool is limited by a capacity limit of 76. Note that Tool 707 has a beginning-of-period usage level of 300. This is already higher than the ideal end-of-period level of 278. We see that if the objective function weight for Tool 707 is changed from 3 to 0, the loading level changes from 0 to 28, causing Tool 707's end-of-period usage level to further deviate from the ideal. Thus lowering the relative importance factor has the effect of allowing the end-of-period usage level to deviate further away from the ideal reference level. Therefore, choosing a relative importance weight u of 0 for Tool 707 would be inappropriate under normal operating conditions.

2.4 Performance Tracking and Other Issues

Our model addresses the issue of accountability by defining a score to measure the "quality" of a PM schedule. The score used can simply be the value of the objective function defined in Section 2.2. When unexpected events occur in the tool group e.g. tool breakage or unexpected fluctuations in WIP profile, our model can be re-run easily to generate updated recommendations for WIP allocation. This will allow technicians to respond quickly to recover from unexpected deviations from the desired PM schedule.

CHAPTER 3

IMPLEMENTATION

Chapter 3 - Implementation

In this chapter, we will describe the implementation of the usage-driven PM scheduling model. First, we will focus on prototype development and customization of the model to the Thin Films area at Intel's Hudson facility. We will then outline subsequent work on developing a production application with the Automation group.

3.1 Thin Films Area Overview

Thin Films systems are used to form metallic interconnects. Interconnect characteristics are critical to high speed semiconductor circuits including microprocessors and communications systems. Interconnects affect signal bandwidth and circuit reliability. While we will not discuss Intel's Thin Films manufacturing processes specifically, general Thin Film preparation methods may include physical vapor deposition, chemical vapor deposition and non-vacuum based deposition.

The Thin Films area at Intel's Hudson facility operates 7 days a week, 24 hours a day. The area is staffed by 12-hour shifts. The functional area we focused on had 7 identical tools. The characteristics of each tool are summarized below:

- Each tool has two processing chambers.
- Each chamber has two major components requiring preventive maintenance.
- Periodically, automated self tests called "TestFires" are conducted to evaluate tool status and identify potential problems.
- On average, the processing and queue time in the Thin Films area is approximately 6 hours.

3.2. Excel Prototype

3.2.1 Background

During prototype development, interviews were conducted with technicians in the Thin Films area. The interviews provided information on the number of upstream process steps to be used in estimating expected output (Q_x). Also, there were ongoing efforts to extend the usage threshold between consecutive preventive maintenance events. Hence any application addressing preventive maintenance in the Thin Films area should be easily modifiable to reflect ongoing changes.

3.2.2 Prototype Characteristics

I. Data Sources

Data from the manufacturing floor is collected real time via Excel links to the following data sources.

(a) Station Controller Log Files from the Thin Films area are updated every 10-15 minutes and provide the following information on each tool:

- Tool status: "Up-To-Production", "Down" or "In Preventive Maintenance".
- For each critical equipment subcomponent, cumulative usage since the last PM event is measured in kilowatt hours.
- The update time of the log file provides additional information on tools with "Down" or "In PM" status.

(b) A legacy database provides real time inventory levels at every process step in the Hudson manufacturing facility. This includes classification of inventory as active inventory or inventory-on-hold.

II. Scheduler

The schedule planning step was implemented using Excel spreadsheet functions. The optimization step was implemented using Visual Basic. See Appendix A for the dynamic programming implementation. Visual Basic can be easily integrated with Excel in the form of macros or with other applications in the form of VB script.

III. User Interface

The prototype consists of two user interfaces - one for the administrator and one for technicians on the manufacturing floor.

Wafer-Based PM Scheduler

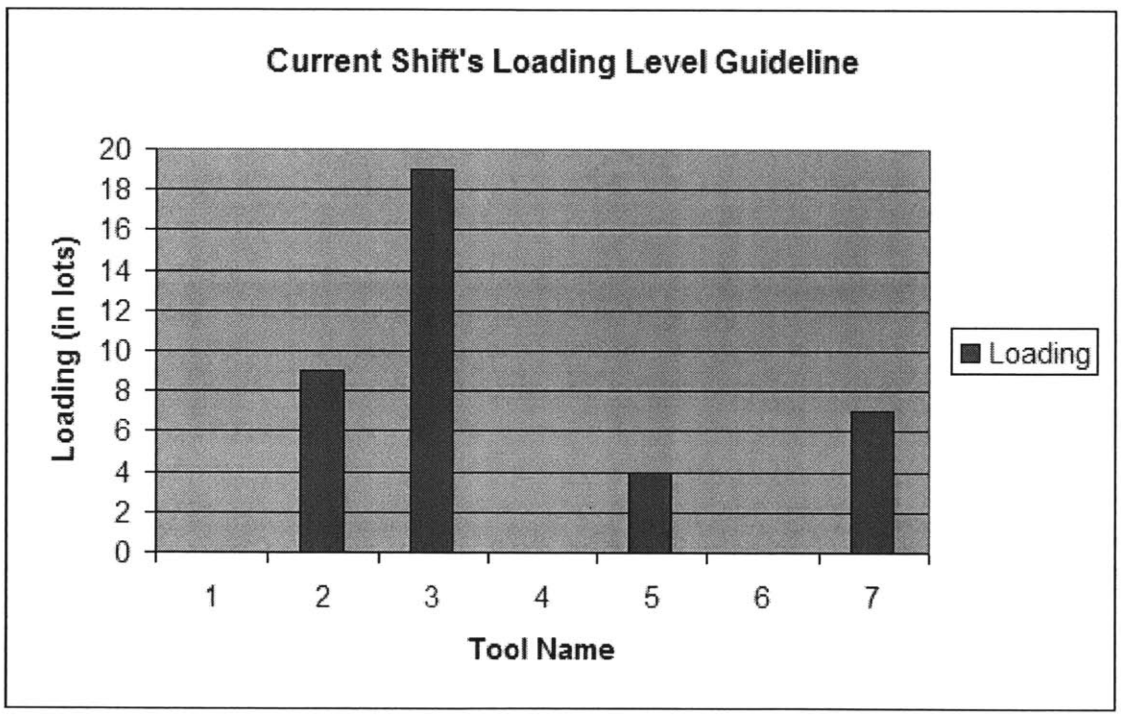
Usage Limit (kwh)	800				
Capacity/ Tool/Shift (kwh)	76.032				
# of Tools	7	# Available	6	6hr Inventory(in kwh)	170
Total Capacity	456				
Target Output	162				
Excess Capacity	294				

Tool Name	Current Usage Level	Goal (Ideal)	Current Shift's Target	Loading Levels	Loading Priority	Status	Estimated Loading(wfrs)	Rank Current Usage
701	632	726	632	0	7	DOWN	0	7
702	348	383	387	39	2	UP	243	4
703	425	497	501	76	1	UP	475	5
704	280	269	280	0	6	UP	0	3
705	596	611	614	18	4	UP	112	6
706	209	154	209	0	5	UP	0	2
707	28	40	57	29	3	UP	181	1
Total	2518	2680	2680	162				

Figure 6 - Administrator Interface of Usage-driven PM Scheduler Prototype

The administrator interface consists of summary information for the functional area as well as for each tool. Parameters such as tool capacity per shift and PM usage thresholds are user inputs and can easily be modified to reflect ongoing changes. Other parameters such as tool status and cumulative usage levels are directly linked to data sources. The "Goal (Ideal)" column shows the results of the schedule planning step. The "Current Shift's Target" column shows the results from optimizing the objective function. The "Rank Current Usage" column ranks each tool's cumulative usage since the last PM. These ranks are used as values for the relative importance factor u_i.

4/13/2004 16:35



Tool Name	701	702	703	704	705	706	707
# Lots	0	9	19	0	4	0	7
Lots to TestFire	62	18	62	46	15	2	60
Lots to PM	42	113	93	130	51	147	193
Tool Status	DOWN	UP	UP	UP	UP	UP	UP

Figure 7 - Shop Floor User Interface of Usage-driven PM Scheduler Prototype

The shop floor user interface consists of a graph showing the recommended loading levels to best achieve a balanced usage-driven PM schedule. Note that a wafer lot typically consists of 25 wafers. The "Lots to Test Fire" row in Figure 7 warns technicians of upcoming automated self test events. The "Lots to PM" row warns technicians of upcoming preventive maintenance events for each tool.

3.3. Challenges Encountered during Prototype Deployment

The following is a summary of challenges encountered during prototype deployment in the Thin Films area.

- Some technicians wanted the system to generate an explanation for each loading recommendation. This challenge is addressed in the rule-based system proposed in Section 4.1.
- Some users expressed preference for a ranking system specifying loading priority instead of loading levels. They felt that the manufacturing floor was too dynamic and that adhering to specific loading levels would be difficult.
- When the model did not recommend allocating wafers to the tool closest to PM, there were questions from the shop floor regarding the rationale behind the system's decision. Through discussing such feedback with technicians, I found that the mentality to get PMs done "as soon as possible" prevailed. While this mentality may at times be justified by labor constraints, it may over the long run lead to systematic biases in the PM schedule.

3.4 Prototype Implementation Results

Figures 8(a) and 8(b) show the occurrence of PM events in the Thin Films area before and after prototype rollout. The start times of each PM are shown during the period August 1-August 28 and during the period October 25-November 21.

	Aug-1		Aug-2		Aug-3		Aug-4		Aug-5		Aug-6		Aug-7	
Shift	D	N	D	N	D	N	D	N	D	N	D	N	D	N
TOOL1		1				1					1			
TOOL2		1												1
TOOL3						1			1					
TOOL4		1					1				1			
TOOL5				1					1					
TOOL6			1									1		
TOOL7	1													

	Aug-8		Aug-9		Aug-10		Aug-11		Aug-12		Aug-13		Aug-14	
Shift	D	N	D	N	D	N	D	N	D	N	D	N	D	N
TOOL1				1									1	
TOOL2											1			
TOOL3	1					1							1	
TOOL4								1						
TOOL5					1									
TOOL6					1							1		
TOOL7	1													

	Aug-15		Aug-16		Aug-17		Aug-18		Aug-19		Aug-20		Aug-21	
Shift	D	N	D	N	D	N	D	N	D	N	D	N	D	N
TOOL1										1				
TOOL2		1												
TOOL3						1								
TOOL4	1						1							
TOOL5		1												
TOOL6				1					1					
TOOL7														

	Aug-22		Aug-23		Aug-24		Aug-25		Aug-26		Aug-27		Aug-28	
Shift	D	N	D	N	D	N	D	N	D	N	D	N	D	N
TOOL1							1							
TOOL2		1					1							
TOOL3						1				1				
TOOL4		1						1						
TOOL5	1								1					
TOOL6	1													
TOOL7	1													1

Figure 8(a) - PM events in the Thin Films area before prototype development

	Oct-25		Oct-26		Oct-27		Oct-28		Oct-29		Oct-30		Oct-31	
Shift	D	N	D	N	D	N	D	N	D	N	D	N	D	N
TOOL1											1			
TOOL2				1										
TOOL3					1								1	
TOOL4						1								
TOOL5		1												1
TOOL6										1				
TOOL7	1											1		
	Nov-1		Nov-2		Nov-3		Nov-4		Nov-5		Nov-6		Nov-7	
Shift	D	N	D	N	D	N	D	N	D	N	D	N	D	N
TOOL1									1					
TOOL2		1												
TOOL3					1								1	
TOOL4					1									
TOOL5														1
TOOL6												1		
TOOL7			1											
	Nov-8		Nov-9		Nov-10		Nov-11		Nov-12		Nov-13		Nov-14	
Shift	D	N	D	N	D	N	D	N	D	N	D	N	D	N
TOOL1										1				
TOOL2	1													
TOOL3				1									1	
TOOL4							1							
TOOL5														1
TOOL6											1			
TOOL7	1									1				
	Nov-15		Nov-16		Nov-17		Nov-18		Nov-19		Nov-20		Nov-21	
Shift	D	N	D	N	D	N	D	N	D	N	D	N	D	N
TOOL1								1						
TOOL2			1											
TOOL3				1						1				
TOOL4						1								
TOOL5														
TOOL6									1				1	
TOOL7		1									1			

Figure 8(b) - PM events in the Thin Films area after prototype rollout

Each calendar day is divided into day and night shifts, denoted by the letters "D" and "N" respectively in Figure 7. The start time of a PM is defined by the transition of tool status into the "IN-PM" state and was obtained from the Thin Films area equipment event log. The August 1-August 28 period was chosen because this is when the usage-driven PM project began. When prototype development began, the Thin Films area did not have a systematic approach for planning usage-driven PMs. The lack of systematic planning and the large number of PM events during this period contributed to multiple tools being brought down for PM within the same shift. Performing multiple PMs during the same shift is undesirable because it lowers tool availability, as defined in Section 1.2, and may cause a functional area to become a factory's capacity constraint. Hence the number of shifts that need to perform multiples PMs is a quality measure for PM planning decisions. During the period August 1-August 28, 11 out of 56 shifts had to perform PMs on more than one tool. The prototype was developed and rolled out in September and October. After prototype rollout, 3 out of 56 shifts had to perform PMs on more than one tool during the period October 25-November 21.

Note that the smaller number of PM events during this period also contributed to the decrease in the number of shifts that had to perform multiple PMs. There were 46 PM events during August 1-August 28, and 36 PM events during the October 25 – November 21 time frame; the difference in the total number of usage-driven PM events during the two time periods is most likely due to differences in cumulative usage in the Thin Films area. Even if we add 10 randomly distributed PMs during October 25 - November 21 such that the total number of PMs during the two time periods were equal, the number of shifts that would have to perform multiple PMs would still be lower during October 25-November 21.

3.5 Economic Benefit Assessment

First, we assume that a functional area becomes the factory bottleneck if tool availability, as defined in Section 1.2, falls below 85%. For a tool group of 7 tools, every time preventive maintenance is performed on 2 tools simultaneously, tool availability drops to 71%. For each shift where multiple PMs are performed, the impact on factory output during that shift will be (85%-71%) = 14%. For every shift where multiple PMs are performed, monthly productivity is reduced by approximately 14%/ 56 = 0.25%, since there are about 56 shifts per month.

If the number of shifts performing multiple PMs were reduced from 11 per month to 8 per month, productivity will increase by 3*0.25% = 0.75%. Assuming that a facility manufactures 1,000 wafers per week and that the profit from each wafer is $1000, the potential increase in profits is $1000*1000*52*0.75% = $390k per year. This estimate is based on improving PM decisions in one functional area only and assumes that customers' demand for wafers is limited by the factory's production capacity.

Note that the cost and production numbers above are hypothetical since actual cost and production data are confidential. The goal of this section is to provide a methodology for assessing economic benefit.

3.6 Production Application Requirements

Beyond the prototyping stage, a production application was developed in conjunction with the Hudson facility's Information Technology team. The following were additional considerations in choosing a platform for the production application.

Scalability

Since usage-driven preventive maintenance was a requirement in many functional areas, our goal was to develop a scalable application that could be easily customized to other tool groups across the manufacturing facility.

Data Storage

To build a reliable application, data from various sources should be replicated and stored locally. This would minimize the impact of partial network outages. In addition, a data retrieval system should be in place to promote accountability among shifts.

Interfacing with other technology components

Legacy systems were used to store certain operations data at the Hudson facility. Although these systems were scheduled to be phased out within the coming year, full compatibility with these legacy systems would allow rapid deployment of our production application.

System Maintenance

Preventive maintenance requirements, such as usage thresholds, changed as a result of ongoing efforts to improve cost efficiency. Users should be able to update the system easily to reflect changes in PM requirements.

3.7 Production Application Components

Microsoft SQL-Server was selected as the platform for implementation. Computations needed for the scheduling step are performed in SQL during the data retrieval process. For the optimization step, VB script is used. The output of the production application is displayed as a web page accessible throughout the facility. It includes loading level recommendations at the beginning of each shift and loading level recommendations for a rolling 12-hour window. Loading recommendations based on a rolling 12-hour window facilitate recovery from unexpected events such as tool breakage.

Figure 9 - Block Diagram of SQL-Server Implementation

CHAPTER 4

ALTERNATIVE APPROACHES

Chapter 4 - Alternative Approaches

In Chapter 2, we developed a model for preventive maintenance scheduling by defining an objective function and used dynamic programming for optimization. In this chapter, we will explore two alternative approaches towards the design of equipment management systems. In particular, we will discuss classes of expert systems used in manufacturing and other application domains. We will propose how such systems can be used to build a preventive maintenance scheduler.

4.1 Introduction to Expert Systems

The goal of an expert system is to accurately capture, represent and distribute expert knowledge. The design of an expert system consists of three main components:

1. Knowledge representation - in the form of logic, rules, constraints, etc.
2. Inference engine - the underlying reasoning mechanism.
3. Control Structure - coordinates the interaction between the inference engine and the chosen knowledge representation.

4.2 Rule-Based Systems Approach

4.2.1 Introduction to Rule-Based Systems

Rules may be an appropriate form of knowledge representation if the application domain is well-understood and can be summarized in heuristics. Expert technicians often use heuristics to perform scheduling tasks in a dynamic manufacturing environment. In many application domains, heuristics used by experts can be summarized reasonably accurately. Rule-based systems can be updated or expanded easily. This is important since manufacturing best practices within Intel's manufacturing facilities are constantly evolving. Updates are needed when new products/ semiconductor processes are introduced or when best practices from one Intel facility is transferred to another.

In some cases, heuristics can be applied only with a limited degree of certainty. Uncertainty can be built into a rule-based system using certainty factors. For example, a hard constraint may have a certainty factor of 1.0 while a soft constraint may be assigned a value of 0.3. As rules are combined using logical operators and as results are "propagated" to the next level, a set of algebraic rules is needed to combine uncertainty factors. MYCIN, a rule-based system for medical diagnosis, provides an example of how certainty factors can be combined.

4.2.2 System Design

In the following sections, we propose a rule-based system for generating loading preferences. The system block diagram shows the chain of reasoning used to derive each recommendation.

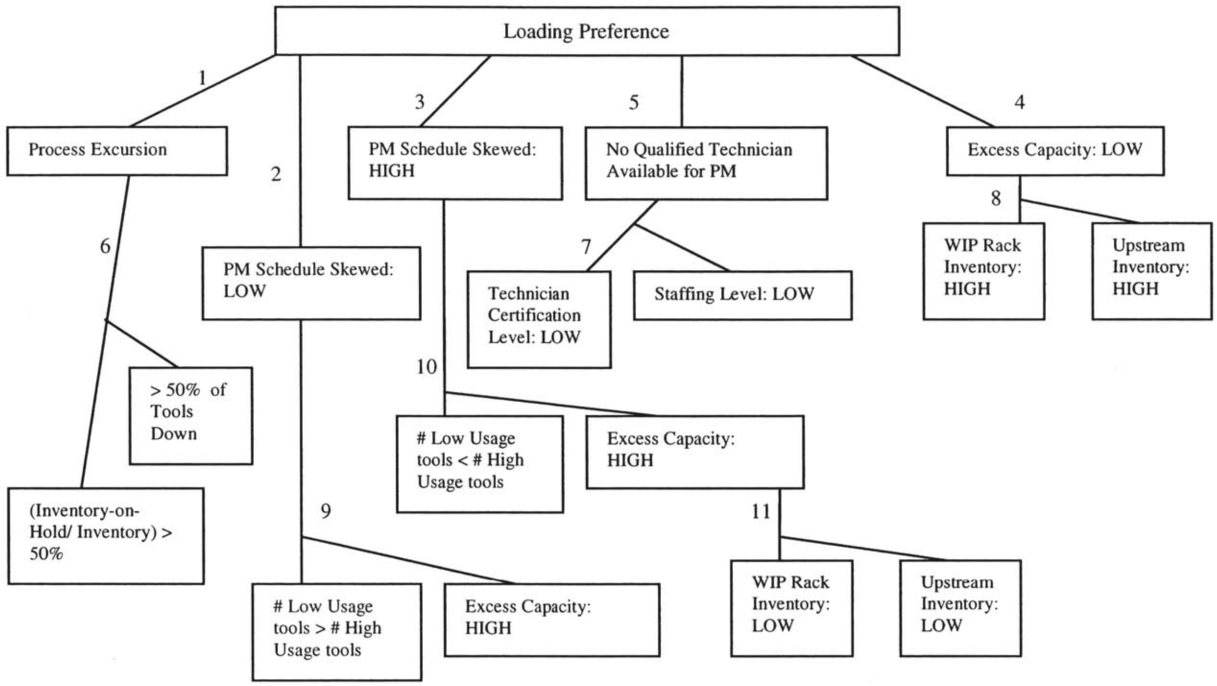


Figure 10 - Block Diagram of a Rule-Based System for Generating Loading Preferences

4.2.3 Implementation

Joshua was chosen for the implementation of this rule-based system. Joshua is an inference language closely integrated with LISP. It consists of five major components:

1. Predications - Assertions/ Statements
2. Database - Stores predications
3. Rules - Define relationships between predications
4. Protocol of Inference
5. Truth Maintenance System - Keeps track of the reasoning process so that explanations behind each recommendation can be given to the user when needed.

An example rule written in the Joshua language is shown below:

```
(defrule excess-capacity-low (:backward :certainty 0.9 :importance 90)
 if [and [wip-rack-level ?who high]
         [upstream-inventory-level ?who high]]
 then [excess-capacity-level ?who low])
```

Please refer to Appendix C for further implementation details.

Program Inputs:

Percentage of tools with HIGH/ MEDIUM/ LOW historical usage, Staffing Level, WIP Rack Level, Upstream Inventory Level, On Shift Technician's Certification Level, % of Tools with "Down" Status, % of inventory on hold

Program Output

The program generates a recommended loading preference summarized by which tools, if any, to load preferentially. As a convenient feature of the Joshua language, the chain of reasoning behind each recommendation and contents in the system database can also be displayed at the user's request.

4.2.4 Comparison

1. A major advantage of the rule-based approach is the availability of customized explanations for each system generated recommendation. Most importantly, the rules used for explanation can be written in the language used by technicians. The importance of customized explanations was highlighted in feedback from technicians during the rollout of the PM Scheduler prototype described in Chapter 3.

2. Since the system is more transparent, technicians will be able to provide actionable feedback to fine tune the system and update values such as uncertainty factors.

3. A robust rule-based system design minimizes interdependency among rules. This enables the system to be easily changeable and expandable.

4. One potential disadvantage of the rule-based approach is the lack of a precise loading recommendation measured in wafers or kilowatt hours. Precise recommendations may not be as important if the manufacturing environment is constant changing as a result of unplanned events.

The following example shows the JOSHUA user interface and illustrates how a rule-based system generates a recommended loading strategy:

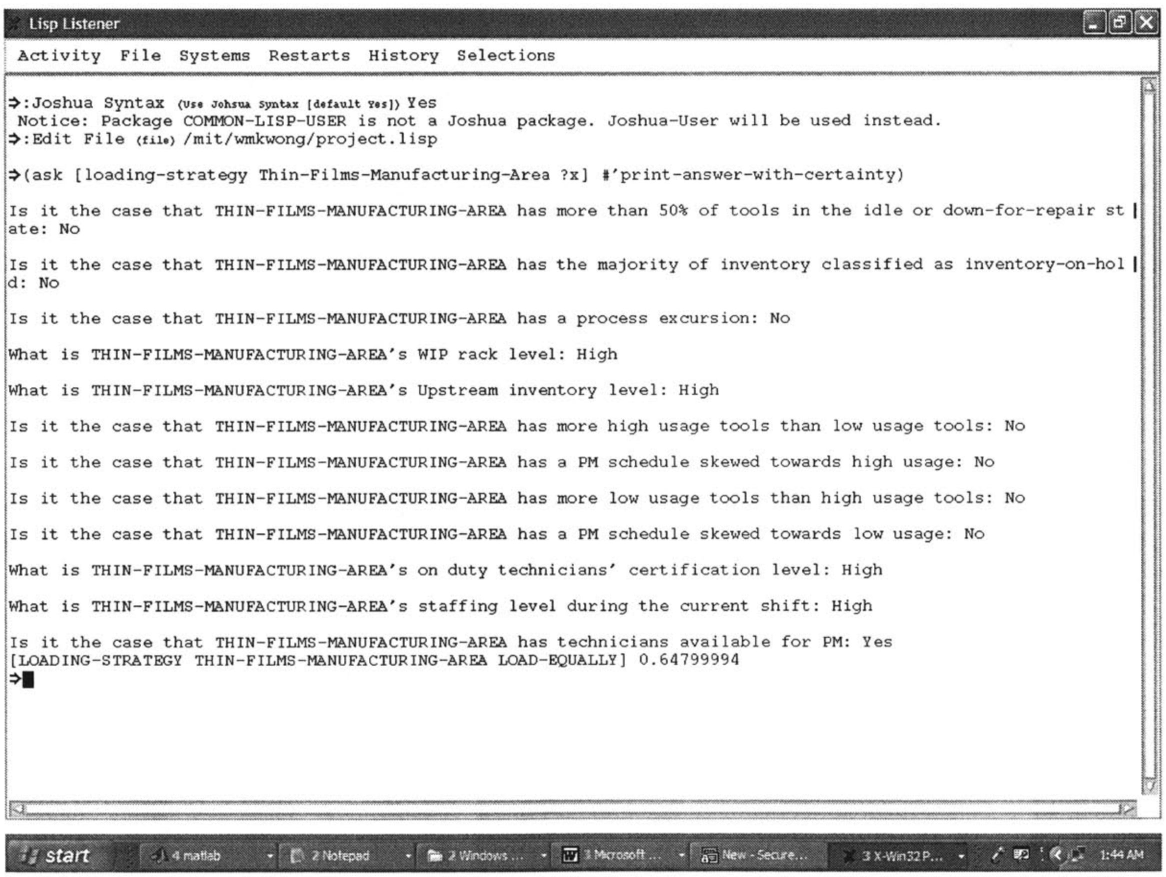
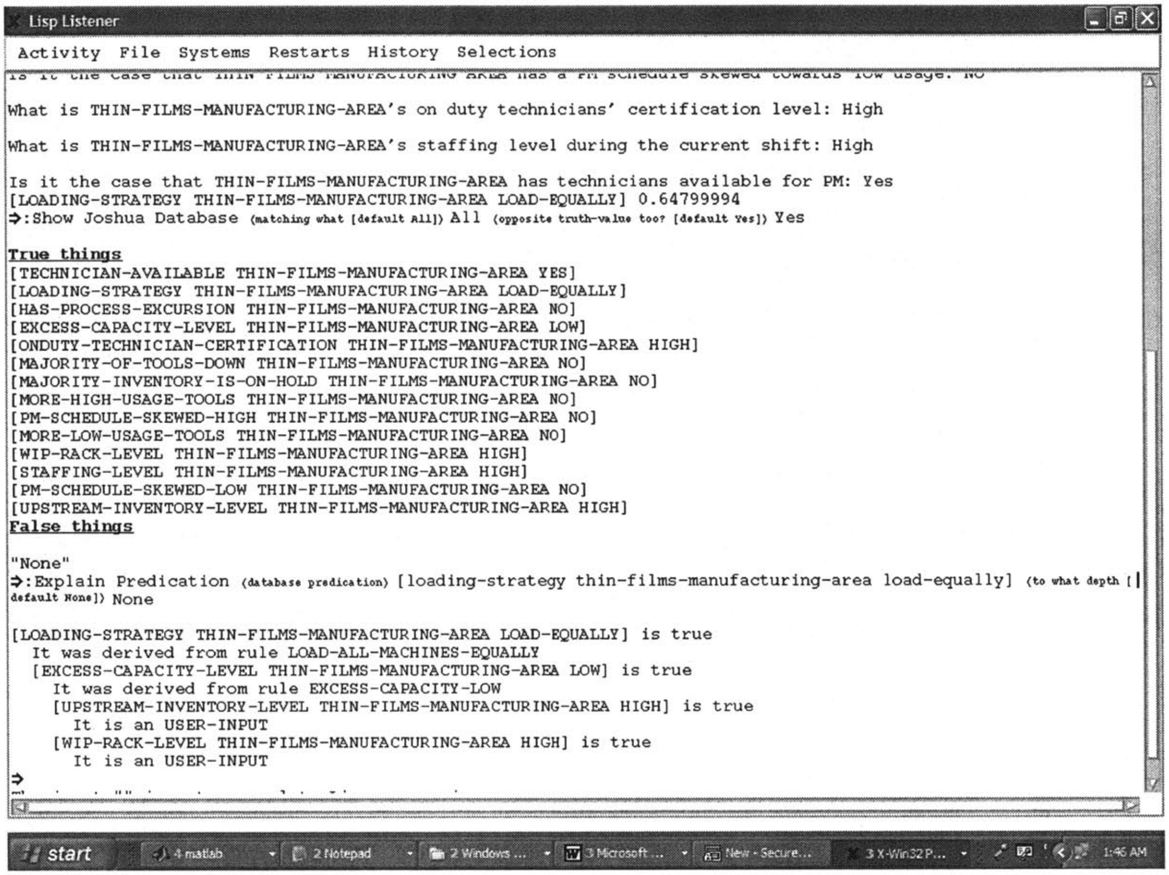


Figure 11 - User Input and Problem-Solving using a JOSHUA Rule-Based System

When excess capacity is low, the rule-based system recommends no preferential loading among tools. Figure 12 shows the logic behind this system recommendation. Specifically, based on the fact that upstream inventory level and current stage's in-process inventory level are both high, the system concludes that excess capacity is low (See Rule 8 in Appendix C). Based on the fact that excess capacity is low, the system recommends that all tools should be loaded with an equal number of wafers (See Rule 4 in Appendix C). The optimization method described in Chapter 2 would also recommend loading all tools equally because there is zero excess capacity to allocate among tools. However, the optimization method cannot generate an explanation backing up the recommendation. In contrast, a rule-based system can have an "explain" function, e.g.":explain predication" in JOSHUA, that shows the user the logic sequence used to derive the recommendation. The "show database" function allows the user to review the system's knowledge base, as shown in the ":show JOSHUA database" function in Figure 12. See Appendix C for the complete set of rules.



```
Activity  File  Systems  Restarts  History  Selections
Is it the case that THIN-FILMS-MANUFACTURING-AREA has a PM schedule skewed towards low usage: No

What is THIN-FILMS-MANUFACTURING-AREA's on duty technicians' certification level: High

What is THIN-FILMS-MANUFACTURING-AREA's staffing level during the current shift: High

Is it the case that THIN-FILMS-MANUFACTURING-AREA has technicians available for PM: Yes
[LOADING-STRATEGY THIN-FILMS-MANUFACTURING-AREA LOAD-EQUALLY] 0.64799994
➔:Show Joshua Database (matching what [default All]) All (opposite truth-value too? [default Yes]) Yes

True things
[TECHNICIAN-AVAILABLE THIN-FILMS-MANUFACTURING-AREA YES]
[LOADING-STRATEGY THIN-FILMS-MANUFACTURING-AREA LOAD-EQUALLY]
[HAS-PROCESS-EXCURSION THIN-FILMS-MANUFACTURING-AREA NO]
[EXCESS-CAPACITY-LEVEL THIN-FILMS-MANUFACTURING-AREA LOW]
[ONDUTY-TECHNICIAN-CERTIFICATION THIN-FILMS-MANUFACTURING-AREA HIGH]
[MAJORITY-OF-TOOLS-DOWN THIN-FILMS-MANUFACTURING-AREA NO]
[MAJORITY-INVENTORY-IS-ON-HOLD THIN-FILMS-MANUFACTURING-AREA NO]
[MORE-HIGH-USAGE-TOOLS THIN-FILMS-MANUFACTURING-AREA NO]
[PM-SCHEDULE-SKEWED-HIGH THIN-FILMS-MANUFACTURING-AREA NO]
[MORE-LOW-USAGE-TOOLS THIN-FILMS-MANUFACTURING-AREA NO]
[WIP-RACK-LEVEL THIN-FILMS-MANUFACTURING-AREA HIGH]
[STAFFING-LEVEL THIN-FILMS-MANUFACTURING-AREA HIGH]
[PM-SCHEDULE-SKEWED-LOW THIN-FILMS-MANUFACTURING-AREA NO]
[UPSTREAM-INVENTORY-LEVEL THIN-FILMS-MANUFACTURING-AREA HIGH]
False things

"None"
➔:Explain Predication (database predication) [loading-strategy thin-films-manufacturing-area load-equally] (to what depth [
default None]) None

[LOADING-STRATEGY THIN-FILMS-MANUFACTURING-AREA LOAD-EQUALLY] is true
  It was derived from rule LOAD-ALL-MACHINES-EQUALLY
  [EXCESS-CAPACITY-LEVEL THIN-FILMS-MANUFACTURING-AREA LOW] is true
    It was derived from rule EXCESS-CAPACITY-LOW
    [UPSTREAM-INVENTORY-LEVEL THIN-FILMS-MANUFACTURING-AREA HIGH] is true
      It is an USER-INPUT
    [WIP-RACK-LEVEL THIN-FILMS-MANUFACTURING-AREA HIGH] is true
      It is an USER-INPUT
➔
```

Figure 12 - An Example Application of the JOSHUA Predicate Explanation Function

The set of rules defined in Appendix C can easily be modified or expanded upon. For example, more specific labor constraints can be added. From this example, we see that Rule-based Systems approach may be preferable when precise loading levels are not needed.

4.3 Case-Based Reasoning Approach

4.3.1 Introduction to Case-based Reasoning

Case-based reasoning uses a library of examples from previous experience to address new problems. Cases provide the context for evaluating potential solutions. They allow the reasoning system to avoid past mistakes and can be applied to planning, diagnosis and design tasks. The library of cases constantly evolves based on user feedback and as new cases are added. While the concept of case-based reasoning is straight-forward, the automation of case retrieval and "knowledge matching" is considerably more involved. Case-based reasoning is a good choice of paradigm when the application domain is not well understood and few generalizations can be made.

4.3.2 Characteristics of the Case-based Reasoning Framework

Aamodt and Plaza (1994) use the following steps to represent the case-based reasoning process:

1. Retrieve - Based on an index of cases and matching knowledge
2. Reuse - Adapt old cases and suggest solution
3. Revise - Verify solution and revise if necessary
4. Retain - Save current case in library for future use

4.3.3 Example Application

One notable application of Cased-Based Reasoning (CBR) Systems to manufacturing is the CLAVIER system used by Lockheed's Sunnyvale Aircraft Composite Fabrication Facility. CLAVIER is used to find the optimal loading configuration of parts inside an autoclave. Prior to CLAVIER, placement of parts was done by expert technicians. Even when performed by experts, the task required a considerable amount of trial and error. To reduce both decision time and the number of scrapped parts, CLAVIER incorporates cased-based reasoning within a complete data management system on the shop floor. It stores previous cases indexed by loading configuration and part number and provides technicians with both successful and unsuccessful references from the past to help them decide on new loading configurations.

4.2.4 Proposed Design for Preventive Maintenance Scheduling

1. Record preventive maintenance scheduling decisions by expert technicians and the corresponding results. Representative cases should be retained in the case library.
2. Within the case library, cases should be indexed by key parameters such as upstream WIP profile and historical equipment usage profile.
3. When a less experienced technician is unsure about how to allocate load, an automated interface allows the technician to access similar cases and decisions made by expert technicians in the past. The indexing scheme will ensure that the cases returned from the library will have similar key characteristics.

4.3.5 Comparison

1. Case-based reasoning systems are widely used in application domains ranging from manufacturing to medical diagnosis.
2. These systems are easily expandable and can be understood by users with a variety of skill level.
3. While indexing and case retrieval in CBR systems can be automated, the application of previous cases to solve new problems requires much more technician involvement compared with other systems. Also, different interpretations by different users may lead

to inconsistent results across shifts. This may limit the systems' effectiveness in reducing variability in equipment availability.

CHAPTER 5

CONCLUSION

Chapter 5 - Conclusion

The project was successful in providing technicians with a framework for usage-driven preventive maintenance planning. Specifically, the framework involves using capacity allocation among tools to control preventive maintenance schedules. An application was developed to standardize decision processes related to usage-driven PM planning.

5.1 Key Lessons on the Design of Automated Decision Systems for Manufacturing

We propose two roles for automated decision systems in a manufacturing environment. The first role involves using optimization to achieve precision levels beyond those attainable by the heuristics of expert technicians. The design and implementation of an optimization method for preventive maintenance scheduling is an example of such a role. The second role involves representing the knowledge of expert technicians and distributing the expert's knowledge across multiple functional areas and across multiple shifts. The rule-based system proposed in Section 4.2 is an example of a knowledge representation and inference system designed for a manufacturing environment. While the effectiveness of each role depends on many factors, the following are key issues to consider in searching for an optimal approach:

- Shop floor attitudes towards automated systems and desired level of complexity.
- Frequency of unexpected events and resulting practical limitations on the precision level of any system recommendation.
- Whether enough is known about the knowledge domain for a representative model to be built for mathematical optimization.
- Balance between application complexity, scalability and ease of maintenance.

- Average experience level of technicians and workforce turnover.

- Level of commitment from information technology team.

5.2 Findings on Organizational Processes: Three perspectives

Findings on organizational processes during the internship will be discussed in the following three perspectives: strategic design, political and cultural.

5.2.1 Strategic Design

The manufacturing organization at Intel consists of a network of semiconductor manufacturing facilities, or "fabs". The "virtual factory" concept makes performance metrics such as yield and unit costs readily available for comparison across different fabs. The Business Operations and Systems group I worked with focused on improving such metrics at Intel's Hudson fab. This group ensures that the Hudson fab remains a competitive member of the Intel manufacturing network. The "virtual factory" structure allows factories to benchmark against each other and to identify areas for knowledge sharing.

At the corporate level, a major component of Intel's manufacturing strategy is to "Copy Exactly". Variability across manufacturing facilities is minimized by efforts ranging from using identical equipment to promoting "Best Known Methods" to transfer knowledge. Within each fab, much focus is placed on minimizing variability in inventory and throughput time. In the Hudson fab, the Business Operations & Systems (BOS) group works closely with shift managers and area coordinators to identify and eliminate sources of variability. The flat organizational structure of the facility and BOS group members' cross-functional experience facilitates such efforts.

5.2.2 Political

The Hudson fab has the most interesting history among Intel's manufacturing facilities. It was acquired by Intel from Digital Equipment Corporation in the late 1990s. Today, a sizeable percentage of the current workforce are former Digital Equipment Corporation (DEC) employees. Understanding the history of work relationships, especially the informal networks dating back to the DEC days, proved helpful throughout the internship. The various stakeholders directly or indirectly involved with the project are shown in Fig. 13.

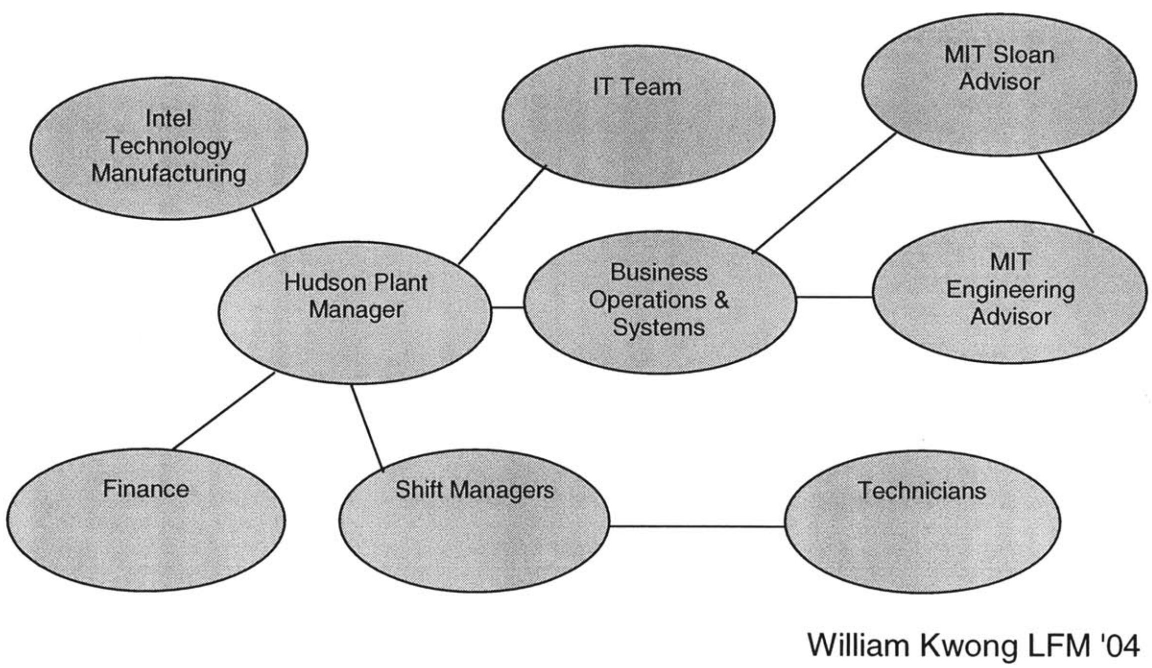


Fig 13 - Stakeholder Map

5.2.3 Cultural

To the technicians on the shop floor, this project was one of many initiatives led by the operations team to improve productivity and reduce cost. The change process in the project involved motivating a more disciplined approach towards equipment management. Driving this change initiative involved understanding the culture at the Hudson facility and building support from technicians, the technology team, engineers and operations/ planning staff.

To understand the work culture, I conducted interviews with technicians, engineers, shift managers and operations staff at the Hudson facility. Most preferred quick and simple solutions to manufacturing problems and some were cautious about automated systems. Information from these interviews guided me in the system design process and throughout the internship.

Participation in Intel Hudson's Very Long Range Planning team provided the opportunity to learn about plant management's long term goals and interactions among different fabs at the Intel corporate level. This helped put my internship project in perspective and provided me with the context for the facility's ongoing initiatives to reduce costs and improve operational efficiency.

5.3 Areas for Future Research

5.3.1 Preventive Maintenance in a High Mix Low Volume Environment

Semiconductor manufacturing for communications applications involves a much broader range of products and lower volumes compared with microprocessor manufacturing. As Intel continues to expand its communications business, managing equipment in a high mix low volume manufacturing environment will present a new set of challenges. Additional sources of variability in equipment availability include the order in which different products are manufactured and the different re-entrant processes required by each product.

5.3.2 Expanding the Role of Information Technology at Intel Hudson

Information systems at Intel Hudson collect and display operations data for the entire facility. To date, the primary role of such systems has been limited to reporting - providing technicians with well-organized data to make decisions and providing operations staff with summary performance reports and data on problem areas. While the limitations of automated systems in a dynamic manufacturing environment should be recognized, there is still much potential for automated decision systems to improve overall factory performance by providing suggestions to engineers and technicians. An example would be the coordination of PM scheduling among related functional areas to minimize variability in both inventory and equipment availability.

5.3.3 A Probabilistic Model for PM Scheduling

Unexpected tool breakages are not modeled in the PM scheduling methods described in this thesis. For functional areas with a high rate of tool breakage, the probability of tool breakage can be built into the relative importance factor u_i. For example, if unexpected breakages occur very frequently, the relative importance factor of tools with the lowest cumulative usage can be adjusted downwards.

5.3.4 Knowledge Representation and Inference Methods in a Manufacturing Environment

One negative feedback the prototype received on the manufacturing floor was that the system was unable to offer a list of reasons to back up each recommendation. Unlike a rule-based system, the optimization approach chosen for the prototype generated a solution by optimizing an objective function. Finding the optimal knowledge representation and inference method suitable for a manufacturing environment is a topic of ongoing research. The combination of different knowledge representation and inference methods is a field of particular interest.

BIBLIOGRAPHY

1. Sutton, Richard 1998, "Reinforcement Learning: An Introduction," MIT Press.

2. Zhou, Ning 2000, "Improving Line Yield at Fab 17," MIT Leaders for Manufacturing Program Thesis.

3. Bailey, Roberta 1996, "A Data-Driven Approach to Improving Capacity Utilization of Semiconductor Test Equipment," MIT Leaders for Manufacturing Program Thesis.

4. Davis R, Shrobe H and Szolovits P 1993, "What is a Knowledge Representation?" *AI Magazine, 14, #1, Spring 1993,* pp. 17-33.

5. Cooke, N.J 1994, "Varieties of Knowledge Elicitation Techniques," *Int. J. Human-Computer Studies,* 41:801-849, 1994.

6. Wellman M. et al. 1992, "From Knowledge Bases to Decision Models," *Knowledge Engineering Review, vol. 6, no. 1, 1992,* pp. 35-53.

7. Sutton R, Barto A 1998, "Reinforcement Learning: An Introduction," MIT Press.

8. Aamodt A, Plaza E 1994, "Case-based reasoning: Foundational issues, methodological variations, and system approaches," AICOM, 7:39-59.

9. Lu S. et al. 1994, "Efficient Scheduling Policies to Reduce Mean and Variance of Cycle-Time in Semiconductor Manufacturing Plants," *IEEE Transactions on Semiconductor Manufacturing, vol. 7, no. 3,* pp.374-385.

APPENDIX

APPENDIX A - Systematic Offset Calculation

The following four steps explain how the systematic offset is calculated. Note that for each tool, historical usage refers to the number of wafers processed since the last PM.

Step 1 - Sum the expected end-of-period cumulative usage levels across all tools. This is given by the sum of the expected number of wafers to be processed in the next T hours and the historical usage levels at the beginning of the time period for all tools:

$$Q_X + \sum_{i=1}^{N} h_i$$

Step 2 - Find the sum of the cumulative usage levels in an ideal PM schedule with zero systematic offset. This can be expressed as the sum of an arithmetic progression with N terms. Since the first term is zero and the last term is $(N-1)\cdot\frac{U_{MIN}}{N}$, the sum of this arithmetic progression is given by:

$$\frac{N(N-1)}{2}\cdot\frac{U_{MIN}}{N}$$

Step 3 - The difference between the results from step 1 and step 2 is divided by the number of tools in the tool group (N) to obtain the systematic offset.

Step 4 - Apply lower bound of zero and upper bound of $\frac{U_{MIN}}{N}$ to systematic offset.

APPENDIX B - VB Dynamic Programming Implementation

```
'Function used for calculating target
Function CBS(n, capacity, c, t, XScap, usageb)
' Precondition:
' Functions returns array of recommended loading levels for each tool
Dim IntResults
Dim i, j, k
Dim alloc  ' alloc(i) = allocated XS capacity for tool i
'Dim s() As Long ' s(i) = score for tool i
Dim fi_y  ' fi_y = MIN score corresponding to state variable at tool i (backward recursion), as fcn
of state variable y
Dim a
Dim yi ' y(i) = state variable y at stage i
Dim stemp
Dim run_sum  ' running sum of excess capacity allocated
Dim minf1_y
Dim invalid
Dim p()

invalid = 10000000

ReDim alloc(n-1)
ReDim p(n-1)

For i = 0 To (n-1)
  p(i) = c(i) + capacity(i)
Next

ReDim IntResults(n-1)
ReDim fi_y(XScap, n-1, 1) ' Store for each stage in 3-D array

' Dynamic Program Description
' Goal: Find optimal allocation of excess capacity across tools 1 to n
'
' Variable definitions:
' a - control variable
' alloc(i) - 1D array stores optimal XS capacity allocation for tool i
' fi_y(,,) - 3D array
'   1st dimension - control variable a (XS capacity allocated to current stage)
'   2nd dimension - tool number
' fi_y(,,0) stores scores and fi_y(,,1) stores XS capacity allocated to stage i
' usageb(i) - 1D array stores relative importance of each tool

' Define fn_y (i.e. final condition)

For a = 0 To XScap
  If (a > capacity(n-1)) Then
    fi_y(a, n-1, 0) = invalid
  Else
    fi_y(a, n-1, 0) = usageb(n-1) * (p(n-1) - a - t(n-1)) ^ 2
```

```
      fi_y(a, n-1, 1) = a
    End If
Next

' Note that XScap may be > or < than cap

i = n - 2
While (i >= 0)
    For k = 0 To XScap
    ' evaluate score of trial solution
      fi_y(k, i, 0) = usageb(i) * (p(i) - t(i)) ^ 2 + fi_y(k, i + 1, 0)
      a = 0

      While (a <= k)
        If (a > capacity(i)) Then
        ' Excess capacity allocated at stage i cannot exceed capacity of tool i
          stemp = invalid
        Else
          stemp = usageb(i) * (p(i) - a - t(i)) ^ 2 + fi_y(k - a, i + 1, 0)
        End If

        ' Find optimal XS capacity allocation for given k
        If stemp < fi_y(k, i, 0) Then
          fi_y(k, i, 0) = stemp
          ' Record optimal XS capacity allocation corresponding to tool i, for given k
          fi_y(k, i, 1) = a
        End If
      a = a + 1
      Wend
    Next
i = i - 1
Wend

alloc(0) = fi_y(XScap, 0, 1)
' Store results in array IntResults
run_sum = alloc(0)
IntResults(0) = p(0) - alloc(0)

For i = 1 To (n-1)
    alloc(i) = fi_y(XScap - run_sum, i, 1)
    IntResults(i) = p(i) - alloc(i)
    run_sum = run_sum + alloc(i)
Next
CBS = IntResults
End Function
```

APPENDIX C - JOSHUA Implementation of Rule-Based System

```
; Predicates for Rules 1 to 5
(define-predicate-with-ancillary-info (loading-strategy value-is-
option-mixin)
    :possesive-suffix "'s" :prompt1 "recommended loading strategy"
:prompt2 "is"
    :possible-values (load-low-usage-first load-one-high-usage-only
load-equally do-not-load-machines))

(define-predicate-with-ancillary-info (pm-schedule-skewed-low value-is-
boolean-mixin)
:possesive-suffix "" :prompt1 "has" :prompt2 "doesn't have" :prompt3 "a
PM schedule skewed towards low usage")

; Rule 6 predicates
(define-predicate-with-ancillary-info (has-process-excursion value-is-
boolean-mixin)
:possesive-suffix "" :prompt1 "has" :prompt2 "doesn't have" :prompt3 "a
process excursion")

(define-predicate-with-ancillary-info (majority-of-tools-down value-is-
boolean-mixin)
:possesive-suffix "" :prompt1 "has" :prompt2 "doesn't have" :prompt3
"the majority of tools in the idle or down-for-repair state")

(define-predicate-with-ancillary-info (majority-inventory-is-on-hold
value-is-boolean-mixin)
:possesive-suffix "" :prompt1 "has" :prompt2 "doesn't have" :prompt3
"the majority of inventory classified as inventory-on-hold")

; Rule 7 predicate
(define-predicate-with-ancillary-info (technician-available value-is-
boolean-mixin)
:possesive-suffix "" :prompt1 "has" :prompt2 "doesn't have" :prompt3
"technicians available for PM")

(define-predicate-with-ancillary-info (staffing-level value-is-option-
mixin)
    :possesive-suffix "'s" :prompt1 "staffing level during the current
shift" :prompt2 "is"
    :possible-values (high medium low))

(define-predicate-with-ancillary-info (onduty-technician-certification
value-is-option-mixin)
    :possesive-suffix "'s" :prompt1 "on duty technicians' certification
level" :prompt2 "is"
    :possible-values (high medium low))

; Rule 8 predicates

(define-predicate-with-ancillary-info (wip-rack-level value-is-option-
mixin)
    :possesive-suffix "'s" :prompt1 "WIP rack level" :prompt2 "is"
    :possible-values (high medium low))
```

```
(define-predicate-with-ancillary-info (upstream-inventory-level value-
is-option-mixin)
    :possesive-suffix "'s" :prompt1 "Upstream inventory level" :prompt2
"is"
    :possible-values (high medium low))

(define-predicate-with-ancillary-info (excess-capacity-level value-is-
option-mixin)
    :possesive-suffix "'s" :prompt1 "Excess capacity level" :prompt2
"is"
    :possible-values (high medium low))

; Rule 9 predicate
(define-predicate-with-ancillary-info (more-low-usage-tools value-is-
boolean-mixin)
:possesive-suffix "" :prompt1 "has" :prompt2 "doesn't have" :prompt3
"more low usage tools than high usage tools")

; Rule 10 predicates
(define-predicate-with-ancillary-info (more-high-usage-tools value-is-
boolean-mixin)
:possesive-suffix "" :prompt1 "has" :prompt2 "doesn't have" :prompt3
"more high usage tools than low usage tools")

(define-predicate-with-ancillary-info (pm-schedule-skewed-high value-
is-boolean-mixin)
:possesive-suffix "" :prompt1 "has" :prompt2 "doesn't have" :prompt3 "a
PM schedule skewed towards high usage")

;;; Rules 1-5 determine loading strategy
; Rule 1
(defrule do-not-load-any-machines (:backward :certainty 1.0 :importance
99)
 if [has-process-excursion ?who yes]
 then [loading-strategy ?who do-not-load-machines])

; Rule 2
(defrule prefer-low-usage-machines (:backward :certainty 0.8
:importance 95)
 if [pm-schedule-skewed-low ?who yes]
 then [loading-strategy ?who load-low-usage-first])

; Rule 3
(defrule load-one-high-usage-only (:backward :certainty 0.8 :importance
95)
 if [pm-schedule-skewed-high ?who yes]
 then [loading-strategy ?who load-one-high-usage-only])

; Rule 4
(defrule load-all-machines-equally (:backward :certainty 0.8
:importance 95)
 if [excess-capacity-level ?who low]
 then [loading-strategy ?who load-equally])

; Rule 5
(defrule defer-pm (:backward :certainty 0.8 :importance 94)
 if [technician-available ?who no]
```

```
 then [loading-strategy ?who load-low-usage-first])

; Rule 6
(defrule process-excursion (:backward :certainty 0.8 :importance 99)
  if [or [majority-of-tools-down ?who yes]
         [majority-inventory-is-on-hold ?who yes]]
  then [has-process-excursion ?who yes])

; Rule 7
(defrule no-technician-available (:backward :certainty 0.7 :importance
99)
if [or [onduty-technician-certification ?who low]
       [staffing-level ?who low]]
then [technician-available ?who no])

; Rule 8
(defrule excess-capacity-low (:backward :certainty 0.9 :importance 90)
 if [and [wip-rack-level ?who high]
         [upstream-inventory-level ?who high]]
 then [excess-capacity-level ?who low])

; Rule 9
(defrule pm-schedule-skewed-low (:backward :certainty 0.9 :importance
80)
 if [and [more-low-usage-tools ?who yes]
         [excess-capacity-level ?who high]]
  then [pm-schedule-skewed-low ?who yes])

; Rule 10
(defrule pm-schedule-skewed-high (:backward :certainty 0.9 :importance
90)
 if [and [more-high-usage-tools ?who yes]
         [excess-capacity-level ?who high]]
  then [pm-schedule-skewed-high ?who yes])

; Rule 11
(defrule excess-capacity-high (:backward :certainty 0.9 :importance 91)
  if [and [wip-rack-level ?who low]
          [upstream-inventory-level ?who low]]
  then [excess-capacity-level ?who high])
```