Equipment Protective Capacity Optimization Using Discrete Event Simulation

by

Anthony W. Newlin

B.S., Engineering Science, Trinity University, 1993
M.S., Chemical Engineering, Colorado School of Mines, 1996

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

Master of Science in Management and
Master of Science in Electrical Engineering

In conjunction with the Leaders for Manufacturing Program
at the Massachusetts Institute of Technology
May, 2000
June 2000

© Massachusetts Institute of Technology, 2000. All rights reserved

Signature of Author ________________________________
MIT Sloan School of Management
Department of Electrical Engineering and Computer Science
May 5, 2000

Certified by ________________________________
Donald B. Rosenfield
Senior Lecturer, Sloan School of Management
Thesis Advisor

Certified by ________________________________
Roy E. Welsch
Professor of Statistics and Management, Sloan School of Management
Thesis Advisor

Accepted by ________________________________
Margaret Andrews
Director of Master's Program, Sloan School of Management

Accepted by ________________________________
Arthur C. Smith
Chairman, Committee on Graduate Studies
Department of Electrical Engineering and Computer Science

MASSACHUSETTS INSTITUTE OF TECHNOLOGY
JUN 1 3 2000
LIBRARIES

ENG

Equipment Protective Capacity Optimization Using Discrete Event Simulation

by

Anthony W. Newlin

B.S., Engineering Science, Trinity University, 1993
M.S., Chemical Engineering, Colorado School of Mines, 1996

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

Master of Science in Management
and
Master of Science in Electrical Engineering

Abstract

Assembly and Test Manufacturing (ATM) at Intel faces new challenges caused by increased competition, cost pressure, and segmented markets. These forces combine to present ATM with increasing line items and processes in the face of extreme demand fluctuations over relatively short time periods. As a result, the factories are challenged with accurately planning capacity.

Currently, ATM utilizes static, Excel-based models to plan capacity and perform what-if scenarios. The applicability of static models in the highly dynamic ATM environment is questionable. These static models neglect the inherent variability of each tool as well as the coupling of variability between tool sets caused by WIP flow. This prevents static models from predicting the values and variabilities of factory outputs and throughput times (TPT) with sufficient accuracy to optimize the business.

Discrete event simulations have the inherent advantage of modeling factory dynamics. They allow for factory experimentation without risking actual production. Examples include availability and run rate improvement impacts, and changes to WIP management policies.

Both static and dynamic approaches share a dependency on the accuracy of the input data. In ATM, a few performance parameters are accurately measured in Workstream including output, TPT, queue size, and yield. Tool performance data (availability, failure details, etc.) are not accurately measured because of the low priority placed on this type of data. Parameters such as utilization are back-calculated instead of being measured directly. No attempt is made to capture other important data like tool idle time.

This thesis explores the development, validation, and application of a full factory simulation including the consequences of data inadequacies. Tool and factory performance data were gathered in the Costa Rica assembly and test factory for WW28-34 1999, and were incorporated into a dynamic factory model. Results from simulation using this model underscored the need for automated tool data collection systems by highlighting the inaccuracies of the tool availability data and labor effectiveness. The model also proved useful for exploring WIP policy alternatives (CONWIP limits vs. drum-buffer-rope starts policies). Reduction of CONWIP limits from 4 days to 3 days appeared robust and generated a 20% decrease in TPT. Equipment protective capacity was optimized. The results indicate that the current gap policy of 10/15/20 is sub-optimal and leads to inefficient capital expenditures. The thesis also shows a logical methodology for optimizing protective capacity levels in factories where there are large capital cost differences among toolsets.

Thesis Advisor: Donald B. Rosenfield
Title: Senior Lecturer, Sloan School of Management

Thesis Advisor: Roy E. Welsh
Title: Professor of Statistics and Management

List of Figures

List of Tables

Acknowledgements

I would like to thank the following people at Intel for the guidance, support, and helpfulness during my internship: Mark Scott, Seth Fischbein, Arelene Say, Robert Serrano, Scott Headley, Enrique Saborio, Tony Leong, Luis Hernandez, and Roy Cristobal.

I would like to extend a special thanks to the four people who were most instrumental in my internship: First of all, thanks to Chris Richard, the project sponsor for his support and advice. Karl Kempf, Intel Fellow, for his guidance and patience. My MIT advisors, Don Rosenfield and Roy Weslch - thanks so much for all of the guidance and insights.

Finally, the Leaders for Manufacturing program provided me amazing opportunities to learn during my time at MIT and I am very grateful for the support and resources of the program.

CHAPTER 1. Introduction and Problem Background

1.1. The Assembly and Test Environment at Intel

In recent years, the microprocessor production environment has rapidly changed. Charles Fine (1998) describes this change as an increase in Intel's clockspeed, which translates into the decreasing product and process lifecycles. Market conditions and increased external competition have caused Intel to address a variety of market segments. This is in stark contrast to Intel's traditional performance as essentially a mass production company.

Over the past decade, Intel essentially produced a single product at a time. Examples of this include the 486 and Pentium® processors. The variation in the product line was the speed, which essentially served to differentiate the market since new, high speed processors cost significantly more than slower speed processors. Intel owned a large percentage of the PC microprocessor market and was able to maintain high profit margins as result. These processors were assembled and tested (A/T) on relatively simple and inexpensive processes.

Over the past 2-3 years, the microprocessor market has become increasingly segmented. Three market segments now exist: value, performance, and server. Intel has reacted by offering a different product for each segment: Celeron® for value PCs, Pentium II/III® for performance PCs, and Xeon® for servers and workstations. In addition to these basic market segments, a strong mobile (laptop) market is developing for each of these segments. Shorter product lifecycles and increased competition in the marketplace have led to gross margin pressure for Intel.

Products for each segment are assembled and tested using different processes. The A/T processes have become more complex and costly and now occupy a larger portion of product cost. As a result of market segmentation, Assembly and Test factories (ATM) now must work with an unprecedented number of products and processes.

Intel is also experiencing the difficulty of trying to predict demand for each of the different market segments. On several occasions, demand has shifted from one segment to another and Intel has not had enough capacity to accommodate the demand changes. Intel has chosen to address this issue through the use of a capacity buffer on all A/T processes to try to prevent A/T capacity constraints in the future. Another way to address this issue is to design products and processes that can be converted between market segments relatively quickly allowing the company to react to market demand changes. Intel has been much slower to act on this front, although recent improvements in product design now allow products to be differentiated by the A/T operation instead of at the time of fabrication. In other words, most of the microprocessors produced by the fabs are identical; the products are differentiated in A/T processes. This allows Intel to react to demand shifts much more quickly, but also creates more complexity for the A/T factories because of the wide array of capacity scenarios that become possible. Although the idea of process fungibility is starting to take hold, the need for processes that can be quickly converted to run products for different segments has not been emphasized enough. For example, performance and server processors are assembled and tested on independent manufacturing lines. A fungible process that could be quickly converted between the two products would help alleviate the capacity challenges faced by ATM.

ATM is a globally dispersed organization with operations in Costa Rica, China, Malaysia, and the Philippines. In contrast to fabrication facilities, ATM factories are relatively labor intensive and much less capital intensive. These conditions warrant factories being located in markets where labor is relatively inexpensive. ATM factories typically contain two sub-factories; assembly and test. WIP flows straight through the factory without any re-entrant flow (re-entrant flow occurs when product is processed by a given toolset more than once during the process). Yields are typically very high and re-work rates low. Typical cycle times for the factories are about 1 week. The major variable in the process is the number of tests that have to be performed on a unit which translates into the total time a unit must spend being tested. Generally, as products mature, test times decrease. Large cost differences exist between factory equipment with testers being the most expensive tool. The variation in test times combined with the high tester cost and relatively complicated operation lead to difficulty in planning factory capacity. Most of the remaining tools in the factory do not experience large variation in processing times between products and product maturing making capacity planning and operational execution an easier task.

1.2. Motivation and Objectives

Because of the relative simplicity and low-cost of A/T processes in the past, the planning and operation of A/T factories has not received much attention. Capacity planning systems and operational policies have not kept pace with the increasing costs and complexity of current A/T processes.

Intel has pushed the concept of Theory of Constraints (Goldratt, 1992) to the next level by adding excess capacity at non-constraint operations to manage factory cycle times and output variability. A blanket capacity planning policy was carried over from fab experience and applied to ATM factories. This policy dictates the amount of excess capacity at all operations is believed to be sub-optimal. The intent of this thesis was to optimize excess capacity levels throughout the factory in order to maximize capital cost avoidance and to develop robust rules that could be applied in a general setting.

It was also believed that the current, Excel-based static capacity models were inappropriate for the highly dynamic ATM environment. An investigation into the usefulness of discrete event simulation (DES) for addressing both tactical and strategic issues faced by the factories was also warranted.

Therefore, the primary objective of the thesis was to optimize excess capacity levels at all operations of the OLGA process (the current desktop segment A/T process). This was to be accomplished through the use of a discrete event simulation. Creation and validation of the simulation required extensive fact finding into all areas of factory operations and naturally led to the application of DES in a wide array of factory operational issues. An assessment of factory data integrity and factory indicators was also incorporated into the scope of the project through the extensive data collection process.

CHAPTER 2. Capacity Planning at Intel Corporation

Intel employs excess equipment capacity to manage output variability and throughput time (TPT) using a methodology known as Gap. Gap is planned idle time (or reserve capacity) at each operation in the line. Before moving forward in an attempt to optimize excess capacity levels at each operation in ATM factories, the history of the Gap policy at Intel needed to be examined. The goal was to expand on historical work in this area in an attempt to prevent reinventing the wheel.

On several occasions, Stan Gershwin (senior research associate, Department of Mechanical Engineering, MIT) has commented on the lack of standardized definition within the manufacturing systems realm. One example he sites is the standard definitions of ohms and volts used by electrical engineers. These units have become widely accepted and allow for comparison between a wide variety of devices. Conversely, no standard definitions exist for basic metrics such as utilization and availability. The lack of a standard metric makes comparisons of factory performance extremely difficult. Even within ATM factories, different definitions of utilization and availability increase. Before exploring the concept of gap, the metrics used in this thesis must be defined.

2.1. Metric Definitions

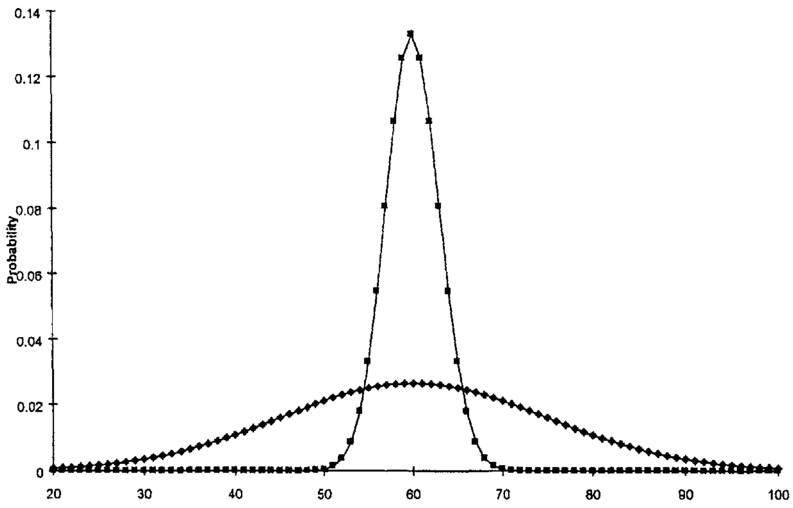
2.1.1. Availability

Availability is the percent of time that a tool is available to run production. Activities that cause a tool to be unavailable are PM's, repairs, assists, conversions, and setups. It should be noted that the activities included in the definition of availability are not absolute but rather agreed upon by management.

Availability is generally expressed as a single number percentage. For example, an IX tester may have an availability of 93%. It is very important to realize that this number is only the average availability. Equally important to the average availability is variation in availability. Figure 2.1 shows two separate tools, both with average availabilities of 60%. However, the distribution of availabilities are very different for each tool. The tool with the larger distribution is a more unstable which significantly adds to the variability of the factory.

The central limit theorem also plays a key role in the variation of availability. This idea basically states that the more tools there are in a given toolset, the lower the variation in total availability for the toolset. Intuitively, this makes sense. A toolset with 20 tools will generally show less variation for the entire toolset than a toolset with 3 tools.

Figure 2.1: Demonstration of Variation in Availabilities



In the following sections, the point will be made that availability is an ignored indicator, although it actually deserves more attention than the current indicators. One of the reasons that it is ignored is because the data are not accurate. Accurately measuring availability is technically involved and is discussed further in Section 3.5.2.

2.1.2. Utilization

Although several definitions of utilization have existed in the past in ATM, utilization can simply be defined as the percentage of time a tool is actually running product. ATM abbreviates this quantity as MU (machine utilization). In order for a tool to be utilized, three conditions must be met:

- The tool must be available
- There must be WIP present for the tool to process
- An operator must be present to run the WIP

If any one of these conditions is not met, the tool can not be utilized. Utilization can never exceed availability, although it can be often be lower than availability due to WIP starvation and/or the lack of an operator.

In some respects, availability is related to utilization, although this relationship is difficult to quantify. As a tool's utilization approaches its availability, the probability of an assist or failure increases which then reduces the availability. Conversely, as utilization becomes much less than availability, the chance for unexpected downtime decreases which allows availability to remain high.

2.1.3. Throughput Time (TPT)

Throughput time is the time it takes for a unit to pass through the entire manufacturing line. TPT is the sum of the total time a part spends waiting in queues added to the total time a part spends being processed. In mathematical terms,

$$TPT = \sum_{\text{all operations}} \text{Processing Time} + \sum_{\text{all operations}} \text{Queue Time} \qquad \text{(Eq. 2.1)}$$

The equation shows that the two ways to reduce TPT are either to reduce queues or to reduce processing time. In ATM factories, the queue times are usually much greater than the processing times which makes WIP management policies a powerful lever for reducing TPT. Although processing times for ATM factories are relatively constant due to automated processing, queue time are highly variable and lead to variation in TPT. Variation in the quantity and duration of lots on hold adds to the overall TPT variation.

The theoretical TPT (the fastest possible TPT) is simply the sum of the processing times for each tool in the line.

2.1.4. Run Rate

Run Rate is the average planned capacity of a tool.

$$\textit{Run Rate (units/week/tool)} = \textit{uph*MU*168 hours/week} \qquad \text{(Eq 2.2)}$$

Where uph = units per hours that a tool can process (without interruption) surprising since maintaining a WIP buffer at the constraint leads to more factory inventory and thus longer TPTs.

Installing additional excess capacity at non-constraint operations was tried as a means to reduce TPT. Excess capacity at non-constraint toolsets leads to expected idle time in these areas. The reasoning is that the excess capacity will reduce queues at non-constraint operations. This reduction in WIP leads to faster TPTs. This strategy was implemented in the fabs and is still evolving today. The fabs face the challenge of several hundred process steps with a high degree of re-entrant flow. These challenges lead to complexities in WIP management and factory design that are much different than the ATM factories.

2.3. The Gap Policy

The methodology of purchasing excess equipment capacity to create planned idle time at toolsets became known as gap. Gap is defined as:

$$Gap = Availability - Utilization \qquad \text{(Eq. 2.3)}$$

As a result of the ToC/excess capacity experiments in the fabs, the 10/15/20 policy was introduced. This means that the constraint toolset should have a gap of 10%, the near constraint(s) a gap 15%, and all other tools (non-constraints) should have a gap of 20%.

Constraints were primarily determined by the toolset cost. The toolset cost is a function of an individual tool's cost and its uph. Thus the implicit policy inherent to 10/15/20 is to have a 10% gap at the most expensive toolset, a 15% gap for other expensive tools, and a 20% gap at the relatively inexpensive tools.

2.3.1. Gap Implementation in ATM

As a result of the dramatic improvements observed in fab operations, constraint management and gap were introduced to ATM in 1995. Initial simulations were run to determine to correct gap for the SPGA factory (Srivatsan et. al., 1995). The starting point for the simulation was the 10/15/20 policy chosen by the fab. The simulation determined that a 10% gap was appropriate for the constraint, a 15% gap for the near-constraint, and 20% for all non-constraint operations. This simulation only consisted of three data points and was simply a starting point for determining ATM capacity policies. Only the constraint gap was optimized; the near and non-constraint gaps were held constant. While the simulation was a good first step, 10/15/20 has become POR (plan of record) in ATM and has not been revisited since 1995.

ATM introduced 10/15/20 and constraint management in the factory with dramatic results. During the first quarter of implementation, output increased by 18% and inventory decreased by 11%. Within a year, TPT decreased by 50% (Kempf et. al, 1998). These results proved the effectiveness of excess capacity and constraint management in an ATM environment.

2.3.2. Variation in the Factory

Before further exploring the gap methodology, factory variation must be visited. Variability is present in all aspects of factory operations. Consider the three requirements for utilizing a tool:

- The tool must be available
- An operator must be present to run WIP on the tool
- WIP must be available to run on the tool

It is easy to see the variation in each one of these conditions. Tools are subject to repairs, PM's, and other sources of downtime making their availability variable. Operators must tend to a variety of tasks making their availability at a particular tool variable. Finally, variation in tool availability and labor availability leads to variation in WIP flows throughout a factory. The end result of all of this variation is unpredictability in factory output and throughput times.

Two options exist for addressing these sources of inevitable variation:

1. Try to eliminate the variation.
2. Manage the variation.

Option 1 is a very costly solution. This option involves expensive engineering solutions

2.3.3.1. TPT Management

TPT management can be easily seen in the following example which develops the TPT-MU trade-off curve. MU is inversely related to idle time: as MU increases and approaches availability, idle time decreases (and vice versa).

Suppose a constant WIP queue of 1600 units is to be processed by a toolset. Each tool can process 160 units per shift. Assuming perfect execution and tool availability, the number of tools in the toolset is varied and the impact on TPT is observed. Table 2.1 shows the results of this example. When 7 tools are operated, the capacity of the toolset is 7 tools * 160 units/tool/shift which equals 1120 units/shift. 1120 of the 1600 units are processed on the first shift and the remaining 480 are processed on the second shift. The total capacity of the toolset is 1120 units/shift * 3 shifts which equals 3360 units. A total of 1600 units were processed which results in a utilization of 48% (1600 units processed/3360 unit capacity). Similarly, 1120 units were completed in 1 shift while 480 units were completed in 2 shifts. The average TPT is a weighted average of these TPTs and equals 1.3 shifts.

Table 2.1: Example Factory Statistics

Queue Size	# of Tools	Toolset Capacity per shift	MU (over 3 shifts)	Average TPT (# of shifts)		Shift 1 units processed	Shift 2 units processed	Shift 3 units processed
1600	7	1120	48%	1.3		1120	480	0
1600	6	960	56%	1.4		960	640	0
1600	5	800	67%	1.5		800	800	0
1600	4	640	83%	1.8		640	640	320

As the number of tools is increasing (while the initial queue size remains constant), MU decreases (idle time increases) while TPT becomes faster. The graphical representation of these data (Figure 2.2) clearly demonstrates this relationship.

The inverse relationship between MU and TPT is clearly shown by the data from the sample factory. In other words, as idle time increases, TPT becomes faster. The makes intuitive sense as well. Idle time at a toolset allows it to immediately process any incoming WIP and minimizes the presence of WIP queues. This in turn leads to faster TPTs. In a more general sense, the MU vs TPT relationship is shown in Figure 2.3.

At some point, MU is low enough (idle time is great enough) that no WIP queues are present in the factory. At this point, the factory TPT is simply equal to the theoretical TPT. As idle time decreases (and MU increases), the TPT increases. Eventually, as the factory becomes overloaded, TPT approaches infinity.

The Gap policy allows Intel to take advantage of this relationship. If the gaps are increased, TPT will decrease (the converse of this is also true). Correspondingly, in order to increase idle time, excess capacity must be purchased. Factory capital cost is therefore inversely related to TPT; TPT decreases as gaps are increased and capital cost increases. Excess capacity requirements need to be driven by TPT targets. Currently, 10/15/20 is simply used and the resulting TPT is accepted. Managers do not appear aware of their ability to influence TPT by capital spending. Furthermore, it is very difficult to quantify the value of faster TPTs in ATM factories. Future efforts in quantify the value of TPT and arrive at TPT targets that make business sense are necessary to optimize any excess capacity strategy.

Figure 2.2: Example Factory Results: TPT vs MU Tradeoff

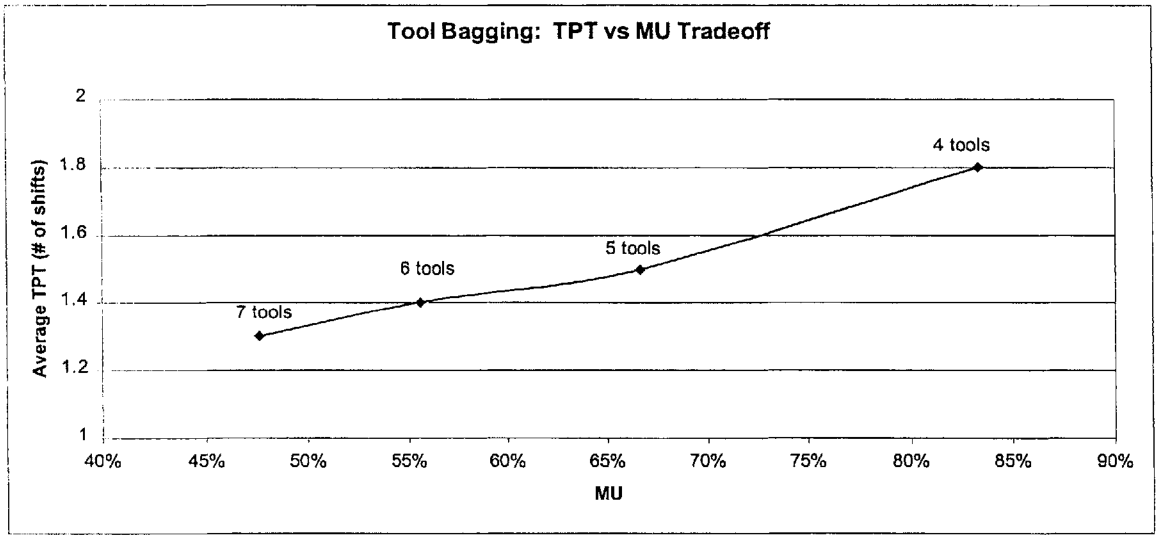
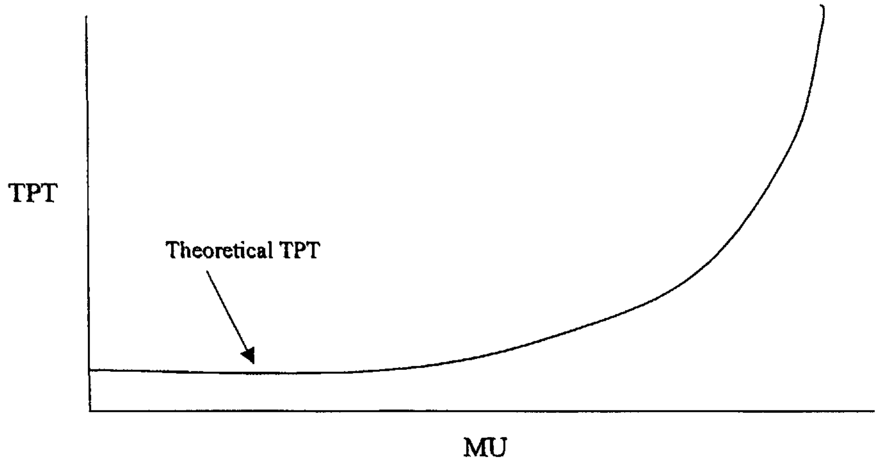


Figure 2.3: MU vs TPT General Relationship



2.3.3.2. Factory Variability Management

The Gap policy of 10/15/20 begs the question of why the gaps are different for constraints, near-constraints, and non-constraints. The answer to this question is revealed upon examination of the components of factory variability.

Remember, three conditions must be met in order to utilize a tool:

- The tool must be available
- WIP be must present at the tool
- An MT must be present to run the WIP

As previously discussed, variation exists in each of these necessary conditions.

Constraint Gap

In theory, the constraint gap is lower than the near and non-constraint gaps because it is only subject to one source of variation. A WIP buffer is maintained in front of the constraint to ensure that it always has WIP available to process. This WIP buffer eliminates variation in WIP at the constraint. Ideally, the constraint does not have variation in operator availability either. Manufacturing should focus its efforts on the constraint tool so that WIP is immediately processed on available tools to maximize factory output. This resource focus should eliminate variation in labor, although in practice this variation cannot be completely eliminated. None the less, the only real source of variability the constraint faces is variation in tool availability.

Non-Constraint Gap

Non-constraint tools are subject to variation in tool availability. However, non-constraints are also subject to variation in WIP and operators which necessitates a larger gap to defend against more sources of variation. Manufacturing resources are not as plentiful at non-constraint operations creating variation in the amount of time it takes operators to load lots on available tools (on the constraint, lots should loaded immediately without any delay). Since a WIP buffer is not maintained in front of non-constraints (doing so would unnecessarily increase TPTs), non-constraints are often starved for WIP. This leads to variation in WIP availability. Unlike the constraint, non-constraints are subject to variation in all three of the conditions necessary to utilize a tool. This increase in variation leads to an increased gap to effectively manage the variation. The larger gap allows the non-constraint toolsets to efficiently process material which leads to predictable factory output and reduced TPTs.

Near-Constraint Gap

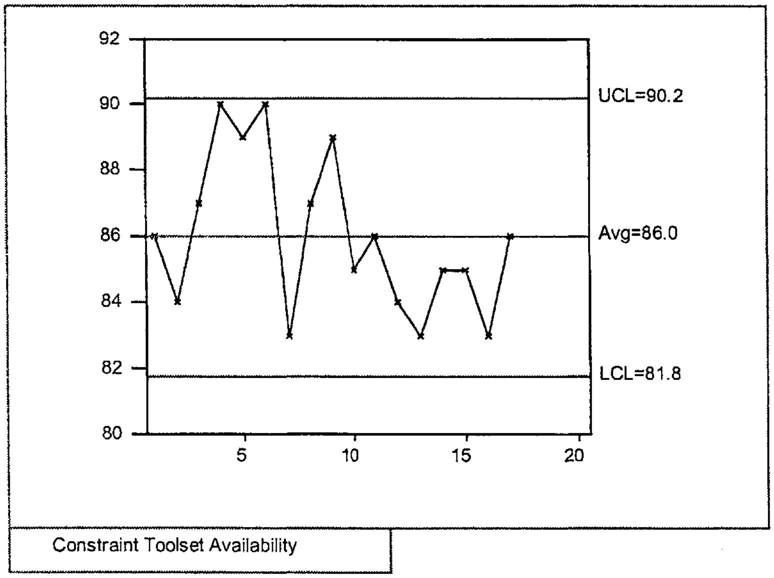
Near-constraints are similar to non-constraints because they are also subject to variability in tool, WIP, and operator availability. The difference is that near-constraints are not subject to the same degree of variation in WIP and operator availability. By definition, near-constraints are closer in capacity to the constraint than non-constraints. The natural dynamics of the factory result in WIP being present at near-constraints a majority of time which leads to lower WIP variation at the near-constraints. Manufacturing also focuses more resources on near-constraints than non-constraints which effectively reduces the variation in operator availability. Therefore, the near-constraint gap is larger than the constraint gap because it is subject to variation in all three tool utilization conditions. However, the amount of variation in WIP and labor is lower at a near-constraints than at a non-constraint which leads to a smaller gap at the near constraint.

2.3.4. Constraint Gap Determination

Variation at the constraint is dominated by variation in tool availability (constraint management dictates that WIP and operators should always be available at the constraint) The strong influence of tool variability on overall constraint variability allows for the gap to be determined from the toolset availability variation. Figure 2.4 shows a control chart for weekly toolset availability. The upper and lower control limits (UCL and LCL) are set at two standard deviations; in other words, the toolset availability should fall within the control limits 95% of the time (assuming the toolset availability is normally distributed).

In this spirit, the gap should be set at the difference between the average toolset availability and the lower control limit. In this example, the gap would be 4.2%. By setting the gap at 4.2%, the factory would be able to produce the output associated with an 81.8% constraint utilization 95% of the time. If control limits were set at three standard deviations (the corresponding LCL is 78.7% which gives a gap of 7.3%), the factory would be confident that it could produce the output associated with a 78.7% MU 99.5% of the time.

Figure 2.4: Weekly Toolset Availability Control Chart



Ideally, the constraint is always staffed with operators and WIP is always present. In reality, the constraint occasionally suffers from operator availability. The gap determined by the control chart method should also include a small buffer to account for unplanned idle time. Labor studies in ATM factories are needed to comprehend the magnitude of this additional constraint buffer.

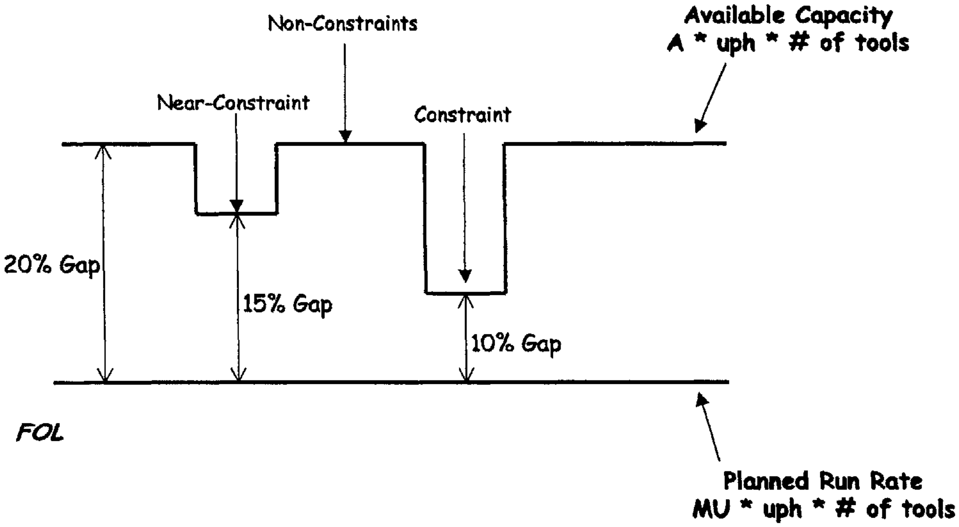
The gap at the constraint serves to give the factory a stated capacity to which it can confidently commit. As constraint cost increases, the gap can be reduced which reduces total capital expenditure (i.e. the control limit sigma is reduced). However, as the gap is reduced, the factory becomes less confident in its ability to consistently achieve the output associated with the gap.

2.3.5. Line Design

Intel employs 10/15/20 in planning capacity for ATM factories. Ideally, the constraint should be the first factory operation (Goldratt, 1992). In most ATM processes, this is not possible. Test is the most expensive operation which mandates that test be the constraint. Testing occurs about ¾ of the way through the entire assembly and test process.

Figure 2.5 shows the design of a typical line at Intel. Run rate (Section 2.1.4) incorporates gap. Remember, MU = Availability – Gap. In this manner, Intel plans capacity to achieve a balanced line in terms of planned run rate. This does not mean the line is balanced since the available capacity of the operations is not balanced. The larger gaps at non-constraint operations lead to higher expected idle time at these operations.

Figure 2.5: ATM Line Design



2.3.6. Advantages and Deficiencies of Gap

The blanket capacity policy known as gap allows Intel to easily implement excess capacity at non-constraint operations without requiring complex modeling for each new process. The implementation of this blanket policy is relatively straightforward, although excess capacity amounts are likely to be sub-optimal.

While the Gap was a great starting place for cutting-edge ToC implementation, it falls short in three main areas.

1. Gap does not scale with Availability and leads to non-intuitive amounts of excess capacity.

2. A blanket Gap policy of 10/15/20 neglects the variation of availability for each toolset.
3. It neglects to specifically consider the cost differences between toolsets.

These deficiencies are addressed by a new concept, equipment Protective Capacity (PC).

2.4. Protective Capacity (PC)

In many respects, PC is simply a computational change of gap. Like gap, PC states the amount of excess capacity at an operation. The gap concept has not been well understood in ATM factories. A major role of PC is to clarify and to further develop the use of excess capacity in ATM. Without this re-branding of gap, it would be very difficult to introduce the paradigm shift associated with optimizing excess capacity levels.

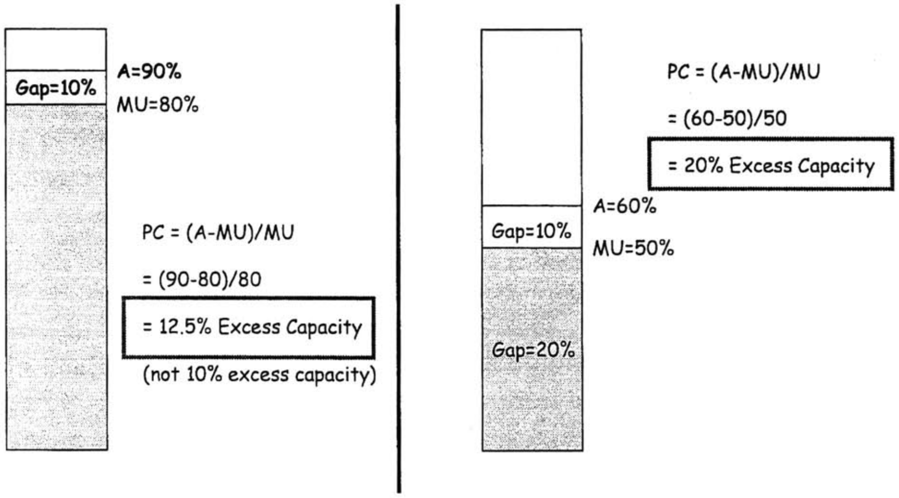
2.4.1. An Accurate Measure of Excess Capacity

Recall that the formula for Gap is A-MU. A blanket gap policy in the face of different availabilities yields different amounts of excess capacity. The solution to this problem is to normalize the Gap formula. Protective Capacity in its simplest form is normalized gap as shown in the following equation. It allows accurate representation of excess capacity, whereas gap does not. This is more easily seen in Figure 2.6.

$$PC = \frac{A-U}{U} \qquad \text{(Eq. 2.4)}$$

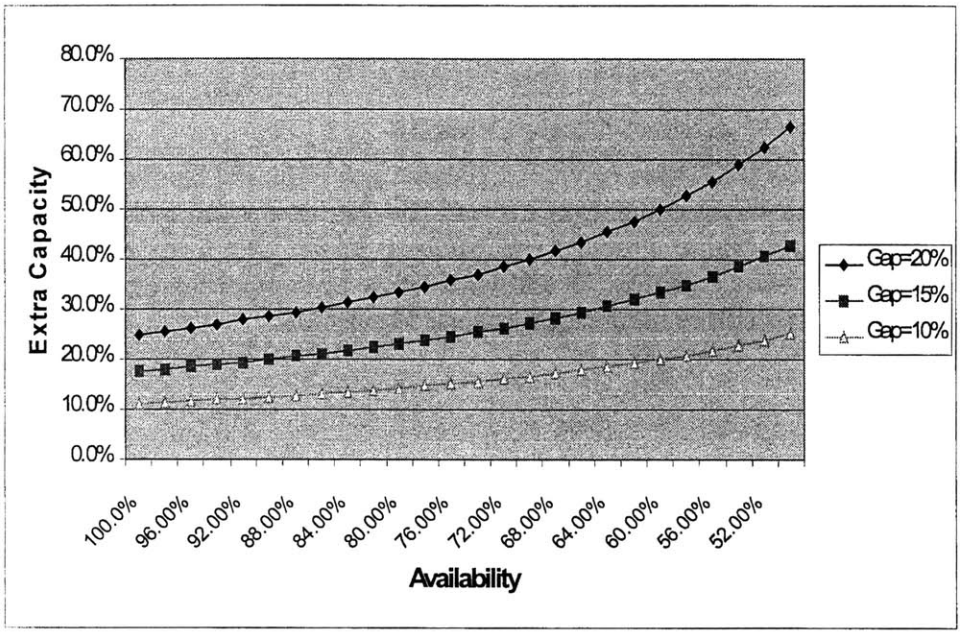
In both examples, a 10% gap is applied to tools with different availabilities. However, since gap is not normalized, the resulting amount of excess capacity is greater than 10%. PC allows for the accurate statement of excess capacity and is a much more intuitive parameter.

Figure 2.6: Gap-PC Comparison



The example shows that as availability decreases (while gap remains constant), the amount of actual excess capacity increases. This relation is shown in Figure 2.7. As availability decreases, the amount of excess capacity increases. This highlights one of the fundamental issues with gap. For a non-constraint tool with a low availability of 70%, the actual amount of excess capacity purchased when using a 20% gap is 40%. Because gap is not normalized, Intel could be buying twice the required capacity of certain tool sets in a worse case scenario.

Figure 2.7: Excess Capacity as a Function of Availability and Gap



2.4.2. Treatment of Individual Toolset Availability Variation

As discussed in Section 2.3.3.2, gap serves to buffer the factory against variation. Variation appears in three main areas: tool availability, technician availability, and WIP availability. For the constraint, tool availability is the main component of variation. It is reasonable to state that variation in tool availability is the main source of variability in the factory. Each toolset has a different amount of variation in availability. For example, a certain tool such as a burn-in oven may be very stable while a highly mechanical operation such as epoxy deposition may be highly unstable.

Since tool variability is the largest component of total variation, a blanket excess capacity policy such as 10/15/20 is likely to be sub-optimal. Instead, a policy that considers each toolset's availability variation would lead to more optimal capacity planning. With the current gap policy, a stable non-constraint with an availability of 70% would have actual excess capacity of 40%. In all likelihood, 40% excess capacity is too much for all but the most unstable operations. In this light, blanket capacity policies sub-optimize factory performance.

One of the premises of PC implementation is that each toolset will have its own PC level (as opposed to a blanket policy). Determination of PC levels based on availability variation could have been performed under the traditional gap policy. However, it is organizationally difficult to attach new meaning to familiar terms. As a result, PC was rolled out in an attempt to help make this paradigm shift.

While the theory behind PC is sound and it makes sense to have PC levels for each toolset, determination of these levels is extremely difficult and is the main driver behind the work in this thesis. Discrete event simulation was utilized in an effort to optimize PC levels for each toolset – this work is described in detail in the following chapters.

CHAPTER 3. Current Factory Capacity Planning Systems and Operational Policies

3.1. Indicators and Factory Goals

Factories operate in a manner consistent with optimizing the indicators by which they are judged. This makes the proper choice of indicators paramount in achieving outstanding factory performance. A close look at the operations of the ATM OLGA factories highlights this point.

Before entering this discussion, the point should be made that only manufacturing-line related metrics will be examined. When asked, any Intel factory manager will respond that their most important metric is safety. In this discussion, it is assumed that employee safety is the most important metric and will not be further discussed.

3.1.1. Behaviors Driven by Indicators

After extensive conversations with ATM managers in several different organizational areas, it was evident that three indicators are paramount in importance. Consequently, managers pursue actions to achieve the best possible numbers for these indicators:

- Output (units)
- MU (Machine Utilization)
- Ku (thousand units)/direct labor head/week

Secondary indicators that are often reported (but receive less focus) are tool availability and TPT.

Output is obviously a very important measurement of factory performance. If a factory fails to meet commitments, Intel's bottom line is impacted. Output is such a concern that factories are rewarded if they are able to exceed commits. Although formal recognition may not exist, factory managers help establish credible reputations by being able to come through at the last minute to meet upside demand requests. The end result of this informal incentive system are factories that may have excess capacity at constraint operations to help ensure the ability to meet upside potentials.

MU is a very interesting indicator. First of all, MU is not measured directly. Instead, it is back calculated using the following equation. MU is calculated for each toolset in the factory.

$$MU = \frac{\text{units produced per week}}{\text{\# of tools in toolset} * \text{uph} * 168 \text{ hours/week}} \qquad \text{(Eq 3.1)}$$

Upon examination of the MU equation, it becomes readily apparent that MU does not reflect factory performance. MU is easily manipulated either by increasing the units processed by a toolset or by reducing the number of tools in the toolset. The drive to increase MU leads to interesting behaviors; the MU indicator will be discussed further in the following sections.

The labor efficiency metric, ku/direct/week, is designed to ensure that headcount does not get out of hand. If this indicator is optimized without regard to the others, the result will be a factory without any operators. This is an extreme example, but serves to illustrate the potential problem with this indicator. As the number of operators is reduced, the amount of factory variability caused by the variation in the time it takes operators to load WIP on tools increases. Given the lack of any type of dynamic labor model, optimization of this metric may lead to sub-optimization of factory performance. In ATM, the largest factory cost is equipment depreciation. While labor costs are significant, enough operators should be employed to ensure maximum return on capital through short delays in WIP loading times. Reduction of ku/direct/week risks large delays in equipment repairs and the time it takes to load waiting WIP onto available tools. The targets for this metric are largely driven by cost pressures and likely lead to sub-optimized factory performance. The findings of this thesis show large gaps in tool availability that may be partially attributed to low staffing levels. Data need to be gathered to determine the applicability and proper targets for this metric (see Section 3.5.2.1 for data collection recommendations).

3.1.1.1. Equipment Hotbagging

MU's power as an indicator has driven equipment hotbagging in the factories. Hotbagging is simply the practice of preventing WIP from flowing through a tool for a week. The tool remains installed on the floor and turned on, but manufacturing does not allow any WIP to flow through the tool. When weekly demand is low (and low MUs would subsequently be calculated), a few tools in each toolset will be hotbagged for the week. The tools are left on and PMs are still performed, but no WIP is processed through the tools. The rational given for hotbagging is to keep the manufacturing organization sharp by demonstrating high tool utilizations.

As shown in Figure 2.3, as MU increases, TPT also increases. Rational reasons do exist for bagging tools. Examples include cost savings through reduced PMs and repairs and the ability to train technicians in other areas. However, PMs are still performed on bagged tools and no financial analysis has been performed to assess the operating cost impact of reducing PMs.

Unfortunately, tool bagging most readily occurs at constraint and near-constraint operations. ATM senior management is most concerned about the utilization of the expensive tools, the hotbagging naturally occurs at the constraints. In the OLGA factories, the most pressure is applied to the testers (constraint) and the epoxy module (near constraint). These areas naturally experience high volumes of WIP in queue and bagging tools in these areas only leads to longer factory TPTs.

While the rational behind bagging tools needs further examination, rational reasons do exist for using MU as an indicator. When the senior managers authorize capital spending, they look primarily at the MU for each tool to determine if they are receiving a fair return on their investment. For example, if a tool has an 80% MU, that is generally acceptable because Intel will be able to receive benefit from its expenditure 80% of the time. If the MU of a tool is only 40%, it is deemed a poor investment because Intel would have to buy twice as many tools compared with a similar tool having an MU of 80%.

Somewhere the notion was born that a high MU in the factory would lead to less capital expenditure. It is easiest to see the flaws in this logic by examining the steps taken to calculate the MU presented to the senior management.

1. The average Availability of a tool is calculated by factoring in expected PM times, failure rates, assist rates, etc.
2. The appropriate gap is subtracted from the calculated average availability.
3. MU = calculated availability – gap

Unfortunately, the planned MU that is used when judging capital expenditure proposals is often confused with the weekly MU that is back-calculated from factory output. While the intentions of measuring factory MU are good, the wrong metric is being measured. If the goal is to improve equipment efficiencies and decrease future capital expenditures, the factories should be measuring equipment availability. As shown in the above MU calculation sequence, an increase in tool availability will actually lead to higher planned MUs and lower capital expenditures.

Ironically, in several weekly reports, TPT is listed in directly below tool MU reports. This is somewhat surprising given the MU-TPT tradeoff. TPT does not seem to be of much concern as long as TPTs are not 'excessively long.' The benefits of fast TPTs are not well understood in the factories. Even though factory demand commonly changes mid-week and results in unwanted products in process (inventory obsolescence), TPT reduction is not a primary focus. In 1999, the theme of Intel's corporate-wide manufacturing conference was 'manufacturing agility.' While the concept is great, it is difficult to understand the meaning of the concept in daily ATM operations. A reasonable definition of agility might be fast TPTs to allow factories to quickly and accurately meet demand.

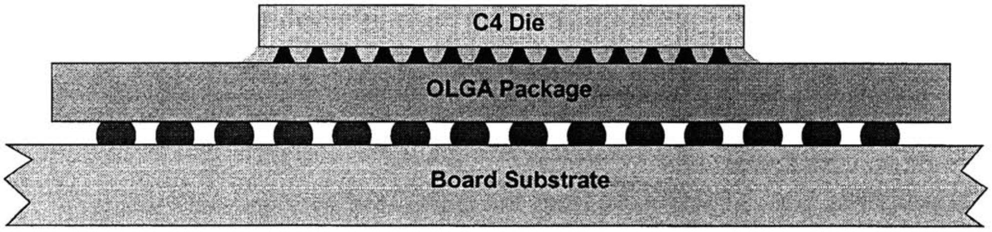
3.2. The OLGA Process

Before exploring factory operations, an introduction to the OLGA process is needed. Intel uses the Organic Land Grid Array (OLGA) as the main assembly process for Pentium© II/III desktop market processors. This assembly process serves to provide a connector between the processor die and a printed circuit board. The OLGA package consists of multiple layers of copper interconnects and insulating material. The finished OLGA package has tin/lead solder bumps to connect to a PC board.

Intel uses the C4 process (Controlled Collapse Chip Connect) as a die-to-package interconnection technology. C4 is used in place of interconnection technologies such as wirebonding or TAB (Tape Automated Bonding). Unlike traditional die to package bonding technologies, C4 bond pads are not limited to the outer die perimeter, but can be placed anywhere on the die surface. This results in higher interconnect density, and higher performance devices.

A cross-section of the OLGA package is shown in Figure 3.1.

Figure 3.1: Cross-section of OLGA Package



Following the OLGA assembly process is a comprehensive device testing process. In this portion of the process, devices are subjected to a burn-in operation to accelerate device failures. After burn-in, all devices are 100% electrically tested before being shipped to the customer. A process flow with brief descriptions follows and the process flow is graphically shown in Figure 3.2.

3.2.1.1. Assembly Process Flow

Bump Reflow: Entire wafers from the fabs are placed in high temperature ovens to reflow the C4 solder bumps on the die. The bumps need to be reflowed to remove any oxides that have grown since bump deposition and to re-shape the bumps in the form of sphere to improve contact later in the process.

Wafer Mount: The entire wafer is adhered to a piece of thick plastic. The adhesive material serves to hold the wafer together during subsequent saw and wash operations.

Wafer Saw/Wash: The die are individually cut-out using a saw that cuts in both the x and y directions. After sawing, the die are washed to remove any debris. At this point, all of the die are held together in the original shape of the wafer by the tape applied in the Wafer Mount operation.

Die Plate: Good die are removed and placed in tape and reel. This medium is used to hold the die for the subsequent operation. Assembly and test lots consist of 1000 units, although this amount decreases through the process as units are yielded out.

APL: The Auto Package Loader (APL) operation is used to transfer the OLGA substrates from the vendor packaging to the trays used for processing during the assembly process. The OLGA substrates contain the circuitry and layers necessary to connect the die with a PC board. Each APL carrier holds 8 substrates. This is a parallel operation and is not part of the main process flow.

SCAM: The Smema Chip Attach Module attaches the die to the substrates (one die per substrate). The operation consists of three main steps: flux application, die placement, and die attach. The tape and reels and APL carriers are inputs to this operation. The APL carrier trays are used to hold the bonded die and substrate when the operation is finished.

Deflux: Affectionately referred to as a large, expensive dishwasher, this operation removes any remaining flux residue from the product. The APL carriers (which hold 8 units) are used to process the units through the operation. A single conveyor belt moves units through the operation.

Epoxy: Epoxy underfill is dispensed between the die and the package (around the connected solder bumps) to relieve stress between the die and package. The epoxy is dispensed by three different dispensers. Units are processed in the APL carrier trays. Again, trays are serially conveyed through the operation.

Epoxy Cure: The units are subjected to high temperatures to solidify and cure the epoxy. Again, units are processed in the APL trays. This is a continuos flow operation with a conveyor belt that moves trays through the oven.

CTL: Carrier Tray Load. Units are transferred from the stainless-steel APL carrier trays to trays made from a different material for back end processing. CTL trays also hold 8 units.

3.2.1.2. Test Process Flow

Burn-In Load (BIL): Units are removed from the CTL trays and placed in the burn-in-boards (BIBs). Each BIB holds 15 units. BIB contain electrical connections to the products.

Burn-In Ovens (BIO): High voltages and elevated temperatures are applied to the devices for a prolonged period of time to accelerate product failures. BIBs are manually loaded into the ovens which each hold 70 BIBs. As products mature, burn-in times decrease. Each oven holds a single lot (up to 1000 units).

Burn-In-Unload (BIU): Units are removed from the BIBs and returned to the CTL carrier trays. The same tools are used for BIL and BIU.

Post Burn-In Check (PBIC): Devices are individually 100% tested. As products mature, test times decrease. Additional tests are performed before BIO and after PBIC early in a product's maturity. All electrical tests use the same equipment. The largest yield hit of a few percent occurs at this operation. Products are grouped into speed (mHz) bins by the tester. If a product fails, it either re-tested 2-3 times (to see if it will eventually pass) or re-worked starting at the burn-in operation.

Semi-Finished Goods Inventory (SFGI): Since the units come from PBIC grouped by speed, speed bins are combined from several lots to create new lot sizes. The maximum allowed lot size (determined by quality and reliability) leaving SFGI is 2100 units.

FQA: Final Quality Assurance test. Intel's conservative approach to quality assurance is the driver behind this seemingly redundant test. It is performed on the same testers as PBIC. 250 units from 1 out of every 10 lots coming from SFGI are tested at FQA.

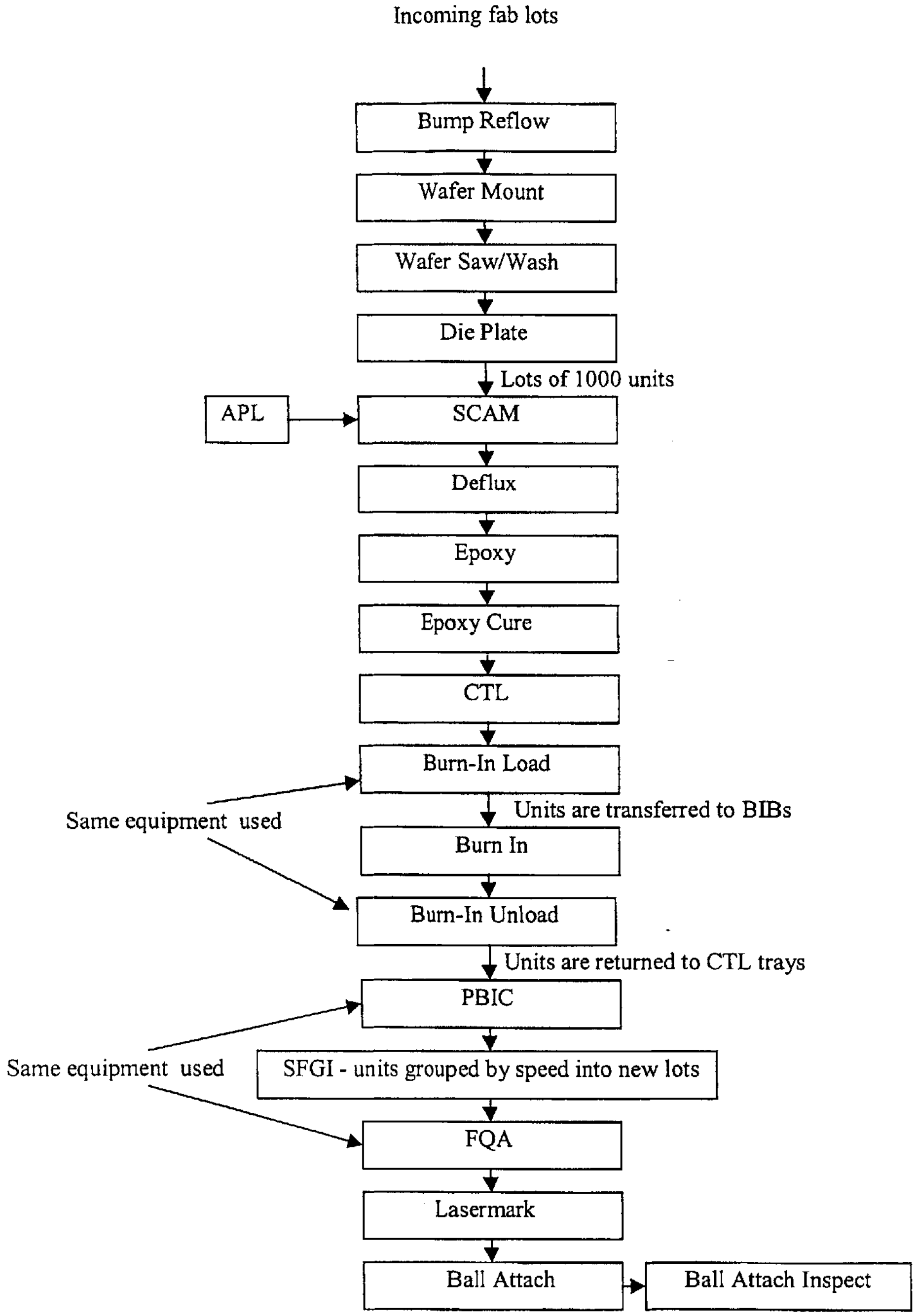
Laser Mark: Each unit is marked for traceability purposes.

Ball Attach: Solder balls are attached to the OLGA package to allow for easy connection to a PC board.

Ball Attach Inspect (BAI): A three-dimensional laser scan is used to inspect to quality of the solder balls.

Pack: After a final visual inspection, the units are packaged from shipment to the downstream customer.

Figure 3.2: OLGA Process Flow



3.3. Factory Operations

ATM OLGA factories are operated by relatively few rigid rules. Manufacturing managers have great flexibility to manage factories as they see fit. Intel's semiconductor fabrication policy of Copy Exactly!, a factory design philosophy where the tool recipes and configurations are identical at all of the different factories, is not as prevalent in operations in the ATM world. As discussed in Section 3.1.1, indicators drive the behaviors of the factories and thus the operational policies. Since MU is the key indicator, policies are designed to maximize MU at the expense of other parameters such as TPT. Following are descriptions of the general policies used to run ATM factories.

Table 4.3 shows the planned availability and uph for the OLGA tools. Cost data and typical tool counts are shown in Table 5.6 and Table 4.1, respectively.

3.3.1. CONWIP Policy

As opposed to the pure drum-buffer-rope (Goldratt, 1992) system preached by The Goal, OLGA factories employ a Constant WIP System, or CONWIP system (Hopp and Spearman, 1996). The logic behind a CONWIP system is that in most factory settings, the 'rope' is unreasonably long. If factory starts are based on a constraint buffer size 16 operations into a line, ample opportunity exists for WIP buildup before the constraint. The supposition is that WIP levels will be erratic at the constraint (and throughout the entire line up to the constraint) and constraint starvation is likely occurrence.

Hopp and Spearman advocate simply controlling the entire amount of WIP between the start of the line and the constraint. By only making starts when the CONWIP level falls below a specified level, the total amount of WIP is the factory is controlled and TPTs stay relatively constant. As with the drum-buffer-rope approach, optimization of the CONWIP level is important to factory performance. If the CONWIP limit is too low, the constraint will sometimes starve (assuming the factory is fully loaded). Alternatively, if the limit is too high, TPTs become unnecessarily long. When used correctly, a CONWIP limit is optimized to ensure a fed constraint while minimizing factory TPT.

OLGA factories face the probability of a shifting constraint between test and epoxy. Intel has cleverly modified the basic CONWIP principle to accommodate these shifting constraints. OLGA factories are divided into three blocks which give rise to multiple CONWIP limits within the factory:

- Block 1: Reflow to Epoxy
- Block 2: Epoxy Cure to Test (PBIC)
- Block 3: Lasermark to EOL

OLGA factories follow a 2-2-1 policy. Ideally, 2 days of WIP are in Block 1, 2 days in Block 2, and 1 day in Block 3.

One day of WIP is defined as the weekly production output goal of the factory divided by 7. ATM factories face the challenge of constraints shifting over time due to process maturity issues. As processes mature, test times significantly decrease. This dramatically impacts tester capacity and often results in the constraint shifting to the epoxy module. To address this issue, Intel has attempted to institute a dynamic CONWIP policy to account for the possibility of shifting constraints. The logic of the policy is as follows:

If Block 1 is full, do not make any new starts

If Block 1 is not full, look at (Block 1 + Block 2)

If (Block 1 + Block 2) > 4 days of inventory, do not make any new starts

If (Block 1 + Block 2) < 4 days of inventory, make new starts

Starts are made in the factory approximately every 6 hours (assuming WIP limits are not exceeded). In theory, this policy allows for the constraint to shift from Test (end of block 2) to Epoxy (end of block 1). In reality, this policy was only loosely followed. On several occasions, 5+ days of inventory were observed in the first two blocks.

It is also worth noting that Block 3 is essentially meaningless. Block three is after both potential constraints. Following the rules of ToC, time lost at these non-constraints is an illusion and can be made up for. Accordingly, the manufacturing organizations do not pay attention to WIP levels in that portion of the line.

3.3.2. Conversions and Setups

It is widely known that conversions and setups negatively impact a tool's availability. However, because the availability indicator receives less focus than other indicators, the impact of conversions and setups are readily apparent. It is difficult to measure availability given the lack of automated data systems in the factory, but this inability can be attributed to the lack of emphasis on this indicator.

Setups are generally negligible at most operations. Most operations are continuous flow, so a new lot can be loaded on a tool right behind a preceding lot. Operations such as epoxy, SCAM, deflux, and cure all have conveyor belts in the tool making it possible to eliminate setup times completely (assuming an operator is present to load the WIP). Most other operations can batch multiple lots making setups negligible. Only two tools have significant setup times: Test and Laser Mark. The laser mark tool runs at an incredibly high uph, but requires at least 30-45 minute setup per lot. Surprisingly, the module engineers and the IEs did not know the exact setup time. This implies sub-optimization of capacity due to a lack of focus on significant setup times.

Testers require about 15-18 minutes between lots to store the acquired data and load a fresh program for the oncoming lot. Again, none of the engineers or IEs actively measured this number for the purposes of continuous improvement. Through conversations it was discovered that the impact of setup time on the capacity of constraint tools was not understood. Additionally, the testers require at least 45 minutes to convert between products and/or tests. The IX tester performs both the PBIC (post burn-in check) test and the FQA (final quality assurance) test on the same product. It takes at least 45 minutes to prepare the tester for a different test. Surprisingly, no one knew the exact time it took to covert a tester. This was very disturbing given that testers are the constraints – again, the relation between tester capacity and conversions was not comprehended.

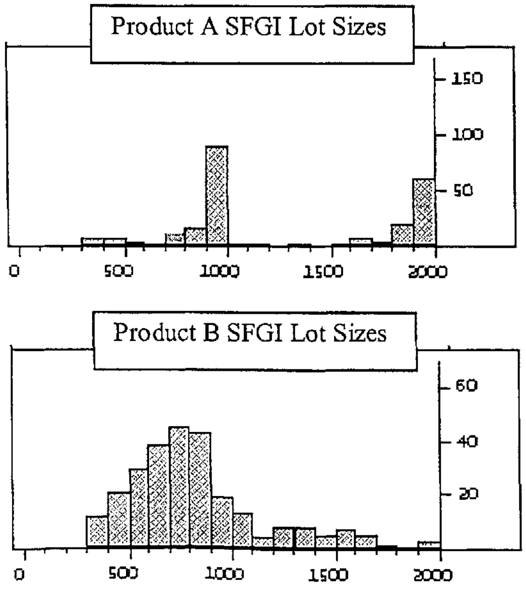
This ignorance of capacity impacts was evident in the lack of defined WIP policies in the test area. Each factory had different rules of thumb, although it appeared that shift supervisors simply converted testers as they saw fit. Unfortunately, data are not readily accessible that show the number of conversions for each tester.

The Costa Rica factory would let the FQA queue size build to 5000 units before converting a tester to FQA. The origin of this number was unclear. In summary, setups and conversions are major consumers of tester capacity and little effort has been placed on optimizing policies in this area. The only focus seems to be MU.

3.3.3. Lot Sizing Issues

An interesting opportunity exists for lot sizing at the SFGI operation. After units are tested, they are placed in new lots according to their speed bins. For example, all of the 600 MHz processors are grouped together, the 500 MHz processors grouped together, etc. The percentage of die that fall into a given speed category is referred to as the bin split. The official specification for the operation states that lots leaving SFGI may be as large as 2100 units. Additionally, the FQA specification only requires that 250 units per lot be tested (regardless of lot size). In other words, the larger the lot size coming out of FQA, the fewer the lots that need to be tested using the FQA test. Since tester capacity is at a premium, it would make sense that SFGI lot sizes should be maximized. The only potential downside to making all lots 2100 units is that some bin splits are rare (i.e. the bin split for the fastest speed processor may be approximately 10%), meaning that waiting for a full lot of 2100 would impact the TPT of those units. Examination of the SFGI lot size data reveal that this issue is not comprehended. Figure 3.3 shows a histogram of the SFGI lot sizing data for each product in Costa Rica.

Figure 3.3: SFGI Lot Sizes for Costa Rica WW26



The histograms show the SFGI lot size is usually below 1000 units. This impacts tester capacity and laser mark capacity due to high conversion and setup times, respectively. By simply ensuring that only large lots leave SFGI, the laser mark setup times could be cut in half along with the time the testers spend converting to FQA.

Additionally, the lot size of 1000 units is maintained from Die Plate to the end of line. Testers are the most expensive tools in the factory and have relatively long setup times of 15-20 minutes per lot. Two lots could be merged at test which would effectively cut the number of setups in half, thus increasing tester availability. There do not appear to be insurmountable quality and reliability issues from merging lots, but there is a lot of momentum behind the traditional way of doing things. Simulation experiments showed that a tester with an average relatively long test time and a setup time of 17 minutes spends 11% of its time in setup per week. It lots were merged, the number of setups could be cut in half and approximately 5% tester availability could be gained.

3.4. Capacity Planning

Capacity planning in ATM has been a major focus area over the past two years. Copy Exactly!, while heavily used in the fab world, is a relatively new concept to ATM. The fact that over 5,000 different capacity models existed for the ATM factory network demonstrates this point. Even factories running the same process used different models. The ATM IE (Industrial Engineering) group has spearheaded the move to a standardized capacity planning system known as CAPS (Capacity Analysis Parameter System).

The goal of CAPS is standardize the capacity planning process. On the most basic level, it provides a central database for tool parameter storage. All factories will agree upon and use the number located in the central database as opposed to each factory using their own numbers. Furthermore, a standardized capacity model is being developed that all factories will use.

Capacity planning is done using static models (both the new standardized system and previous models use this approach). By definition, static models do not have the ability to deal with the dynamics of a manufacturing system. Instead of attempting to capture variation in a parameter, they simply use an average number. For example, while a tool's actual availability may be distributed between 70% and 90% with an average of 85%, the static models simply treat the tool as having an availability of 85%. It is hard to imagine a parameter within the models that does not have an associated distribution. None the less, static models only consider the average of each distribution. Examples of distributed parameters where only averages are considered are MTBF, MTTR, MTBA, MTTA, PM time, setup time, and the time it takes to change consumables. The cumulative affect of this variation is ignored by only considering the averages and leads to inaccurate capacity models. Furthermore, static models do not have the ability to properly account for conversions. Instead, the number of average conversions is used – the origin of this number is an educated guess at best.

The CAPS project has established the first version of a common database for tool parameter data. While it currently feeds static models, the hope is that it can one day be modified to store parameter distribution data and feed dynamic simulations.

3.4.1. Methodologies

As one can image, it is terribly difficult to model a highly dynamic system using a static model. The results of these attempts are incredibly complex Excel spreadsheets that demand data which will never be measured. Testers are the most operationally-complex tools in the factory due to long setup and conversion times, varying test times and sample rates, and highly variable re-test rates. Tester models are incredibly complex due to their attempt to capture factory dynamics. However, crucial parameters such as conversion frequency are simply estimated. This makes planning capacity at the expensive constraint tool and arduous and inaccurate task.

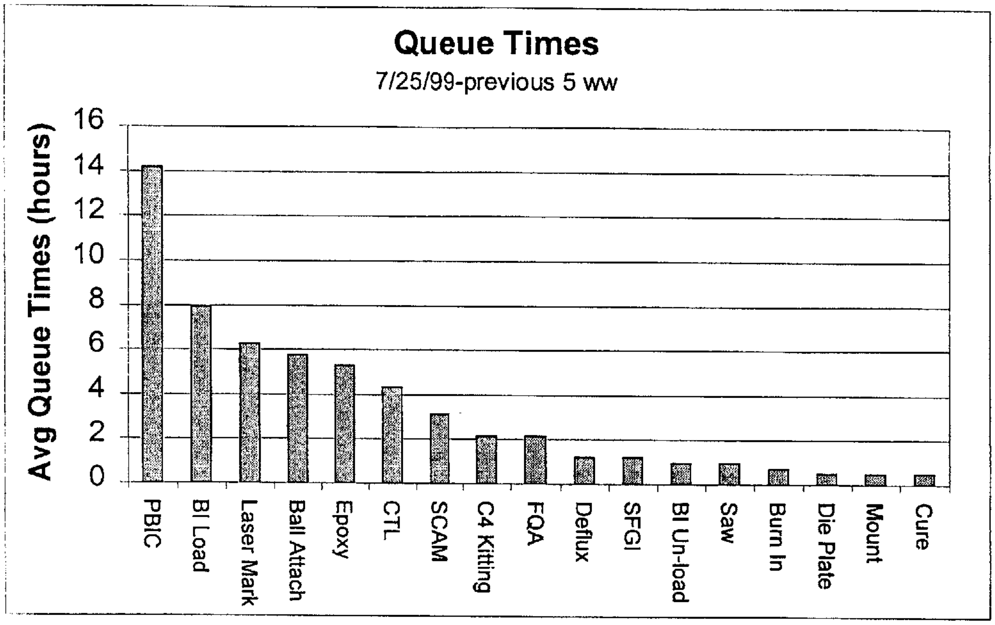
Capacity IEs plug factory loading data into these massive spreadsheets and observe the final output of number of tools required. If the factory is being planned, they will round the tool requirement up to a whole number and order that amount of tools. If the factory is running, the model output will determine how many excess tools are on the floor for bagging purposes.

The internal calculations primarily lead to a tool's average availability. Once the average availability is calculated, the appropriate gap (10/15/20) is subtracted to arrive at the expected utilization (MU). This MU is multiplied by the uph to arrive at a the run rate (Section 2.1.4).

These static capacity models supercede any real time analysis in the factories. For example, constraint are determined by simply looking at the capacity model and seeing which toolset has the lowest predicted capacity. An active approach of monitoring WIP levels in the factory to identify constraints is not used. Given the static model's inability to capture any type of variation, it is a good idea to actively identify constraints by looking for large WIP buildups in the factories. Active WIP level monitoring helps determine the actual constraints of the factory.

Figure 3.4 shows the average queue times for all the factory operations over a five week period. This type of graph was new to the factory organizations, but provides valuable insight into actual factory performance. The IE models predicted that PBIC test would be the factory constraint. The longest queue is time observed at this location and validates the capacity model. However, the static models then predicted that epoxy was the near-constraint of the factory. In fact, several operations had more WIP than epoxy indicating that several other toolsets constrained the factory more than epoxy. This simply underscores the inherent inaccuracy of the static models and the need to collect real-time factory data.

Figure 3.4: Queue Times by Operation for the Philippines Factory



3.4.2. Systematic View of Capacity Planning

The newfound focus on ATM capacity planning has lead to a much needed systematic view of capacity planning. Coordination and standardization on activities such as capital purchasing, ATM capacity buffers and standardized capacity models is rapidly improving the performance and effectiveness of the ATM IE organizations. The creation of systematic processes in all of these areas is necessary to effectively manage the business. However, these processes will be sub-optimal until ATM factories understand how their capacity is actually performing. In other words, capacity planning systems will need to be jointly developed with systems that accurately measure tool performance in order to maximize their usefulness. Without a basic understanding of how capacity actually performs, capacity planning systems will continue to be sub-optimal. Improved data collection systems are the next necessary step in the continuous development and improvement of ATM capacity planning.

3.4.3. Factory Output Estimation

The final result of the static capacity planning models is the prediction of factory output capability. Since static models are unable to capture variation, factory output is shown as a single number. In reality, factory output approximates a normal distribution. After all, if variation is present in every operation in the factory, why would the output contain no variability? This type of thinking needs to be changed in order to maximize the effectiveness of the factories. The current gap policy is designed to cushion the variability in the factory; the resulting output estimations from this policy are extremely conservative. With such large gaps (a.k.a. idle times) planned at all operations, the factory is extremely likely to be able to meet the stated capacity of the factory.

It appears that the stated factory capacity is actually about the 95% confidence of the normal distribution. Factories should almost be able to meet these commitments. It is not understood that factories are simply taking advantage of the factory's ability to produce at levels higher than the 95% confidence level. The mindset that a factory's output is a fixed number leads to the loss of the upside potential of the factory. It would be more useful to state output confidence intervals for each factory.

3.5. Problems Common to Both Static and Dynamic Models

Whether in reference to a static or dynamic model, the old saying 'garbage in, garbage out' applies equally to both models. In short, manufacturing performance cannot be improved unless current performance can be accurately measured. Both models are dependent on tool performance data; primarily tool availability data and run rate data. This project highlighted the degree of data inaccuracy from these systems (Section 4.3.4). In essence, the development of improved capacity planning models must be undertaken in parallel with the development of improved data collection systems.

3.5.1. ATM Data Collection Systems

ATM factories only measure two manufacturing performance parameters accurately: output and TPT data. These parameters are stored in the WIP tracking system call WorkStream. Each time a lot is ready to start an operation, it is 'proc'd' in by an operator. The time and operation are recorded. Similarly, when a lot finishes an operation, it is 'proc'd' out by an operator. This leads to reasonably accurate measurement of total TPT, processing time for each operation by lot, and queue time at each operation by lot. Also, the number of units entering and leaving the factory is accurately counted. A web-based data filter/extraction interface called EATS has been built and allows for easy and customized data extraction from WorkStream. This project required extensive extraction of queue times for each operation for very specific time periods – the interface proved robust for this type of extraction. It should be mentioned that yield and re-work rates are also measured accurately, but these parameters are quite healthy and do not severely impact factory performance. These values are also stored in WorkStream.

Unfortunately, all tool related performance parameters rely on manual logging systems for measurement. The predominant system is known as CEPT. CEPT relies on the operator to log all tool activity. If a tool is down, it is up to the operator to notice that the tool is down, log it down in CEPT, and give the reason why it is down (PM, repair, etc.). As one can imagine, it is very difficult to obtain accurate data from such a system. First of all, an operator runs several tool simultaneously and may not notice that a tool is down for 10-15 minutes. Operators are extremely busy and may not log short failures and/or assists. Finally, factory management does not emphasize the need for accurate CEPT data, so operators do not have an incentive to make it accurate. If they did, they might also overstate availabilities for fear of looking bad if tools are down for long periods of time.

All of these factors lead to distrust in CEPT data accruacy. While interviewing people from several functions in the factory (engineering, manufacturing, and industrial engineering), not one person stated that they trusted CEPT data. In fact, no one thought it was accurate within 10%. This leads to the obvious question of why CEPT is even used, but this issue was not discussed during the project.

The IE's have realized the need for better data at the constraint operations (test and epoxy) and have created manual logging systems for the operators (in addition to CEPT). These are either paper sheets or spreadsheets on the computer; these systems are called 'MU studies.' IE's train the operators to record tool events (downtime and cause, idle time, utilization, etc.) on the sheets. More emphasis is placed on the need for accuracy and it is likely that these methods provide better results than CEPT. However, the issue of busy operators not having time to log all events and the fear of looking bad if too much downtime is observed still exists. One other discovery that emphasized the weakness of this system: in one of the factories, the IE's were concerned with capturing the idle time of the epoxy tools. The IE's called this 'Gap' on their tracking logs. It was later discovered that operators were logging 'Gap' events whenever there was an empty slot (not occupied by a tray of product) in the epoxy oven. While this is a logical interpretation of gap, it demonstrates that ineffective training and emphasis of such tools. The results suffer a tremendous loss of credibility from such situations.

3.5.2. Proposed data collection system

Clearly, the opportunity exists for better data collection systems. Until more accurate factory data are collected, the improvement of capacity planning systems and operational procedures will be limited. This is a major effort that would require coordination across several organizations and layers of management. An attempt is made to give a brief outline for an improved data collection system.

3.5.2.1. What needs to be measured

Before even thinking about how to collect better data, the type of necessary data must be identified. Millions of unmanageable data points could be collected from a running factory and easily overwhelm all potential users of the data. It is therefore very important to create system that collects a manageable and useful set of data.

In order to improve the factory's performance and optimize capacity planning, four tool states need to be measured:

- The tool is utilized – it is actively processing WIP
- The tool is idle because of lack of WIP – the tool is available and ready to run WIP, but none is available

- The tool is idle because of lack of an operator – WIP is present in the area and the tool is available, but no operator is present to run the WIP
- The tool is unavailable – it is down due to PM, repair, assist, setup, or conversion.

An argument can be made that the factory can be optimized using these four basic parameters. WIP loadings and CONWIP limits can be judged by measuring utilization and idle time due to lack of WIP. Staffing levels and efficiencies can be measured by looking at how often a tool is available but not utilized because an operator is unavailable. Overall availability can be easily computed by taking 100% less the unavailable percentage.

Tools with excessive downtimes can be easily identified and appropriate attention given to them. An enormous benefit would be the ability to understand staffing levels and operator efficiencies. At this time, ATM does not have a solid understanding of how much of a tool's availability is wasted because an operator is not present to load WIP on the tool.

3.5.2.2. How to measure it

It was often suggested to simply have an IE monitor a toolset 24 hours a day for a certain period of time in order to measure tool performance. On the surface, this sounds like a good idea and may provide a badly needed picture of where true tool performance lies. In the short term, this may be reasonable on a constraint tool. However, this solution is infeasible in the long term. First of all, it will be difficult to find (and justify the cost for) a group of people to do nothing except sit in the factory and record tool performance. More importantly, operator performance will likely be impacted by the presence of an observer. The idea of IE monitoring introduces a disturbance to the system that cannot be properly accounted for, especially when the goal is to record actual tool and operator performance.

Manual logging systems that rely on operators have already proven their inadequacy. Even if emphasis was placed on data accuracy by management, it is unreasonable to expect that an operator can accurately log events for several tools in an area. The IE-initiated MU studies have demonstrated this.

Therefore, automated data collection systems need to be employed to collect these data. In theory, the data should be easy to collect. A program would monitor a tool's state – a tool is either utilized, available and idle, or down to production. If the tool is available, the program would look at WorkStream and see if WIP is available in the area and then log the appropriate type of idle state (without WIP or without operator).

The first parameter that needs to be established is the time scale for data reporting. Factories currently are judged on weekly performance and that seems like a reasonable time scale to use. At some point, the time scale becomes so short that the variation becomes ridiculously large. Conversely, a time period that is too long does not allow for proper reaction to issues or the effectiveness of new policies. Lastly, since the factories are already used to weekly metrics, it makes sense to try to minimize the necessary paradigm shift and leave a few things alone. However, if TPTs are reduced to less than a week, it would make sense to shorten the time scale to something less than 1 week.

Secondly, the amount of data to be collected needs to be determined. For example, when a tool is down, should the system try to capture why the tool is down (repair, PM, etc.)? Initially it seems reasonable to keep the system as simple (and robust) as possible. Downtime pareto charts would be quite helpful to engineering and manufacturing, but simply capturing accurate downtimes would be extremely helpful. A seemingly infinite number of tool parameters could potentially be captured by such a system, but successful implementation would require maintaining an achievable scope that would provide immediate returns. Minimizing the scope to the parameters listed above would allow for such goals to be met.

After observing the factories, it is likely that simply providing accurate downtimes, idle times (and their causes), and utilizations would lead to several months worth of factory improvement projects.

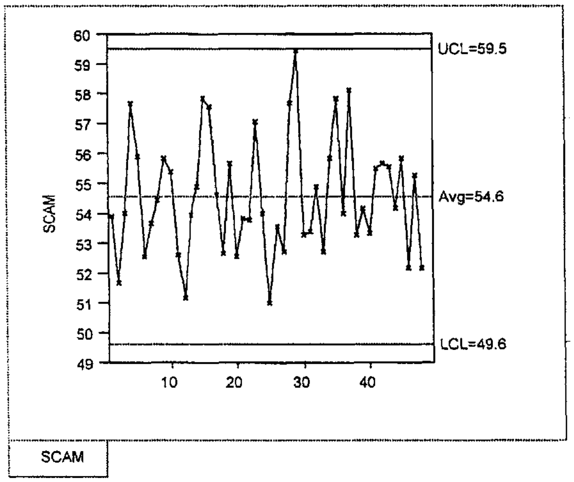
The cost and time required to implement such a system are large and need to be considered. With such large capital cost differences between toolsets, the highest returns on production optimization obviously come from improving the most expensive tools. In this spirit, data collection systems should be installed on the 3-4 most expensive tools in the factory – test, epoxy, SCAM, and ball attach, respectively (Table 5.6). This would allow the system to be piloted on a relatively small scale (as opposed to factory-wide implementation) and to prove the system by demonstrating improvements on factory constraints.

The exact details of how such a system would work are yet to be determined. Intel has skills in factory data collection systems from the fabs and this should be leveraged as much as possible. Such a project would entail the collaboration of toolset engineers, tool vendors, fab automation engineers, and representatives from manufacturing, engineering, and IE. ATM factories will continue 'driving blind' until such a system is installed.

Availability data would allow the engineers to monitor tool performance metrics that they can actually control (currently, some engineering groups are judged by the MU of their toolsets – they do not have any control over MU since it is solely dependent on WIP levels). Engineering could employ availability control charts such as the example chart shown for the SCAM module in Figure 3.5. The goal of the engineering organization should be to increase the average availability of their toolset while concurrently reducing the variation in availability. Weekly control charts could be plotted for each tool along with a composite chart for the entire module and would provide engineering with a clear picture of the health of their tools. The definition of availability in this paper includes setup and conversion times, so engineering would have to work to reduce both of these. This would result in a more agile factory (currently, there are not any groups working to reduce setup and conversion durations).

The data provided on the amount of idle time caused by the lack of an operator would be invaluable in allowing manufacturing to efficiently staff the factory. They could easily monitor this metric for each area and add/remove operators until the amount of idle time caused by not having an operator present is essentially 0. A pareto chart of ATM factory costs reveals that equipment depreciation is the largest cost in the factory with labor costs trailing as a distant second. Therefore, the goal should be to have the factory adequately staffed to achieve the highest return on invested capital. Current incentives are reduce labor content by a set percentage each year. A data collection system that provides actual data about operator performance in each area would allow staffing levels to be optimized instead of simply minimized.

Figure 3.5: Tool Availability Control Chart



ATM factories need to have a thorough understanding of what actually happens on the factory floor before significant improvements in factory performance are possible. A data collection system that measures the parameters detailed in this section would provide dramatic insights into actual factory performance and most likely lead to several easy solutions that would improve factory performance and agility. The difficulties of installing such a system are primarily creating the paradigm shift from the old, inadequate system and indicators to a newer, simpler method of operating factories. While this sounds simple and efficient in theory, the organizational and leadership skills needed to create such a change are the largest challenge associated with such a project. Skill is required to educate and convince managers that they need a new system without insulting the way factories have been operated for the past 15 years. The technical issues associated with how to collect the data are relatively easy compared to the organizational difficulties.

An ongoing automated data collection project (AEPT) is active in ATM. The project has recently become a higher priority and is slated for installation on most toolsets in future processes. Coordination with the AEPT project to incorporate the data collection ideas proposed in this thesis would help to achieve a meaningful system that helps improve factory performance.

CHAPTER 4. PROPOSED SYSTEM: DISCRETE EVENT SIMULATION

4.1. Overview of DES

Factories are inherently filled with variation. Variation is present in the three conditions required to process WIP (tool must be available, WIP must be present, and an operator must be available to run the WIP on the available tool). As shown in previous chapters, static models cannot capture the dynamics of a manufacturing system. Mathematical approaches using detailed statistical methods (Gershiwn, 1994) can adequately describe simple manufacturing environments involving only a few tools. However, as the number of tools and products increase, the mathematical approach is quickly bogged down by the complex calculations and the method becomes unfeasible.

The elimination of static and mathematical models as viable approaches for understanding and predicting factory dynamics logically leads to discrete event simulation (DES) as the tool of choice. DES utilizes underlying statistical distributions to predict the behavior of a system over a given period of time. DES models the factory over time because it is dynamic. Tool availabilities are generated from statistical distributions, WIP flows between operations and are processed dynamically, and the TPT and output of the factory are predicted.

It is important to realize the inputs and outputs of DES. For the purposes of this project, labor was not explicitly modeled due to the lack of any data on labor performance. Tool variation was suspected of being the major component of factory variation and therefore composed the entire initial model. Inputs to DES are as follows:

DES Inputs

- Tool availability distributions: The distributions are calculated from the following underlying distributions (all of these input parameters were assumed to vary): PM duration and frequency, failure occurrences, repair duration, assist occurrences, assist duration, and consumable change duration and frequency.
- Tool performance data: Processing time by product for each operation, tool capacity, number of tools per operation, setup time per run, and conversion time between products.
- Factory loadings: Weekly loading by product. Yield by operation is also needed.
- WIP Policies: For this project, Block Limit rules were coded in along with conversion/dedication strategies. Lot sizes are also included.

DES Outputs

- Tool utilizations
- Total output by product
- Total TPT by lot
- Queue times for each operation

Simply the creation of a DES does not solve all of a factory's problems. Managers must understand the inputs of any outputs of a model (and thus the factory). Optimization of a factory using DES is a tedious and time-consuming process. For the OLGA process, 15 weeks of simulation took approximately 20 minutes to run and an additional 40-60 minutes to analyze the output.

DES allows for factory optimization based on capital cost requirements, required cycle time, tolerable variation in output, and several other things that are discussed in detail later in this chapter. In the past, an iterative approach has been used to optimize factories. If capacity planning at minimum capital cost to achieve a given minimum output were the goal of the simulation, a toolset would be chosen and the simulation would be run. Tools would be iteratively added/removed until an 'optimal' toolset was chosen. It is obvious that one must be careful of sub-optimal minimums in such a process. One of the goals of this project was to find more precise methods of optimizing capacity requirements using DES.

DES is powerful because it can model factory variation and allow users to see how variation affects their factory. Variation can be classified into two broad categories: independent and dependent. Independent variation occurs regardless of other factory dynamics. The best example of this is tool availability. Tool A may fail and need a repair; this failure does not impact the availability of Tool B. In other words, tool availabilities are independent of each other. Because the simulations are time based, DES can easily accommodate these random factory events.

A good example of dependent variation is WIP flow through the factory. If tool A is upstream of Tool B and Tool A fails, Tool B may become starved for WIP. In this instance, the variability of Tool A impacts the WIP availability of Tool B. DES shows the impact of related factory variation and allows for the prediction of parameters such as total cycle time. Further uses of DES are explored in the remainder of this chapter.

While DES is a powerful tool, it does require much more effort than traditional static models. For this project, a program called AutoSched produced by AutoSimulations was used. Thousands of lines of underlying code are required to run the simulation. Excel spreadsheets were used as input and output modes for the simulation program.

4.2. Literature Survey of Semiconductor Factory Simulations

As previously mentioned, Intel appears to be on the cutting edge of optimizing excess capacity at non-constraint operations to manage TPT and output variability. Few references exist for such a topic, especially in the context of semiconductor manufacturing. A common theme running throughout the simulation literature is the lack of accurate factory performance data.

Kotcher and Chance (1999) recommend that any given tool in semiconductor manufacturing should not be loaded to more than 85% of its capacity. In the terms of this thesis, the authors recommend that utilization/availability should always be less than 85%. They observe that any tool loaded too close to 100% results in a significant cycle time penalty. Kotcher is simply making an observation from the fundamental utilization-TPT relationship shown in Figure 2.3. His paper does demonstrate the lack of documented research into excess capacity optimization.

Grewal et. al. (1998) touched on the issue of using excess capacity to manage cycle times, although it was in reference to fab maufacturing. They referred to the fact that most static models in semiconductor manufacturing use excess capacities of 10-30% across all operations to meet cycle time requirements. They then state that this is a 'brute force' method that leads high costs. They suggest a more cost effective method is to plan a factory with a small amount of buffer capacity at all operations, then to purchase excess capacity at operations with high cycle times in order to manage overall cycle times. While this strategy may work for fabrication facilities where costs are relatively equal across toolsets, it clearly does not apply for A/T factories which have such large cost differences between toolsets. Grewal does adhere to the principle that the maximum allowable U/A is 85%. No data were given showing the relationship between excess capacity and cycle times.

Domaschke et. al. (1998) address the issue of cycle time reduction in assembly and test factories, although the idea using excess capacity to manage TPTs is not discussed. They did suggest a batching policy at burn-in that lead to MUs of 96% and higher. They commented that MUs higher than 96% caused cycle time to degrade, although no MU vs queue size data are shown. They are demonstrating the relationship shown in Figure 2.3. The group used a commercially available software package to model their A/T factory. They were able to match actual factory performance within 10%. The model also attempted to concurrently model factory operators. Their main use of the simulation was to examine WIP policy changes such as burn-in batching techniques and tester dedication. Several potential scenarios were explored using the simulation, although details were not given about the success (or lack) of implementing the findings from the model. In general, the paper was sparse on model details and validation, but did highlight several potential uses for an A/T factory model.

Chance et. al. (1996) wrote about the basic role of simulation in manufacturing. They quickly identify that while spreadsheets are simple to use, they are deterministic and cannot predict cycle times or WIP flows. They generally describe the fact that simulation can model almost any detail of a manufacturing process, but that identification of a feasible scope is the most important part of creating a successful simulation. The lack of data was sighted as a common cause of failure for simulations. Unnecessary complexity of the models was given as a common cause for the data issues.

In a separate paper, Grewal et. al. (1998) addressed the issues of validating cycle times in their simulation. Initially, the cycle times predicted by the simulation were significantly shorter than actual factory cycle times. The lack of labor efficiency (staffing levels and cross-training) and equipment dedication rules in the model were identified as major causes of the under-predicted cycle times. Data were not readily available for these topics and adjustments were made to the model so it would match actual factory cycle times.

Kurz (1995) explored line design in the face of differing factory settings. For factories with expensive bottleneck operations and relatively inexpensive non-bottlenecks, he advocates purchasing enough buffer capacity at non-constraint operations to ensure that the constraint is always fed with WIP. The methodology proposed by Kurz is applicable to the ATM factory setting. The next step is to optimize the level of excess capacity at non-constraint operations to ensure high constraint utilization and to meet factory throughput time goals.

4.3. OLGA Factory Simulation

The OLGA factory simulation was originally intended to optimize PC levels in the OLGA factories. After the model was build and validated, it became obvious that PC level optimization was only one of the many potential uses of the model. The underlying learning from the project was the enormous potential of simulation models to address a wide variety of factory issues without risking actual factory production.

The Costa Rica (CR) OLGA factory was chosen for modeling purposes. There were three primary reasons for this choice:

1. Product Mix
2. Data Availability
3. Factory Accessibility

During the summer of 1999, the CR factory ran two products in the Pentium® family of processors: Product A and Product B. Both of these products were fairly mature at the time which meant the yields were stabilized and extra testing was at a minimum. Furthermore, both products were manufactured in significant volumes.

Although data quality was generally poor, Costa Rica did appear to have more accurate MU studies for the test and epoxy modules. (the issue of data quality is discussed in Section 3.5.1).

The geographical location of the Costa Rica factory made communication much easier than with the Asian factories. Costa Rica is only one time zone ahead of Arizona making daily phone calls a reality. The work days of the Asian factories only overlap with Arizona for 1-2 hours making communication more challenging. The ability to pick up the phone and get a quick answer to a simple questions was invaluable. Additionally, the Costa Rica management was very helpful in providing information and contacts.

Work weeks (WW) 28-34 (excluding WW32) were chosen for the initial study. The goal was to validate the model using WW28-30 and then see how accurate it was at predicting WW31, 33-34 (WW32 was not used due to unusual power failures that disrupted production that week). Accurate prediction of actual performance in these three weeks would add immense credibility to the model. During this entire time period, reliable data existed for the number of operation tools per area, factory loadings by WW, and product TPT for each lot. Furthermore, two other factors made these weeks ideal choices:

1. Only two products were being run – Products A and B. These products were very similar and most tools did not require conversion to switch products. The modeling of multiple products would add credibility to the model, but a variety of product mixes would have added unnecessary complexity to the model at this stage of development. During this time period, a variety of mix and volumes were run between the two products.
2. All of the IX testing units were using SDH handling units. In other factories, testers were being converted to Summit handlers. Handlers are the tools that transfer product from the loading trays to the tester. Summit handlers operated faster, but were new to the factories making data collection even more difficult.

The number of operational tools by WW is shown in Table 4.1 (the tool quantities have been disguised and do not necessarily agree with some of the statements made in the thesis). As discussed in Section 3.1.1.1, tool bagging is a popular activity. The factory was still in a ramp during WW28-34, so more tools were installed than were needed at certain times. The pressure to demonstrate high MUs is really only applied to the expensive tools. Epoxy and SCAM tools were bagged when the capacity models predicted excess tools. The quantity of bagged tools are also shown in the figure.

Table 4.1: Tool Counts by Work Week

Toolset	WW28	Bagged Qty	WW29	Bagged Qty	WW30	Bagged Qty	WW31	Bagged Qty	WW33	Bagged Qty	WW34	Bagged Qty
Reflow	10		10		10		10		10		10	
Mount	10		10		10		10		10		10	
Saw	25		25		25		25		25		25	
APL	20		20		25		20	21st up for 2 days	25		25	
Die Plate	15		15		15		15		15		15	
SCAM	40	8	40	8	40	8	40	8	40	4	40	4
Deflux	15		15		15		15		15		15	
Epoxy	40	4	40	4	40	4	40	4 for 3.5 days	40	4	40	4 for 100 hrs
Cure	25		25		25		25		25		25	
CTL	20		20		20		20		20		25	
BLU	70		70		75		75		70		70	
BI Oven	70		70		70		70		70		75	
IX Testers	40		40		40		45		45		45	
Laser Mark	15		15		15		15		15		15	
Ball Attach	15		15		15		15		15		15	
Ball Attach Inspect	15		15		15		15		15		15	

The overall factory performance data are shown in Table 4.2 (starts data are disguised and do not reflect actual values). Starts and output by product along with TPTs are given. TPT has a high variation due to queue time variation, so TPTs are represented by two statistics: the average and 90th percentile values. The reason for choosing these variables is simply because they are easily provided by the database. The target WIP level for the factory is 5 days and the theoretical TPT of the process is about 1.6 days which leads to an expected TPT of 6.5 days. In most cases, the average TPT is around 5 days. In reality, the third block is ignored since it is after the constraint (test) and product usually spends less than 1 day in block 3. It is also worth noting that with a total TPT of 5 days, each lot is spending an average of 3.4 days in a queue. Note that the starts data and all factory output data in this thesis have been disguised and are not necessarily consistent with each other.

Table 4.2: Factory Performance Data

	WW28	WW29	WW30	WW31	WW33	WW34
Product A						
Starts (ku/wk)	401.0	333.6	436.5	529.0	536.3	299.8
Average TPT (days)	5.1	5.0	4.8	4.2	4.8	4.6
p90 TPT (days)	6.9	6.7	7.0	5.4	6.3	7.6
Product B						
Starts (ku/wk)	306.6	392.3	345.5	246.8	378.8	517.0
Average TPT (days)	5.1	5.2	4.9	5.4	5.1	4.2
p90 TPT (days)	6.2	7.6	7.0	7.3	6.8	5.4
TOTAL LOADING		725.9	781.9	775.8	915.1	816.8

4.3.1. Data Collection for the Model

First, the appropriate time horizon for the data had to be determined. Factory and tool performance data are commonly reported in weekly units, so it was decided that the project would be unnecessarily complicated by attempting to change the time units.

The primary input to the model was tool performance and availability data. Unfortunately, accurate data collection turned out to be nearly impossible to find. Very few automated data collection systems exist in ATM factories. In fact, conscious decisions were made by the Assembly and Test Development (ATD) group to eliminate common data communication ports on most tools for cost savings. Two types of data needed to be collected for all of the tools in the factory:

1. *Availability parameters*
2. *Performance parameters*

Data were collected from the Philippines in addition to the Costa Rica factory. It was initially believed that the Philippines factory had the most accurate data, but this belief was dispelled after spending a week in the factory interviewing personnel. Where possible, Costa Rica data were used for the model. Costa Rica did not have data for SCAM or ball attach inspect, so data from the Philippines factory was. The Copy Exactly! policy for tool use made this a reasonable substitution.

4.3.1.1. Tool Availability Parameters

As discussed in Section 3.5.1, availability data come from CEPT, a system which is provides inherently inaccurate data. The engineers extract availability data from CEPT, although their definition of availability is simply 100% - (percentage of time the tools spends in repair plus the percentage of time the tool spends in PMs). By computing downtime in this manner, they have effectively stated that setup times, conversion times, and change consumable times belong to manufacturing. This lack of understanding of the importance of availability is also demonstrated by the absence of cross-collaboration between engineering and manufacturing on important issues such as setup time and conversion. Additionally, all of the engineers agree that the CEPT data are inaccurate, but go to great lengths to record the data and compare performance between the factories. Unfortunately, this was the primary source for availability data. Data are stored weekly by tool. These data were collected and IE estimates of setup and conversion time (a fixed number per week) were subtracted from the engineering availabilities. The resulting availabilities were used for most of the toolsets.

A few toolsets did not have availability data at all: cure, ball attach, die plate, reflow and mount. In the case of reflow and mount, two tools of each type exists in each factory and are redundant. Presumably the tools are quite stable and the large amount of excess capacity at these operations does not lead to the need for availability data – this seems like a reasonable assumption. Ball attach actually attempted to use CEPT data, but it was so inaccurate they did not even record it (or make an attempt to improve it. The answer was to ask IE to do a comprehensive MU study). During almost each interview with an engineer, they were quite eager to provide their availability data and demonstrate how much work had been put into it. After asking a few questions, it was obvious that no one trusted the data. It makes sense to simply stop collecting inaccurate data, although no one ever mentioned this idea to me. It appeared more that the data were collected to give the engineers something to do. MTTR and MTBF were calculated for each tool from the CEPT data. Perhaps the engineering organization could focus on data quality and be significant players in the automated data collection effort.

PM data, change consumable data, setup time, and conversion time were obtained through interviews with engineering and manufacturing. No definite times were recorded, so time ranges were gathered from the interviews and used for the model. It should be noted that over 50 hours were spent simply interviewing manufacturing and engineering personnel to obtain these data.

IE-initiated MU studies were conducted on a few toolsets: test, epoxy, SCAM, and ball attach. The SCAM and ball attach studies failed due to poor operator training/logging discipline and were abandoned. The test and epoxy studies required operators to log major tool events (PM, repair, change consumable, idle time) on a piece of paper or enter the data into a spreadsheet. The IEs compiled these data on a weekly basis for each tool. While these data appeared to be more trustworthy, no one had complete confidence in them. Availability data from the MU studies were used for the epoxy and test modules.

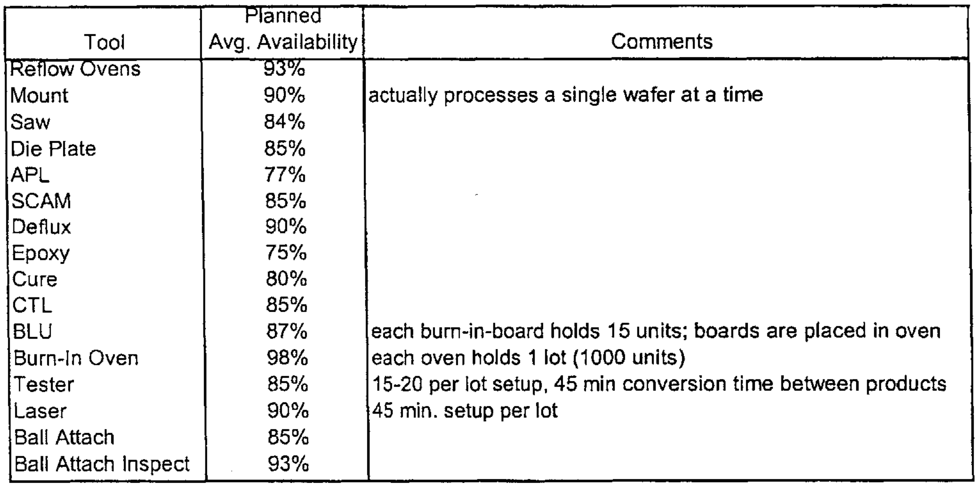
In the case of ball attach, die plate, reflow, and mount, data were taken from the IE capacity planning spreadsheets to create the availability distributions. These data were provided by the vendor and the process development group and consisted of MTTR, MTBF, PM schedule, MTTA, MTBA, and change consumable times. Single number averages were provided for all of these quantities.

4.3.1.2. Tool Performance Parameters

Tool performance parameters were much easier to collect because they are much easier to measure. Processing times are generally expressed as run rates (units/hour or uph). This is simply the inverse of processing time. This parameter can be easily and accurately measured by an IE with a stopwatch. Processing times were assumed to have no variation since the operations were automated. While the processing time may slightly vary in actuality, the variation is minor compared to the availability variation and was therefore neglected.

The tool performance parameters are shown in Table 4.3. The mount and saw rates vary by product because the die are different sizes which leads to a different number of die on each wafer. In this case, Product B die are smaller than Product A. For the tester, the Product B test times are longer than Product A primarily because it's a more complicated device with more transistors. Laser mark runs extremely fast, but requires long setups.

Table 4.3: Tool Performance Parameters



Tool	Planned Avg. Availability	Comments
Reflow Ovens	93%	
Mount	90%	actually processes a single wafer at a time
Saw	84%	
Die Plate	85%	
APL	77%	
SCAM	85%	
Deflux	90%	
Epoxy	75%	
Cure	80%	
CTL	85%	
BLU	87%	each burn-in-board holds 15 units; boards are placed in oven
Burn-In Oven	98%	each oven holds 1 lot (1000 units)
Tester	85%	15-20 per lot setup, 45 min conversion time between products
Laser	90%	45 min. setup per lot
Ball Attach	85%	
Ball Attach Inspect	93%	

4.3.2. Models CR OLGA WW28-34

While the details of the actual modeling code are not necessary for the purposes of this thesis, insight into the mechanics of the model are useful. An attempt to capture all of the variation in the factory would lead to a terribly complex model, so assumptions were made in order to manage the model's complexity. First of all, actual factories experience lots that go on hold to await engineering disposition (typically less than 8%) and engineering lots (usually less than 2% of total volume). While the presence of these two factors impacts factory performance, the ROI for modeling these events is low. Hold lots add variation to the overall TPT and this was accounted for by only considering the 90th percentile for the TPT data. This statistic is questionable, but it routinely given in output reports and was judged appropriate for this model. The impact of engineering lots was assumed to be small compared to the variation in toolset availability.

The initial operations actually use lots containing 25 wafers (reflow, mount, saw). The model simply used larger lots at these initial operations to represent the average number of working die per wafer lot. After the saw operation, the lots were divided into 1,000 unit lots. Yield data for each operation were averaged over WW28-34 at the Costa Rica factory. The model yielded this average percentage of units at each operation and resulted in lots of less than 1,000 moving through the factory.

The testing operation is the most complex in the factory and required some simplification and assumptions. After extensive interviewing, I concluded that tester conversions occurred at the discretion of the supervisors. To approximate the real performance, each tester ran a single product on the PBIC test until the queue of a product was empty. Once this happened, a tester was converted to another product for the PBIC test.

In Costa Rica, testers were converted to FQA test once the total SFGI queue reached 5,000 units. Only 250 units of each lot were tested at FQA and the 5,000 units represents the total units in queue, although only a fraction of those units will be tested (depending on the number of lots). The model only ran FQA on a single tester and converted it once the queue reached 5,000 units. Once the FQA test finished, the tester was converted back to PBIC.

After passing the PBIC test, units are divided into new lots at the SFGI operation according to their speed or bin split. The goal of this model was to improve capacity planning, not to model the details of the testing operation. Therefore, bin splits were not modeled since product speed does not impact factory performance at the remaining operations after SFGI. Lot size does impact laser mark (long setup times for each lot) and SFGI. The SFGI distributions shown in were approximated using a bimodal distribution for Product A and a normal distribution for Product B.

Two types of rework exist at test: re-test and re-burn. Re-test simply involved re-testing units at PBIC; a unit could be re-tested up to three times. Average percentages for the re-test rates were calculated for WW28-34 in Costa Rica and were found to be fairly low for both products. To model this, the test time required for each unit was increased by a small percentage and the yield at test was adjusted appropriately.

A percentage of units fail PBIC in a certain fashion and required re-burn. Units in need of re-burn are grouped together and sent back to the burn-in area where they are loaded into ovens, re-burned with the standard recipe, and the tested at PBIC again. The percentage of units requiring this again is a very small percentage. In the model, a small fraction of all lots at PBIC were sent on a re-burn route where they were inserted into the BL load queue and processed normally through the rest of the line.

It takes time to move lots from one operation to the next. A delay of 12 minutes was given to each lot as it moved from one operation to another in an attempt to model this transport time.

4.3.3. Warm-up period determination

Before validation could commence, the necessary warm-up period for the model had to be determined. All simulations need a certain time period to stabilize before meaningful results can be obtained. For each simulation, the factory started empty and it took time for the WIP levels to reach a pseudo steady state. Figure 4.1 shows factory output as a function of time for both products. The factory actually reached a stable level in the relatively short period of 3 weeks.

Figure 4.1: Simulated Factory Output over Time

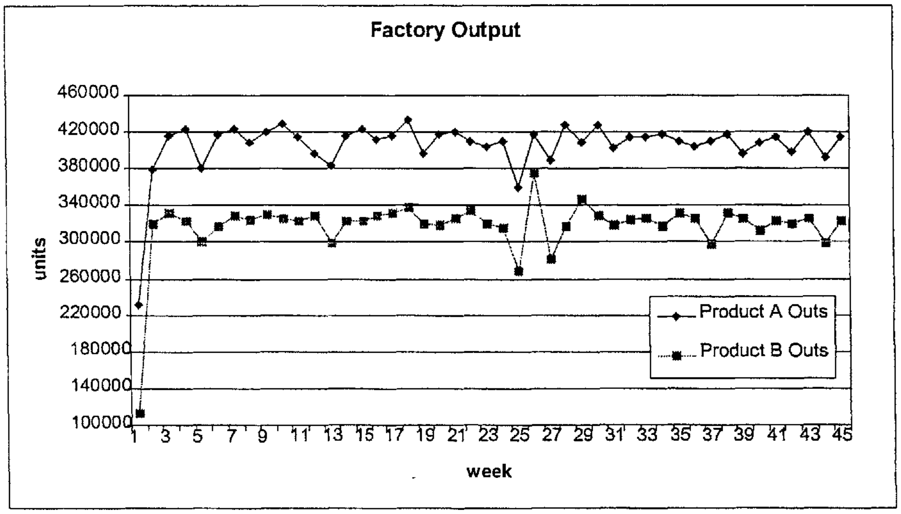


Figure 4.2 and Figure 4.3 show utilizations and queue sizes for tester and epoxy tools – the results are similar. To be conservative, a warm-up time of 7 weeks for chosen for all simulations. For all experiments, data for weeks 1-7 were omitted and results were only reported for weeks 8 and higher. If an experiment was stated to have run for 10 weeks, the simulation was actually run for 17 weeks, but the first 7 weeks of data were omitted.

Figure 4.2: Epoxy and Test Utilization over Time

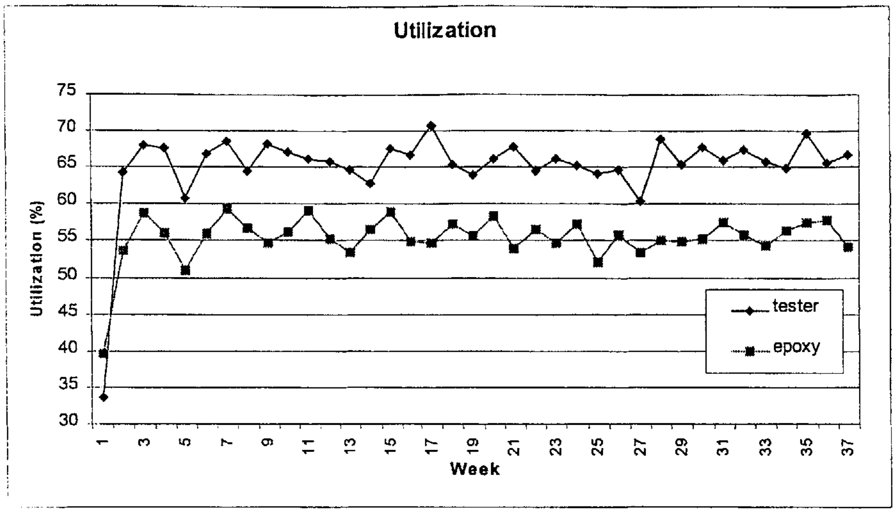
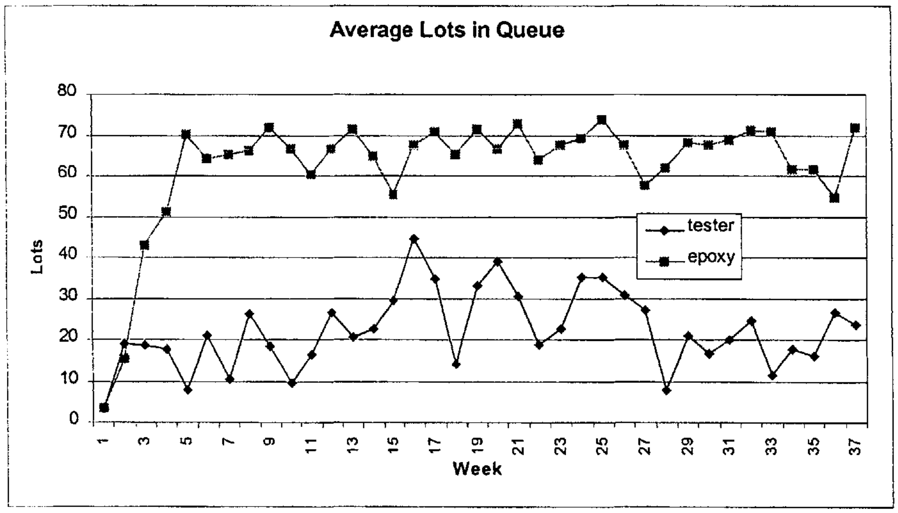


Figure 4.3: Average Lots in Queue for Test and Epoxy over Time



4.3.4. Validation Process

Validation of the model was a two step process. First, the simulation availabilities needed to be validated against the availability distributions from the factory data. Although confidence in the accuracy of the factory data was low, this was a necessary first step to highlight any obvious problems with the factory data.

Second, the simulation needed to be compared against the actual factory for WW28-30. Factory output and queue times for each operation were used as the final validations for the model. The choice of factory output is somewhat obvious; this is ultimately the most important measurement of the factory which makes it crucial to validate against. The queue times of each operation allow for total TPT comparison as well as providing details on the accuracy of the model for each operation. Again, these comparisons would highlight any discrepancies between the model and reality and allow them to be addressed before moving forward.

4.3.4.1. Comparison Between Factory Availabilities and Simulation Availabilities based on Engineering Failure Data

The simulation was run for 10 weeks and weekly tool availabilities were recorded. The resulting distributions were compared against the factory data obtained during the data collection process. The results are compared using box and whisker plots in Appendix A (availability data were not obtained for reflow, mount, or ball attach, so graphs are not shown for these toolsets).

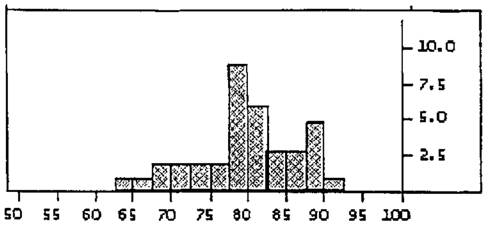
In all cases, the variability predicted by the model is much lower than the variability shown in the factory data. Additionally, the range of model availabilities is generally higher than actual data. Since there is not an automated data collection system, these results are not surprising. There is not an emphasis placed on accurate failure, assist, and PM data. Also, the operators probably reduce down times for fear of looking bad.

In conclusion, the failure, assist, and PM data gathered does not lead to availability distributions consistent with factory availability data (which is also highly suspect). It is hard to believe that there is less variation in the factory data than what is shown in the factory availability distributions; rather, the opposite it likely to be true. As a result, the availability component data (failure, pm, assist, etc.) were deemed inaccurate and the decision was made to simply match the factory availability distributions.

4.3.4.2. Comparison Between Factory and Simulation Availabilities

The original intent of the model was to create tool availability distributions using repair, PM, and conversion data. The large discrepancies between shown in the previous section lead to the simple matching of the availability distributions for each tool. Triangular distributions were used to model the actual tool availability distributions. A triangular distribution consists of a peak value (mode) and minimum and maximum values. Figure 4.4 shows the tester weekly availability (for individual tools) for the Costa Rica factory. The distribution has a long tail toward lower values and a relatively steep tail toward higher values. This typifies most of the availability distributions for all the tools and is expected. Given PM schedules and failure rates, a maximum availability of less than 100% is expected which leads to a shorter tail above the peak. The lower limit is 0% and unusual downtimes serve to lengthen the lower tail. A normal distribution would not be appropriate to model this behavior. Given the inherent inaccuracy of the data and the non-symmetric nature of the distribution, a triangular distribution works well to approximate the actual availability distributions. These distributions are best described using a peak value (mode), a 10% quantile and a 90% quantile. These quantiles were chosen to eliminate any unusual activity in the tails of the distributions. For the distribution in Figure 4.4, the peak is 79.4, the 10% quantile is 69.5% and the 90% quantile is 88.9%.

Figure 4.4: Costa Rica Tester Weekly Availability Histogram



The simulation was modified so that each availability distribution was modeled as a simple triangular distribution. The attempt to create the availabilities using the reported PM schedules and failure data was abandoned. Setups, conversions, and change consumables remained in the model.

Table 4.4 shows the actual factory distribution statistics and compares them with the model distribution statistics (the model was run for 12 weeks to generate the distributions). In most cases, the model matches the data provided by the factory engineers within a few percent. Given the large amount of variation in the factory, this accuracy in distribution matching seemed adequate.

Table 4.4: Factory and Model Tool Availability Distribution Statistics

	Factory Data			Simulation Data			Difference % (Actual-Model)		
Tool	Peak	10%	90%	Peak	10%	90%	Peak	10%	90%
Reflow	n/a	n/a	n/a	91.6	86.9	93.5			
Mount	n/a	n/a	n/a	97.8	95.5	97.9			
Saw	79.6	63.7	85.1	76.2	62.3	87.0	4.3%	2.2%	-2.2%
Die Plate	94.1	91.6	97.3	92.6	91.4	96.1	1.6%	0.2%	1.2%
APL	84.6	73.5	86.8	86.2	72.6	88.3	-1.9%	1.2%	-1.7%
SCAM	90.7	74.5	92.9	89.3	71.6	94.0	1.5%	3.9%	-1.2%
Deflux	90.4	73.3	91.0	88.1	74.8	93.5	2.5%	-2.0%	-2.7%
Epoxy	76.2	65.6	83.4	78.4	68.4	83.4	-2.9%	-4.3%	0.0%
Cure	n/a	n/a	n/a	95.1	86	96.7			
CTL	91.9	87.7	94.8	92.1	89.1	96.4	-0.2%	-1.6%	-1.7%
BLU	92.6	81.2	95.3	91.4	83.5	92.5	1.3%	-2.8%	2.9%
Burn-In Oven	93.9	85.1	100.0	92.6	86.4	98.0	1.4%	-1.5%	2.0%
Tester	79.4	69.5	88.9	82.1	70.1	85.7	-3.4%	-0.9%	3.6%
Laser	81.8	75.7	83.3	83.4	75.7	83.8	-2.0%	0.0%	-0.6%
Ball Attach	n/a	n/a	n/a	84.7	80.3	85.0			
Ball Attach Inspect	93.8	87.7	94.1	94.7	89.9	96.1	-1.0%	-2.5%	-2.1%

4.3.4.3. Results of WW28-30 Validation

After the availabilities were corrected as described in the previous section, the model was loaded with the factory data (loading by product and tool count) for each of work weeks 28, 29, and 30. The model was run for 15 weeks and the model output compared to the actual factory output and queue times for each operation. For all three weeks, the initial runs over-predicted output by at least 25% and under predicted queue times by an even larger margin. Clearly, there was more variation in the factory than the model was capturing.

The additional variation in the factory had two sources:

- Inaccurate tool availability data: It is highly likely that the availability data is inflated and has lower variation than reality due to the inaccurate manual collection system.
- Contribution of labor to variation: The model assumes that if a tool is available and WIP is available at that operation, the WIP is immediately loaded on the tool and processed. In reality, this may not be a good assumption. A tool may sit idle for some period of time while waiting for an operator to load a new lot. This contribution was initially assumed to be small, but may in fact be quite significant.

In order to improve the model's accuracy, a 'Wait for Operator State' was implemented. A delay was created each time a lot was loaded onto a tool. This delay was normally distributed and adjusted until the model unit output and queue times approximated the actual performance data for WW28-30. In essence, the Wait for Operator state simply reduced a tool's availability. A typical wait for operator state had a mean of 12 minutes and standard deviation of 3 minutes.

As mentioned above, the Wait for Operator state encompasses both inaccuracies in tool availability data and labor delays. Until more accurate data collection systems are introduced into ATM, it will be impossible to measure the contribution of each of these two factors.

Obtaining correct Wait for Operator factors proved to be a very time consuming and iterative process. If nothing else, it demonstrated how slight changes in tool availabilities affect WIP dynamics. Wait operator states were adjusted until queue times for each operation and total factory output approximated actual factory performance for WW28-30.

Early on it was discovered that the burn-in area had some anomalies that could not be compensated for by simply adjusting the wait for operator state. The actual factory data showed that the queue time for burn-in board load (BL) was substantially longer than the burn-in board unload (BU) queue time (see the Burn In Load and Unload graphs in Appendix B – the difference was often greater than 10 hours). This area is thought to often be constrained by the total number of burn-in boards in the factory. The model was modified to account for the burn-in boards (BIBs – Costa Rica provided an inventory estimate), but the queue times still did not approximate actual performance. Lastly, operations reported that burn-in was a poorly run area and that labor inefficiencies may be extremely high. In light of all these variables, the decision was made not to validate against BL and BU queue times.

Appendix B shows the queue times for each operation for each of WW28-30. The actual factory performance is shown for both products, while the model prediction is shown as a single value. Because conversions were not needed at all operations (except for test), the model did not have a reason to produce significantly different queue times for each product. Thus, it did not add value to display queue times for both products. The actual average queue times were reported along with the 90th percentile queue times for each product. For the model, average queue times and the maximum queue time were reported. Since the model did not place lots on hold, the 90th percentile actual queue times were compared to the model's max queue time. While less than 10% of the actual factory lots were on hold at any given time, this seemed like a reasonable way to compare the amounts of variability between the two and allow for a reasonable wait for operator standard deviation to be used. While this prevented a rigid statistical comparison between the two numbers, the factory did contain more variability than the model could capture through hold lots, engineering lots, and other delays in the system. For the test operation, the decision was made not to differentiate the queue times by product because they were essentially identical (<1%) in all cases.

We do not understand why the actual queue times vary so much by product. For example, SCAM shows equal queue times for the two products in WW28, while epoxy shows differences of more than 8 hours for WW28-29. It can only be concluded that this resulted from an emphasis to push a certain product through the constraint faster, although the 'hot product' appears to change weekly. Since setup times are essentially non-existent at most operations, the model processed material in a FIFO manner (regardless of product type) which resulted in the queue times being nearly identical for the different products.

The goal was to predict the average queue time for the two products while coming close to approximating the variability which was captured through the max queue time comparison. The comparisons are shown graphically in Appendix B. A comparison between the average queue times is shown in Table 4.5. The average actual queue times are a weighted average based on production starts for each product in a given week. The tabular comparison highlights the near randomness of factory performance. In one week, the queue time will be under-predicted, only to be over-predicted the following week. It is not reasonable to expect the model to match perfectly on this level of detail, but rather to simply 'come close' to actual factory performance. It is more important that the model accurately predict output and total TPT while providing general insights to queue times at each operation. The comparison between maximum predicted queue times and actual 90% queue times is graphically presented in Appendix B. Table 4.5 clearly shows the reason the BL and BU operations were omitted from the model – the errors are huge.

The Wait for Operator states used to obtain the results in Table 4.5 are shown in Table 4.6. This table shows the weekly average and standard deviation of the availabilities, utilizations, and wait for operator states for 8 weeks of model runs (Costa Rica WW30). Only the saw data provided by engineering encompassed the variability seen in the factory data and did not require the wait for operator delay. The prominent lesson from this table is that the availability data are much too optimistic. In most cases, a wait for operator delay that was greater than 20% was needed to approximate the observed queue times. This indicates that tool availabilities are much lower than the factories believe they are. Remember that the wait for operator delay encompasses both inaccurate availability data and operator delays and that it is impossible to distinguish between the two. However, that does not detract from the fact that tools are available about 20% less of the time than anticipated. This fact alone warrants a better data collection system to determine the cause of this availability loss. The model utilizations were compared to the reported factory utilizations for test and epoxy and matched within 1%. The need to determine to cause of the loss of availability cannot be overemphasized.

Table 4.5: Comparison of Actual and Predicted Queue Times

	WW28 Queue Times (hours)			WW29 Queue Times (hours)			WW29 Queue Times (hours)		
Operation	Actual Average	Model Average	Difference	Actual Average	Model Average	Difference	Actual Average	Model Average	Difference
Reflow	0.4	0.3	0.1	0.4	0.3	0.1	0.3	0.3	0.0
Mount	0.5	0.4	0.1	0.5	0.3	0.1	0.2	0.4	-0.1
Saw	0.4	0.3	0.0	0.2	0.3	-0.1	0.4	0.4	0.0
Die Plate	0.9	1.0	-0.2	1.3	0.8	0.5	1.0	1.0	-0.1
APL	1.4	2.0	-0.5	2.6	1.0	1.6	1.9	3.7	-1.7
SCAM	4.1	3.7	0.4	2.7	2.5	0.2	2.6	3.3	-0.7
Deflux	0.9	0.9	-0.1	0.7	0.9	-0.1	1.1	1.0	0.1
Epoxy	9.9	15.0	-5.1	10.0	7.3	2.7	11.3	8.1	3.2
Cure	1.1	0.8	0.3	0.6	0.7	-0.1	0.6	1.1	-0.4
CTL	2.7	2.4	0.3	1.2	1.8	-0.6	2.5	2.3	0.3
BI Load	10.1	0.0	10.1	3.5	0.0	3.5	1.8	0.1	1.7
Burn In	1.2	1.4	-0.2	1.0	0.7	0.2	1.0	1.2	-0.2
BI Un-load	1.3	0.0	1.3	1.1	0.0	1.1	0.9	0.1	0.8
PBIC	23.5	8.6	14.9	28.5	30.7	-2.2	30.3	32.2	-1.8
Laser Mark	2.5	1.5	1.0	1.0	1.6	-0.7	0.9	1.6	-0.6
FQA	4.9	7.6	-2.8	8.0	8.3	-0.2	7.7	8.6	-0.8
Ball Attach	2.1	1.6	0.5	1.6	2.0	-0.4	1.5	2.2	-0.7
RVSI	0.8	1.2	-0.4	1.3	1.1	0.2	0.3	1.2	-0.9

Table 4.6: Availability, Utilization, and Wait for Operator States for WW30

	Availability %		Utilization %		Wait Operator %	
	average	std dev	average	std dev	average	std dev
Reflow	90.0	2.8	16.6	0.6	3.5	0.3
Mount	94.5	1.4	9.8	0.3	19.9	1.0
Saw	74.6	4.4	35.4	1.2	0.0	0.0
Die Plate	89.0	2.4	37.1	1.3	34.1	1.2
APL	78.5	2.3	50.9	2.2	22.8	1.0
SCAM	82.0	2.5	49.5	2.7	27.4	1.5
Deflux	75.9	3.0	29.9	1.6	28.6	1.6
Epoxy	73.1	2.4	55.5	3.1	17.1	1.0
Cure	88.1	6.8	42.6	2.6	23.1	1.4
CTL	82.9	1.7	46.7	3.4	29.7	2.1
BLU	79.9	1.2	52.0	4.0	9.5	0.7
Burn In Oven	91.8	1.4	64.6	5.1	19.0	1.4
S9k	78.2	2.3	65.2	5.2	12.0	0.9
Laser Mark	79.5	3.4	24.8	2.1	39.0	3.3
Ball Attach	83.0	3.0	49.5	4.2	25.7	2.2
RVSI	85.8	2.5	54.0	4.8	22.8	2.0

As previously mentioned, a better measure of model accuracy is its ability to predict factory output and TPT since it is unreasonable to expect any model to capture the near random behaviors at each operation week after week. The model product output results are compared with the actual factory output results in Table 4.7. Since factory output varies, the model output was averaged over 8 weeks to arrive at the reported value. As shown, the model matched actual factory output within approximately 5% for the three weeks. This fact that output was accurately predicted for three weeks with different product loadings and tool counts adds a great deal of credibility to the model.

Table 4.7: Comparison of Model and Factory Unit Outputs for WW28-30

	WW28		WW29		WW30	
	Product A	Product B	Product A	Product B	Product A	Product B
Average Model Output (units)	420309	312983	334833	390688	414482	322938
Actual Factory Output (units)	397800	304000	330866	389194	432960	342688
Difference (Model-Factory)	5.7%	3.0%	1.2%	0.4%	-4.3%	-5.8%

Since no attempt was made to accurately model the BL and BU operations, TPT estimates were not expected to closely match actual values. As shown in Appendix B, the BL and BU queue times are unusually high compared to the model (especially in WW28), so a TPT adjustment factor was needed to match the model and actual TPTs. Table 4.8 shows the predicted and actual TPTs for WW28-30. A constant factor of 0.7 days was added to the model for each week. This factor captures delays not incorporated in the model along with the unusually high BL and BU queue times for these weeks. The model did a reasonable job of predicting TPT and was within 10% for all cases.

Table 4.8: Comparison of Model and Factory TPTs for WW28-30

	WW28		WW29		WW30	
	Product A	Product B	Product A	Product B	Product A	Product B
Actual TPT (days)	5.08	5.07	4.99	5.2	4.77	4.89
Model Average TPT (days)	4.64	4.68	5.02	5.01	5.24	5.31
Difference	-8.7%	-7.6%	0.5%	-3.6%	9.8%	8.5%

4.3.4.4. Results of WW31, 33-34 Simulation Prediction

Validation of the model for WW28-30 was a precursor to further tests of the robustness of the model. Factory data (product starts and tool counts) were loaded into the model for WW31, 33-34. The model was run for 8 weeks and the output was averaged over the entire 8 weeks. The model TPT and output results are compared to actual factory values in Table 4.9. The output values match within 10% which again seems quite reasonable. TPTs are over-predicted by approximately 30% using the constant factor of 0.7 days added to the model TPTs. If this factor is removed, TPT prediction becomes more accurate and is within approximately 20% for all cases. Removal of the TPT factor may be warranted because BL and BU load queue times are much lower (see Appendix C) for these time periods than for WW28-30. If the main component was compensation for the high BL and BU queue times, it is reasonable to remove the factor.

Appendix C shows graphs comparing the actual and model average queue times for all operations along with a comparison of the maximum and p90 values. A summary of the results is shown in Table 4.10. In general, the difference between the predicted and actual queue times were highly erratic. Differences between the two values were often quite large, although the simulation would under-predict one week and then over-predict the following week. The BL/BU continue to baffle the model. There did not appear to be a consistent theme in the model error, although overall prediction of factory performance was worse than for WW28-30. These results underscore the need for better data in order to accurately predict factory performance. The error in availability data combined with the uncertainty of labor staffing and efficiencies, tool dedication strategies, and BL/BU management policies are simply too great to generate an accurate model. Before ATM procedures can be improved through the use of simulation, data quality must improve. However, the failure of this model to accurately replicate factory performance (even after the addition of significant adjustment factors) does not preclude its usefulness for running experiments that demonstrate the tactical and strategic uses of DES in the ATM environment.

Table 4.9: Comparison of Model and Factory Output and TPT for WW31, 33-34

	WW31		WW33		WW34	
	Product A	Product B	Product A	Product B	Product A	Product B
Average Model Output (units)	514035	237467	482956	336535	298422	500187
Actual Factory Output (units)	524800	244800	532000	375800	297400	513000
% Difference	-2.1%	-3.0%	-9.2%	-10.4%	0.3%	-2.5%
with constant 0.7 day factor						
Average Model TPT (days)	5.1	5.2	6.7	6.8	5.7	5.8
Actual Factory TPT (days)	4.2	5.4	4.8	5.1	4.6	4.2
% Difference	21.0%	-3.8%	40.8%	33.1%	25.5%	36.3%
without constant 0.7 day factor						
Average Model TPT (days)	4.4	4.5	6.0	6.1	5.0	5.1
Actual Factory TPT (days)	4.2	5.4	4.8	5.1	4.6	4.2
% Difference	4.4%	-16.7%	26.1%	19.3%	10.2%	19.8%

Table 4.10: Comparison of Model and Factory Queue Times for WW31, 33-34

	WW31 Queue Times (hours)			WW33 Queue Times (hours)			WW34 Queue Times (hours)		
	Actual Average	Model Average	Difference	Actual Average	Model Average	Difference	Actual Average	Model Average	Difference
Reflow	0.5	0.4	0.1	0.5	0.3	0.2	0.9	0.4	0.6
Mount	0.4	0.4	0.0	0.5	0.4	0.1	0.2	0.4	-0.2
Saw	0.5	0.5	0.0	0.5	0.4	0.1	0.6	0.3	0.3
Die Plate	1.1	2.7	-1.6	0.8	1.0	-0.3	1.4	2.1	-0.6
APL	4.9	0.2	4.7	4.3	0.9	3.4	2.2	0.2	2.0
SCAM	4.2	1.3	2.9	3.6	3.6	0.0	1.8	1.1	0.7
Deflux	1.0	1.9	-0.9	1.6	0.5	1.1	1.1	1.8	-0.7
Epoxy	3.7	14.8	-11.1	9.3	5.9	3.4	7.0	6.6	0.4
Cure	0.6	3.2	-2.7	1.8	1.0	0.7	1.0	1.6	-0.6
CTL	3.7	0.4	3.3	3.8	3.4	0.4	4.7	0.3	4.4
BI Load	2.3	0.0	2.3	3.7	0.1	3.6	1.8	0.0	1.7
Burn In	0.8	15.9	-15.1	1.0	1.5	-0.5	1.0	3.8	-2.8
BI Un-load	1.2	0.0	1.2	1.4	0.0	1.3	0.8	0.0	0.8
PBIC	22.3	8.1	14.1	7.0	32.9	-25.9	9.6	27.5	-17.9
Laser Mark	1.2	3.6	-2.4	3.5	1.4	2.2	1.9	4.3	-2.3
FQA	10.4	9.5	0.9	13.7	8.4	5.3	9.8	9.5	0.3
Ball Attach	2.4	16.8	-14.5	6.8	2.3	4.5	2.6	9.1	-6.5
Ball Attach Inspect	0.5	15.6	-15.1	0.5	1.5	-1.0	1.3	8.0	-6.7

CHAPTER 5. TACTICAL AND STRATEGIC USES OF DISCRETE EVENT SIMULATION

Discrete event simulation is useful for understanding a wide variety of tactical and strategic issues that face a factory. Although the simulation used in this project had to be heavily modified to approximate actual factory performance, that does not preclude experiments to demonstrate the usefulness of the tool. Extremely poor data quality lead to validation difficulties and this issue must be addressed before DES can be employed as a factory tool. The simulation did a good job of replicating data in WW28-30, so those operating conditions will be used to demonstrate the tool for the following experiments.

5.1. Tactical

Operations managers face tactical challenges daily. The lack of a tool that can accurately assess the impact of tactical operational decisions leads to operations managers relying on past experience and instinct to make these decisions. DES would enable managers to address issues using a data-driven approach. This requires a factory to have competency in DES and a dedicated staff to make sure the model is kept up to date. The need for these resources may initially seem unreasonable, so the following sections aim to demonstrate the usefulness of such a tool. It should be noted that after the completion of this thesis, a factory scheduling tool that utilizes DES is being implemented in several of the ATM factories. In addition to the suggestions presented in the following sections, future Intel projects may benefit from incorporating the learnings from this thesis into the factory scheduling software and process.

5.1.1. TPT Prediction

A simple and obvious use of DES is to predict factory TPT for a variety of loadings and product mixes. TPT is closely related to the CONWIP limits and TPT prediction must done in parallel with CONWIP block limit optimization. In a capacity constrained situation, DES could be used to predict factory TPT. A high-confidence delivery date could then be passed along to the customer.

Currently, Intel does not have any systems that allow TPT prediction. A new A/T process, known as Interposer 2 (INT2), will be deployed in the fall of 2000. This process involves several new toolsets and has more operations than any previous A/T process. The theoretical TPT of the process is approximately 4.0 days, compared to 1.6 days for OLGA. The methodology used by the IEs to estimate actual TPT is to take the theoretical TPT and multiply it by 3 or 4; this results in a 12-16 day TPT for INT2. Furthermore, the manufacturing systems organization is trying to establish kanban sizes for each operation. Besides the obvious conflict of incorporating a kanban system with a

CONWIP system, this is underscores the lack of fundamental understanding of manufacturing systems. The process for doing this simply involves guessing at how much WIP should be at each station. As this project has shown, the amount of WIP buffer necessary at an operation is largely dependent on the variability of the toolset. This is not even considered in estimating TPT.

A 16 day TPT will likely be unacceptable to ATM management, but the factories will not have a tool to aid in the reduction of TPT. If a DES were used for the process, the factory would have a much better chance of reducing TPT without risking output. This one example clearly demonstrates the necessity for DES in estimating TPT.

5.1.2. WIP policy optimization

As mentioned in the previous section, TPT prediction/improvement is closely related to CONWIP block size optimization. The OLGA factories run with block limits of 2-2-1 (days) and it did not appear that any attempt had been made to reduce these limits. The issue of reducing block limits is confounded by the obsession to bag tools. As tools in constraint toolsets are bagged, the ability to reduce block limits diminishes. DES provides a method to test lower block limits without risking actual production. The benefits of lower block limits are faster TPTs, which makes the factories more agile.

5.1.2.1. Block Size Optimization for Costa Rica WW28-30

A series of experiments were run to examine the effects of reducing the block 1 and block 2 limits of the OLGA line during WW29 operating conditions. The block 3 limit was ignored since block 3 is after the constraint (test) and does not significantly affect TPT. The goal was to reduce the block 1 and 2 limits as much as possible until the tester idle time became statistically greater than 0% and output was subsequently decreased. Table 5.1 shows the preliminary results from the block size optimization screening experiment. 2, 2 (block 1 limit, block 2 limit) was used as the baseline condition. Each set of block limits was run for 8 weeks and the TPT for both products and the tester idle time were averaged for all 8 weeks and presented in the table. Experiments were run diagonally up the table grid from 2,2 until tester idle time exceeded 0.0% (a tester idle time greater than 0.0% meant the tester was partially starved at some point and output was sacrificed because the WIP buffer in front of test was not large enough).

As shown, 1.75, 1.75 and 1.50, 1.50 both had 0.0% tester idle time and the decrease in TPT was as expected. For example, the 1.50, 1.50 simulation should have decreased TPT by approximately 1.0 day from the 2.0, 2.0 simulation because the system contained 1 less day of WIP. At 1.25, 1.25 the tester idle time was greater than 2.0%, so the combination of block limits between 1.25, 1.25 and 1.50, 1.50 were explored. 1.25, 1.75 and 1.75, 1.25 gave similar results, although the extremes of 2.0, 1.0 and 1.0, 2.0 resulted in significant tester idle times. This indicates the need for balanced WIP in the system and demonstrates the effectiveness of positioning blocks between constraints and near-constraints. As a final test, the output for each product under the 1.50, 1.50 limits was statistically compared to the output for the 2.0, 2.0 limits (Appendix D). The output for both cases is statistically equivalent while the TPTs for the two experiments are not equal. In summary, the block size reduction to 1.5,1.5 would lead to a 20% reduction in factory TPT without risking output. This was simply accomplished by using DES to optimize the amount of WIP in the factory.

Table 5.1: Block Size Optimization Results

		Block 1 Limit (days)				
		1.00	1.25	1.50	1.75	2.00
Block 2	1.00					0.4, 4.03
Limit (days)	1.25		2.2, 3.52	0.6, 3.76	0.1, 4.02	0.0, 4.27
	1.50	2.5, 3.39	0.9, 3.76	0.0, 4.03		
	1.75	1.5, 3.53	0.0, 4.00	0.0, 4.27	0, 4.53	
	2.00	1.0, 3.62	0.0, 4.24			0, 5.01

* *(average tester idle time, average TPT)*

The 1.5,1.5 limits were tested using the WW28 and WW30 conditions to help judge the robustness of proposed block limit reductions. For both weeks, the simulation was run for 13 weeks in order to improve the statistical conclusions from the data. For both weeks, the factory output did not decrease using the 1.5, 1.5 block limits while the TPT did decrease by an average of 20% (1 day).

When DES is not employed as a tool to optimize block limits, actual production is risked as the factory slowly lowers the limits until output is impacted. When a factory is demand constrained, such an exercise it too risky to undertake and factory continues to run with sub-optimal CONWIP limits and TPTs are unnecessarily long. DES allows for a scientific approach to TPT reduction without risking actual production output.

5.1.2.2. Results from drum-buffer-rope experiment

CONWIP limits are proposed to be more effective than a traditional drum-buffer-rope policies (Hopp and Spearman, 1996). This idea was tested by converting the simulation model to a drum-buffer-rope system in which new starts were determined by the test buffer size. In other words, new starts were only made when the test buffer fell below a certain level of WIP. This is in contrast to a CONWIP policy in which new starts are made when the block limits fall below a certain WIP level. WW30 data were used for the experiments.

The starting point for the experiments was 1.5 days of WIP in front of test. In other words, new starts would only be made when the test buffer fell below 1.5 days of WIP. The model was run and the output was highly erratic. The long delay between starts and the actual test buffer led to frequent tester starvation and exceedingly long TPTs. Logically this makes sense. The long delay does not allow the system to respond in real time. Instead, as WIP builds up in the front of the line due to tool instability, the test buffer size continues to decrease so more WIP is released into the line. The result is an unstable system with large WIP bubbles and frequent tester starvation. This result underscores the effectiveness of a CONWIP system in a manufacturing system where the 'rope' would prohibitively long due to feedback delays. An increase in the test buffer size may dampen the variability, but it would also lead to an increase in the already long TPT.

5.1.3. Expected Value Forecasting

As discussed in Section 3.4.3, factory output is not a single value (as stated by static capacity models), but is a distribution that results from the combinatorial nature of variation throughout the factory. A straightforward use of DES is to predict the output distribution of a factory for a given set of input conditions. The simulation was run with the WW29 product loadings and tool counts for 32 weeks and the output was observed. As Figure 5.1 and Figure 5.2 show, the output does indeed vary. The probability plot show that the output is nearly normal. If utilized correctly, this information would be extremely valuable to the sales organization during times when capacity is constraining the sales process. For example, the factory could confidently commit to producing 320ku/week of Product A (90% confidence) and could also give a 50% chance of producing 340ku/week. This type of information would allow Intel to take full advantage of its capacity. Customer demand is not a fixed number and also contains confidence intervals. Accurate statistical information on factory output would allow the sales organization to use factory variability to its advantage in dealing with customers. For example, potential factory upside (50% confidence output) could be sold to customers ahead of time with the understanding that the upside would only occur 50% of the time.

Figure 5.1: Variation in Product A Weekly Output for WW29

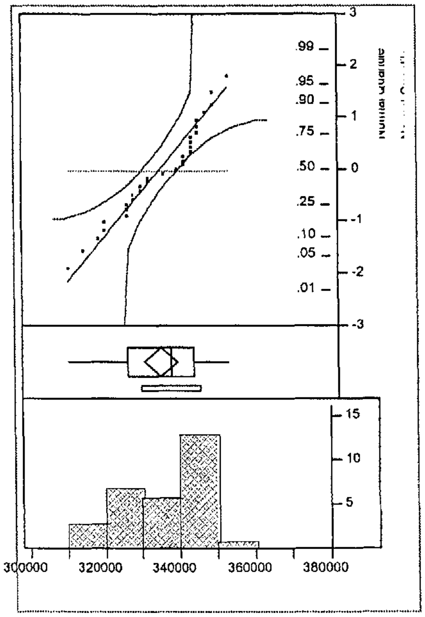
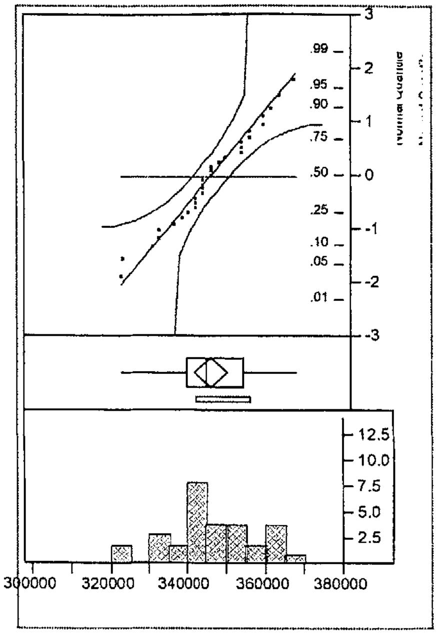


Figure 5.2: Variation in Product B Weekly Output for WW29



5.2. Strategic Uses of DES

5.2.1. Capacity Planning Optimization

The most obvious use of DES for strategic purposes is the optimization of capital investments in capacity. Factories are dynamic systems where WIP flows are determined by the independent actions of the toolsets and operators. DES allows for these complicated interactions to be anticipated. Additionally, the amount of excess capacity at non-constraint operations helps to determine the factory TPT. When planning a factory, a target TPT needs to be stated before capacity can be planned. This is in stark contrast to current factory planning methods in ATM where only desired output is stated; TPT is a result of the 10/15/20 policy. Using DES, output and TPT targets need to be stated before a factory is planned.

Several experiments were performed using the validated simulation model to explore potential capacity planning methodologies. Since the validation availabilities were much lower than the planned availabilities that the IEs are currently using for each tool, the validated availabilities (Table 5.2) were used for the experiments. Tool availability is defined as (tool availability – wait for operator time). Since the wait for operator state contained both inaccuracies in tool availability data and operator effectiveness factors, subtracting this quantity from the tool availability results in the actual amount of time the tool is available for production. After all, if a tool is up, but no operator is available to run WIP, the tool is essentially down to production. The average availabilities used are shown in Table 5.2.

Table 5.2: Average Toolset Availabilities Used for Experiments

Toolset	Average Availability
Saw	75%
Die Plate	55%
APL	59%
SCAM	55%
Deflux	47%
Epoxy	57%
Cure	66%
CTL	53%
BI Oven	73%
Tester	66%
Laser	40%
Ball Attach	57%
Ball Attach Inspect	62%

Reflow and mount tools were not considered in the capacity optimization experiments since each of them have such large capacities and are relatively inexpensive. One tool of each is more than enough to satisfy the capacity of any OLGA factory, but a second redundant tool is used in each factory to mitigate production loss risks. This policy of redundant tools is core to Intel's planning policies and was not considered for this project. Additionally, due to the large discrepancies found in the data from the BU and BL operations, these tools were also eliminated from the experiments. Further data is needed in this area before it can be properly modeled.

5.2.2. 10/15/20 Baseline Case

Before exploring new capacity planning methods, the current 10/15/20 policy was examined. Since most operations in ATM factories only have 3-5 tools per toolset, the gap targets are rarely met. For example, it the static capacity model shows the need for 3.2 saws, 4 saws will be purchased since incremental tools cannot be obtained. This is in contrast to the fab where most toolsets have 20-40 tools which allows for the gap targets to generally be met.

Incremental tools can be simulated in the model by inserting extra downtime on a given tool. For example, if 0.5 saws are needed, a single saw tool will be shut down for 3.5 days/week. Since the simulation is run for several weeks, this does a reasonable job of approximating how the factory would run if incremental tools could be installed.

Based in the availabilities shown in Table 5.2, the 10/15/20 gap policy was used to arrive at the number of tools necessary to meet the production needs of WW29 (Table 5.3). Using the standard capacity calculation methods used by the factory IEs, the gap was subtracted from the average availability to arrive at the expected utilization (MU). The resulting protective capacity (Eq. 2.4) was calculated for comparison purposes with other experiments. The number of tools required was calculated using Equation 5.1. For simplicity purposes, the uph for tester (the only tool where run rates vary by product) was calculated using a weighted average based on the production needs for WW29 (Table 4.2).

$$\text{Tools Required} = \frac{\text{Factory Output Required}}{MU * uph * 168(hours\,/\,week)} \qquad \text{(Eq. 5.2)}$$

Table 5.3: Tool Requirements for 10/15/20 Gap Policy for WW29 Production

Toolset	Average A	Gap	PC	Expected MU	Required Tools
Saw	75%	20%	36%	55%	7.0
Die Plate	55%	20%	57%	35%	9.3
APL	59%	20%	52%	39%	14.9
SCAM	55%	20%	58%	35%	24.8
Deflux	47%	20%	74%	27%	9.5
Epoxy	57%	15%	36%	42%	27.1
Cure	66%	20%	44%	46%	13.6
CTL	53%	20%	61%	33%	16.6
BI Oven	73%	20%	38%	53%	49.3
Tester	66%	10%	18%	56%	26.4
Laser	40%	20%	102%	20%	11.3
Ball Attach	57%	20%	54%	37%	12.1
Ball Attach Inspect	62%	20%	47%	42%	11.5

The number of required tools was placed in the model and the model was run with conservative block limits of 3-3-1. The results are shown in Table 5.4. The idle times for SCAM, epoxy, and test (the most expensive tools in the factory) are surprisingly high. Even more disturbing is the fact that the model fell well below the target output for both products. The TPTs are high because block limits of 3-3-1 were used to try to ensure that the constraints were fed at all times. As shown by the large idle time for the constraint tools, even these large block limits could not keep the constraints fed. The results of this experiment demonstrate the role of the variability in factory output.

In summary, these results suggest that if ATM factories could actually implement the 10/15/20 policy by installing 'incremental tools,' the inadequacy of this capacity policy would quickly be seen. However, due to having to round up to the next integer tool quantity for almost every toolset, the factories are spared this reality. In essence, 10/15/20 cannot be implemented correctly in ATM due to the small tool quantities in each toolset. The organization is spared this reality because of the large amount of excess capacity that is actually installed due to tool rounding. The actual amount of gap for the tools in WW29 is shown in Table 5.5. As shown, the gap is often twice as large as desired. For the expensive tools (test, SCAM, and epoxy), gaps are also above their target levels indicating the potential for significant capital savings. These results underscores the dramatic need for an improved capacity planning policy in ATM.

Table 5.4 : 10/15/20 Simulation Results

SCAM Idle Time	30%
Epoxy Idle Time	24%
Test Idle Time	19%
Average Product AOutput	285667
Std Dev/Average Output	17%
Average Product B Output	369600
Std Dev/Average Output	13%
Average Product A TPT (days)	7.58
Average Product B TPT (days)	7.65

Table 5.5: Planned and Actual Gaps for WW29

Toolset	Planned Gap	Actual Gap
Saw	20%	41%
Die Plate	20%	31%
APL	20%	21%
SCAM	20%	31%
Deflux	20%	47%
Epoxy	15%	21%
Cure	20%	28%
CTL	20%	24%
BL/BU	20%	58%
BI Oven	15%	2%
Test	10%	23%
Laser Mark	20%	53%
Ball Attach	20%	35%
Ball Attach Inspect	20%	39%

5.2.3. Protective Capacity Optimization

The original goal of this project was to determine a better method of planning capacity if the 10/15/20 gap policy was found to be inadequate. Ideally, Protective Capacity values would be optimized for each toolset and universally applied to all factories. With this goal in mind, a framework was created to optimize PC values. Although the 10/15/20 example showed that universal excess capacities lead to capital inefficiencies due to tool rounding, PC values were still optimized to demonstrate the inadequacy of the excess capacity levels stated by gap.

The first possible method of optimizing PC values was to iteratively add and remove tools until cost and TPT targets were achieved. While this is possible, it's a very time consuming process. The large disparity of capital costs between toolsets in ATM leads to a logical method of minimizing cost during the PC optimization process. The large difference in tool costs is shown in Table 5.6 (the tool costs have been normalized against testers which is the most expensive tool). The planned availabilities for each tool are shown. The purpose of the table is to provide a general sense of where the factory expense lies. The tool purchase cost was divided by the weekly output to arrive at the cost/unit of weekly output. This measure gives a more complete picture of unit costs than simply showing the tool costs. As the data show, test is a much larger component of unit cost than any other operation. Epoxy and SCAM are the next most expensive operations; the rest of the tools are relatively cheap compared to these three.

Table 5.6: Tool Cost Comparison Table

Tool	Planned Availability	Cost/unit/week (normalized)
Saw	84%	$ 0.09
APL	85%	$ 0.07
Die Plate	77%	$ 0.14
SCAM	85%	$ 0.54
Deflux	90%	$ 0.06
Epoxy	75%	$ 1.00
Cure	80%	$ 0.21
CTL	85%	$ 0.12
BLU	87%	$ 0.18
BI Oven	98%	$ 0.26
IX Testers	85%	$ 3.11
Laser Mark	90%	$ 0.03
Ball Attach	85%	$ 0.29
Ball Attach Inspect	93%	$ 0.09

These results highlight the need to minimize capital purchases at these three expensive operations and to add excess capacity (to manage TPT and output variability) at the remaining relatively inexpensive operations. In this spirit, the following methodology is proposed for PC optimization for a factory with large disparities in tool costs. The tool availability distributions were assumed to be roughly normal for the purposes of this methodology. If actual tool availabilities significantly deviate from normality, appropriate control charts should be employed.

1. Identify the tools that are the largest contributors to unit cost (2-3 tools). These tools should become the constraint and near-constraints for the factory.

2. Plot weekly availability data for each of the tools identified in step 1 using a x-bar control chart. Identify the 2.0σ and 2.5σ LCLs for each of the charts.

3. $\text{Constraint PC} = \dfrac{(\text{Average Availability - }2.0\sigma\ \text{ LCL value})}{2.0\sigma\ \text{ LCL value}}$

4. $\text{Near Constraint PC} = \dfrac{(\text{Average Availability - }2.5\sigma\ \text{ LCL value})}{2.5\sigma\ \text{ LCL value}}$

5. Using simulation, add capacity as needed at non-constraint operations until output and TPT goals are achieved. This is a highly iterative process. A good starting PC for all non-constraints is the -3σ PC value.

The control charts are used to determine the minimum tool availability that is achievable on a weekly basis. By choosing the 2.0σ LCL for the constraint and assuming the constraint will always be staffed with an operator and a WIP will always be maintained, a level of constraint output can be committed to with 95% confidence. In reality, the constraint will be able to exceed this committed output in most weeks, but this provides the factory with a level of output it can achieve 95% of the time. Since staffing and WIP levels are slightly more erratic at near constraint operations, the 2.5σ LCL is used. It is important to remember that staffing effectiveness is built into the model availabilities using the wait for operator state. If labor effectiveness factors are not built into tool availabilities, the method will not be reliable because it will likely over predict tool output.

The 2.0σ and 2.5σ control limits for the constraint and non-constraint operations (respectively) are somewhat arbitrary and need to be determined on a case by case basis. A balance between factory cost and constraint output confidence needs to be achieved. In a factory setting where the constraint is extremely expensive in comparison to other tools, the PC may be set at a value lower than 2.0σ. While this will result in lower factory costs, the likelihood of the constraint under-producing increases. This may be an acceptable trade-off and is at the discretion of factory designers. However, this methodology is proposed for a factory environment where 3-4 tools are much more expensive than the rest of the tools. In this type of factory, the proposed guidelines provide a reasonable design methodology. Factory planners do have the option of adjusting the control limits (which in turn adjust the PC levels for the constraint and non-constraint operations) depending on the risks associated with missed output and factory costs.

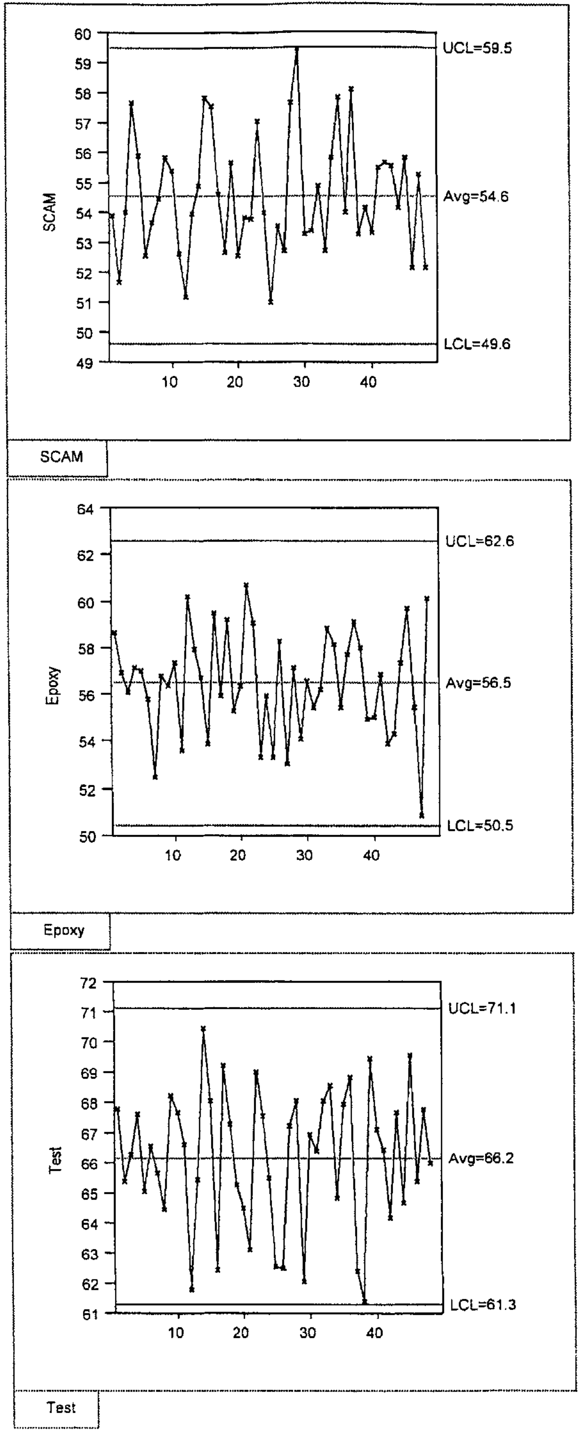
The OLGA simulation was run for 47 weeks under fully loaded conditions (WW29) and weekly toolset availability data were collected and plotted in x-bar control charts (Figure 5.3).

PC levels for the constraint and near-constraints can be calculated from the control charts. For example, the Tester PC is calculated as follows:

$$\text{Tester PC} = \frac{66.2 - 61.3}{61.3} = 8\%\,\text{PC}$$

Using the calculation, the SCAM PC = 7.9% and the Epoxy PC = 9.5%. This method of calculating constraint and near-constraint PC makes intuitive sense because it allows the factory to commit to output levels that it can confidently commit to, given the inherent variability of the factory. The Goal (Goldratt, 1992) teaches that factory output will be equal to constraint output. This method of determining PC for the constraints simply takes Goldratt's ideas one step further by considering toolset variability and its effect on factory output.

Figure 5.3: Availability Control Charts for SCAM, Epoxy, and Test



3σ control charts were created for the rest of the tools in the factory (reflow, mount, and BL/BU were excluded for reasons already discussed) and PCs were determined for each toolset. APL was not considered because it is a parallel operation and is not impacted by the dynamics of the rest of the line. APL capacity was always sufficient to ensure that SCAM was not starved for material coming from APL. Running the simulation at 3σ PC levels resulted in low, erratic output even with large block limits.

The initial goal was to create a factory that achieved the output for WW29 using a TPT target of approximately 6 days (the block 1 and 2 limits were set to 3 days each). Capacity was added at non-constraint operations in increments of 5% PC and the simulation was run. Adjustments were based on WIP queue sizes and tool idle time. The goal was to add enough capacity at non-constraint operations to ensure that test was the constraint with SCAM and Epoxy remaining the near constraint.
Each simulation was run for 8 weeks and the output was statistically compared to the output for the WW29 simulation (Table 4.7). A two-sided t-test with $\alpha=0.05$ was used to ensure that the hypothesis that the means of the two distributions were equal could not be rejected. Capacity was added until the output at a given TPT target met this statistical criteria.

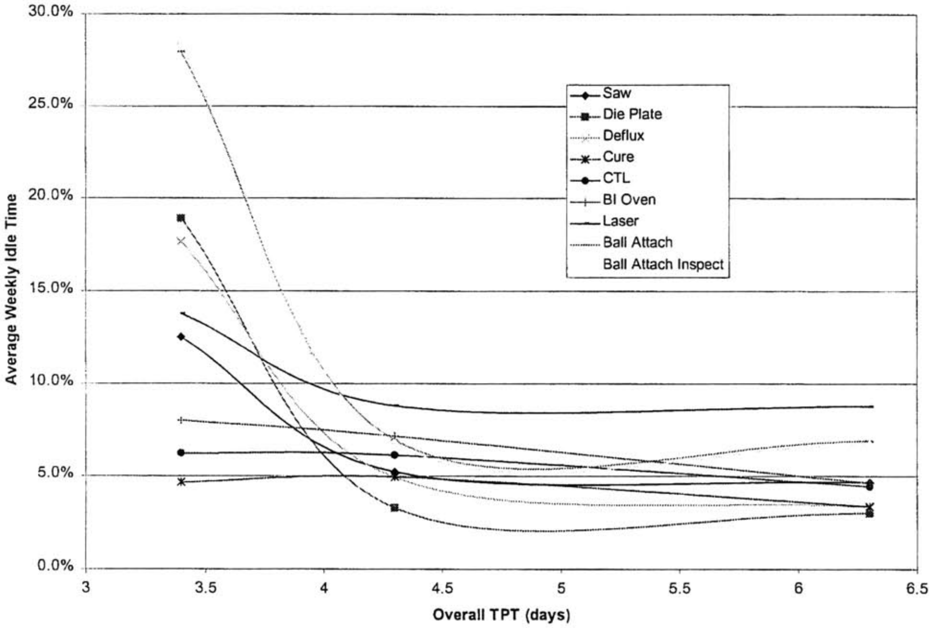
Capacity was iteratively added in the same manner to meet TPT targets of 3 and 4 days (Block 1 and 2 limits of 2.0 and 1.5 respectively). The results from this work are shown in Table 5.7. Basically, as the target TPT is decreased, the amount of excess capacity at non-constraint operations (PC) must be increased. This in-turn increases the amount of idle time at the non-constraint operations. In all three cases, the constraint remained at test as shown by the low idle time and large numbers of lots in its queue. SCAM and epoxy also remain the near constraints as shown by the same characteristics. The only exception is the 1.5-1.5-1 case for SCAM where it still have relatively low idle time, but only a few lots in the average queue. This simply shows the need to reduce constraint queue sizes in order to reduce overall TPTs; this can be accomplished by increasing excess capacity at non-constraint operations.

Figure 5.4 shows the relation between overall TPT and average idle time at non-constraint operations. As expected, idle time increases as overall TPT decreases. As excess capacity is added, WIP queues decrease which results in decreased overall TPT. It is also expected that this relation is non-linear; as the overall TPT approaches the theoretical TPT, the amount of excess capacity required to eliminate WIP queues in the face of factory variability increases. The amount of excess capacity required to reduce TPT was added in an iterative process in an attempt to keep the constraints fed and several combinations of non-constraint excess capacity would likely yield the same overall TPT. Therefore, only the general trend should be observed in this graph.

Table 5.7: Toolset PC Optimization Results

	3-3-1 Block Limits			*2-2-1 Block Limits*			*1.5-1.5-1 Block Limits*		
Toolset	PC	Average Idle Time	Average Lots in Queue	PC	Average Idle Time	Average Lots in Queue	PC	Average Idle Time	Average Lots in Queue
Saw	49.4%	4.7%	5.3	59.4%	5.2%	5.0	74.1%	12.5%	0.2
Die Plate	16.6%	3.0%	5.8	26.6%	3.3%	5.6	51.7%	18.9%	0.3
SCAM	7.9%	0.9%	33.3	7.9%	2.8%	12.6	14.3%	6.4%	0.7
Deflux	31.1%	3.3%	11.4	41.1%	4.9%	9.7	63.9%	17.6%	0.6
Epoxy	9.5%	2.1%	32.2	9.5%	4.8%	22.2	20.5%	12.2%	6.2
Cure	18.0%	3.4%	16.1	28.0%	4.9%	8.8	27.3%	4.6%	2.8
CTL	9.1%	4.4%	5.4	14.1%	6.1%	5.0	16.1%	6.2%	3.9
BI Oven	14.2%	4.6%	5.1	14.2%	7.1%	4.3	17.6%	8.0%	2.3
Tester	8.0%	0.0%	85.6	8.0%	0.5%	24.5	7.2%	0.3%	28.0
Laser	47.5%	8.8%	6.1	47.5%	8.8%	6.4	59.9%	13.8%	1.5
Ball Attach	18.6%	6.9%	10.0	18.6%	6.9%	10.1	52.5%	28.0%	0.5
Ball Attach Inspect	17.5%	6.7%	5.3	17.5%	6.8%	5.3	53.2%	28.4%	0.3
Average Product A Output	330000			330000			332000		
Average Product B Output	291000			290000			290000		
Average TPT	6.3			4.3			3.4		

Figure 5.4: Non-Constraint Idle Time as a Function of Overall TPT



5.2.4. Factory Agility Curves

A use of DES that was not explicitly explored was factory agility curves. Factory agility curves are complex graphs created to address what-if capacity scenarios. For example, if the toolset is fixed, DES could be used to show the resulting output volumes for a variety of product mix scenarios. These data sets could be expanded to include what-ifs for product test times. ATM managers are in need of tools to help them assess capacity in the face of uncertain and ever-changing demand. Although this particular use of DES is not fully explored in this thesis, it would be a logical progression for future work with DES in ATM.

CHAPTER 6. Conclusions and Recommendations

6.1. Advantages of Discrete Event Simulation

Discrete event simulations have the inherent ability to consider dynamic events within the factory. Although they are more complicated to build and maintain than static models, their added value is apparent. Static models are unable to account for any type of variation; this leads to their inability to predict TPTs and WIP distribution in a factory. This is underscored by the graph shown in Figure 3.4. The static capacity models predicted epoxy as the near constraint when in fact there were several toolsets that constrained the factory more than epoxy. The simple reason for this is that the static models cannot account for variation in tool availability or dynamic WIP flows.

DES offers tactical and strategic advantages. On the tactical front, managers can quickly assess the impact of new product loadings, proposed process improvements, test time variation, and WIP policy modifications. As an example of tactical uses of DES, this thesis optimized CONWIP limits and lead to a conservative 20% reduction in TPT simply by reducing the amount of WIP in the factory.

6.2. The Importance of Data Quality and Proper Indicators

While the simulation was successful in demonstrating the potential uses and advantages of DES, the model could not replicate historical factory performance with a great deal of accuracy. The inaccurate tool performance data provided by the factories necessitated the use of a 'Wait for Operator State' to account for the poor tool data and potential delays in loading WIP caused by operators. These adjustment factors were quite large (>20% of a tool's available time in most cases) and raised the question of whether staffing levels and/or operator efficiency are too low. These large operator states question the fundamental assumption that labor effects can be ignored when modeling ATM factories.

Without improvement in factory data quality, significant improvement in ATM factory performance and capacity planning systems will be severely limited. The measurement of four tool performance parameters is need to assess and improve performance:

- Tool utilization
- Tool idle time caused by lack of WIP
- Tool idle time caused by lack of an operator
- Unavailable tool time

These four parameters will allow factories to accurately asses tool and labor performances. Some type of automated data collection system is needed to measure these parameters. The organizational difficulties of installing, using, and learning from such a system are immense and require a coordinated effort across several organizations. In order for the system to succeed, the scope needs to be narrowly defined and effectively managed. Pilot implementation of the system on 3-4 of the most expensive toolsets in the factory would help limit the scope as well as provide the greatest return on investment. In fact, since test is the most expensive process in the factory and the most operationally complex, it would be advisable to gather accurate data on this toolset first and work to improve performance through WIP policy improvement and setup/conversion time reduction. Basic issues such as SFGI and PBIC lot sizing and conversion guidelines would likely lead to quick wins for the test area (and the effective use of DES as a factory tool).

6.3. Capacity Planning in the ATM Environment

The current 10/15/20 gap policy was shown to be inadequate. If 10/15/20 could actually be implemented, TPTs would be excessively long and factory output would be sacrificed. The fact that most ATM factory toolsets contain only 3-6 tools lead to a large amount of 'tool rounding.' When the static capacity models show the need for 4.3 tools, the factory buys 5. This policy leads to gaps much larger than 10/15/20 for all toolsets (Table 5.5). The original intent of this thesis was to create universal PC values for all toolsets in the factory. Unfortunately, capacity calculations based on universal PC values are subject to the same tool rounding issues.

As a result of tool rounding, the capacity of each factory should be designed using simulation. Use of a blanket capacity policy such as 10/15/20 or universal PC values will lead to sub-optimal capital purchases. When designing a factory, the desired output and TPT should be stated and the factory designed to meet these targets. This is in contrast to the current method of simply stating desired output and the factory TPT being determined by the 10/15/20 gap policy. ATM managers need to be made aware of their ability to influence TPT and subsequently need to understand the implications of TPT on the business.

Guidelines for determining PCs in a factory environment with large cost discrepancies between tools were created. This method allows for factory design with minimized capital spending while still allowing for a great deal of flexibility in managing TPT. The proposed method is as follows:

1. Identify the tools that are the largest contributors to unit cost (2-3 tools). These tools should become the constraint and near-constraints for the factory.

2. Plot weekly availability data for each of the tools identified in step 1 using a x-bar control chart. Identify the 2.0σ and 2.5σ LCLs for each of the charts.

3. $\text{Constraint PC} = \dfrac{(\text{Average Availability - }2.0\sigma\ \text{ LCL value})}{2.0\sigma\ \text{ LCL value}}$

4. $\text{Near Constraint PC} = \dfrac{(\text{Average Availability - }2.5\sigma\ \text{ LCL value})}{2.5\sigma\ \text{ LCL value}}$

5. Using simulation, add capacity as needed at non-constraint operations until output and TPT goals are achieved. This is a highly iterative process. A good starting PC for all non-constraints is the 3σ PC value.

The lack of data quality in ATM makes the implementation of full-factory simulation a difficult task given the large scope of the project. Since the test operation is the most expensive in the factory (by a large margin) and is responsible for most of the variation in factory capacity (due to product health and test times), initial focus should be on this area. After a proper method of gathering the four types of necessary data (listed in the section above) is created, a detailed tester simulation would allow for maximum capital savings and effective demonstration of DES as a valuable factory tool. Once the test area is modeled and optimized, the ROI on proliferation of the project throughout the factory should be examined.

Bibliography

Chance, F. et. al., "Supporting Manufacturing with Simulation: Model Design, Development, and Deployment." Proceedings of the 1996 Winter Simulation Conference, San Diego, CA.

Domaschke, J. et. al., "Effective Implementation of Cycle Time Reduction Strategies for Semiconductor Back-End Manufacturing." Proceedings of the 1998 Winter Simulation Conference, Washington, D.C.

Fine, C. H. Clockspeed. Perseus Books, 1998.

Gershwin, S. B., Manufacturing Systems Engineering. PTR Prentice Hall, 1994.

Goldratt, E. M., The Goal. North River Press, 1992.

Grewal, N. et. al., "Validation Simulation Model Cycle Time at Seagate Technology." Submitted to the 1999 Winter Simulation Conference.

Grewal, N. et. al., "Integrating Targeted Cycle-Time Reduction into the Capital Planning Process." Proceedings of the 1998 Winter Simulation Conference, Washington, D.C.

Hopp, W. J., and Spearman, M. L., Factory Physics: Foundations of Manufacturing Management. McGraw-Hill, 1996.

Kempf, Karl. Conversations and documented presentations. July-Dec, 1999.

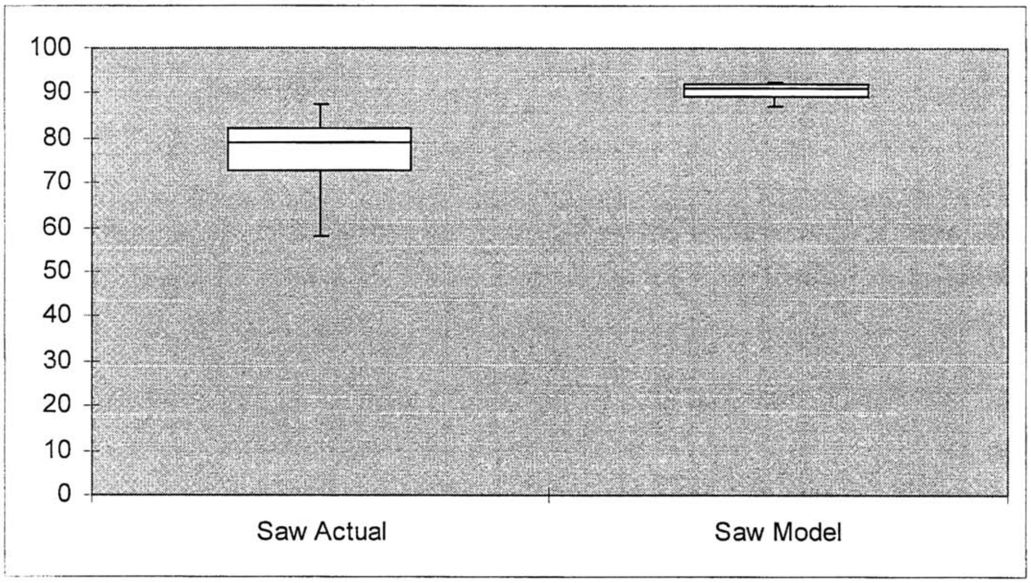
Kempf, Karl and Gray, Kenneth. "Increasing Output and Decreasing Throughput Time through Focused Continuous Improvement." Paper given at the Intel Manufacturing Excellence Conference (IMEC), 1998.

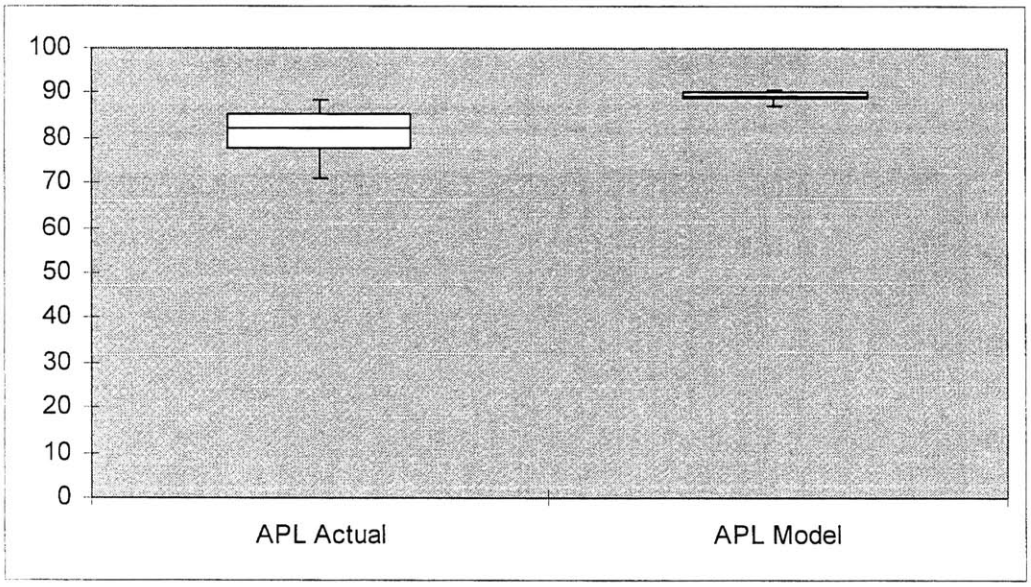
Kotcher, R. and Chance, F., "Capacity Planning in the Face of Product-Mix Uncertainty." 1999 IEEE Symposium on Semiconductor Manufacturing Conference Proceedings, 73-76, Santa Clara, CA.

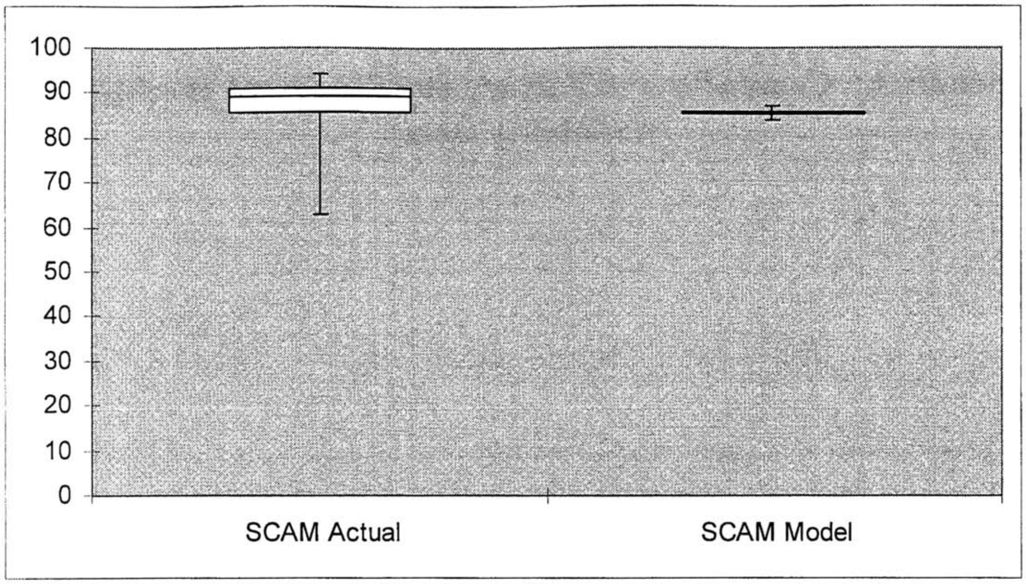
Kurz, M. R. Selection of Operations Management Methodologies in Disparate Cost Environments. Leaders for Manufacturing Thesis (MIT), 1995.

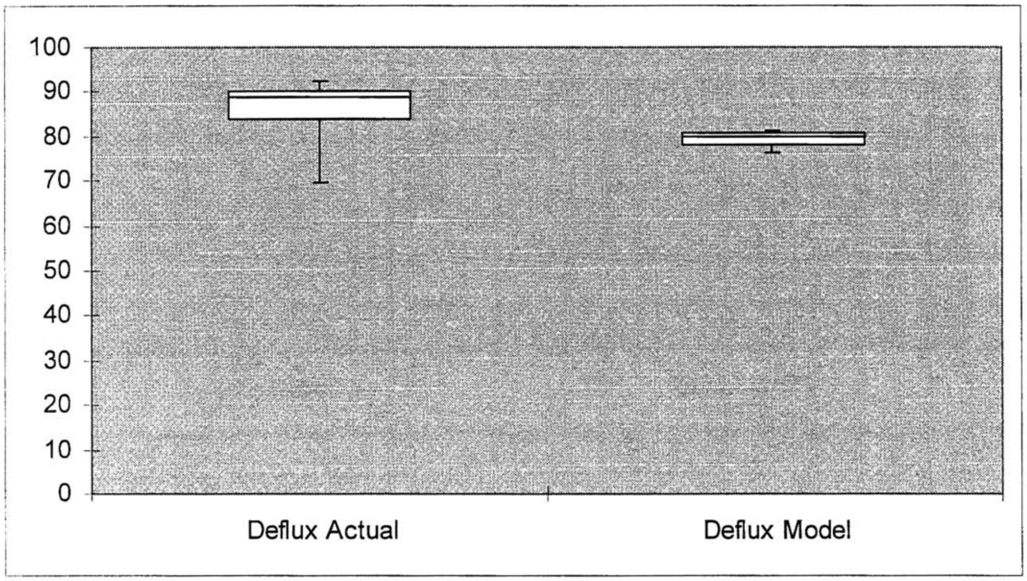
Srivatsan, V.N., Smith, S.P., Villegas, L. Internal Intel document to the IE JET titled "Simulation Results on Constrained vs Balanced Equipment Set Selection." 1995.

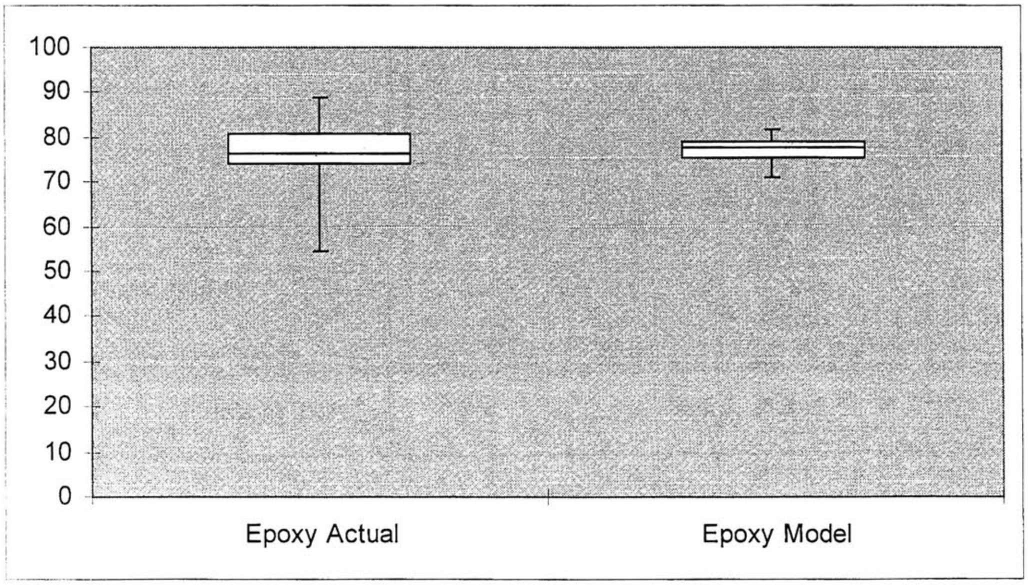
Appendix A: Comparison of Factory and Simulated Weekly Tool Availability Data

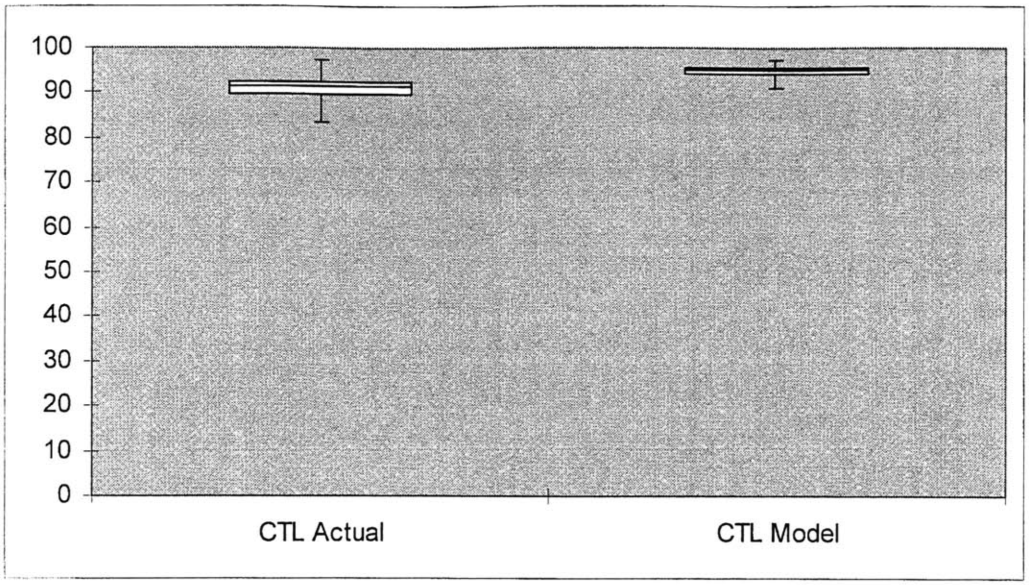


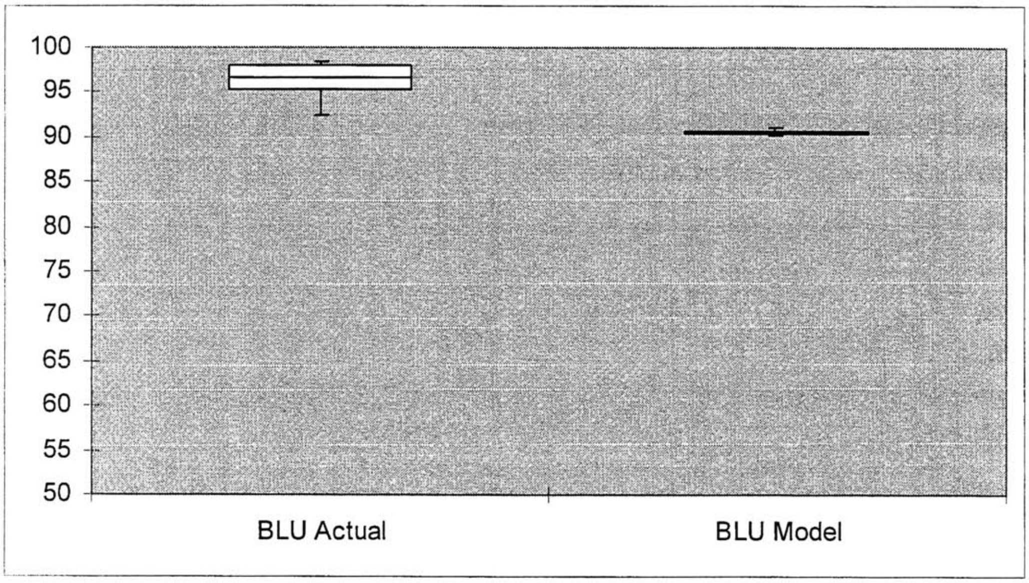


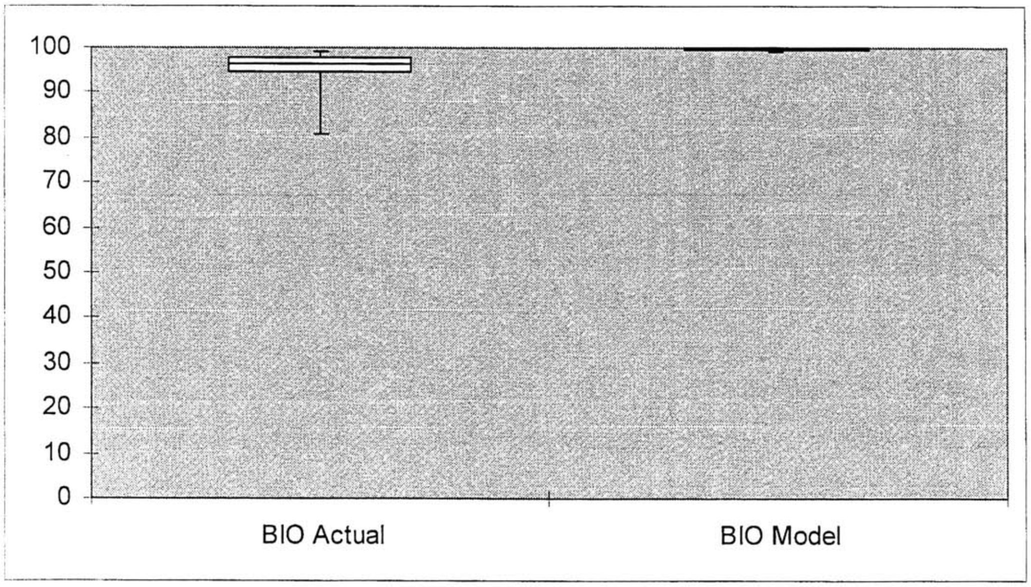

100
90
80
70
60
50
40
30
20
10
0
SCAM Actual
SCAM Model

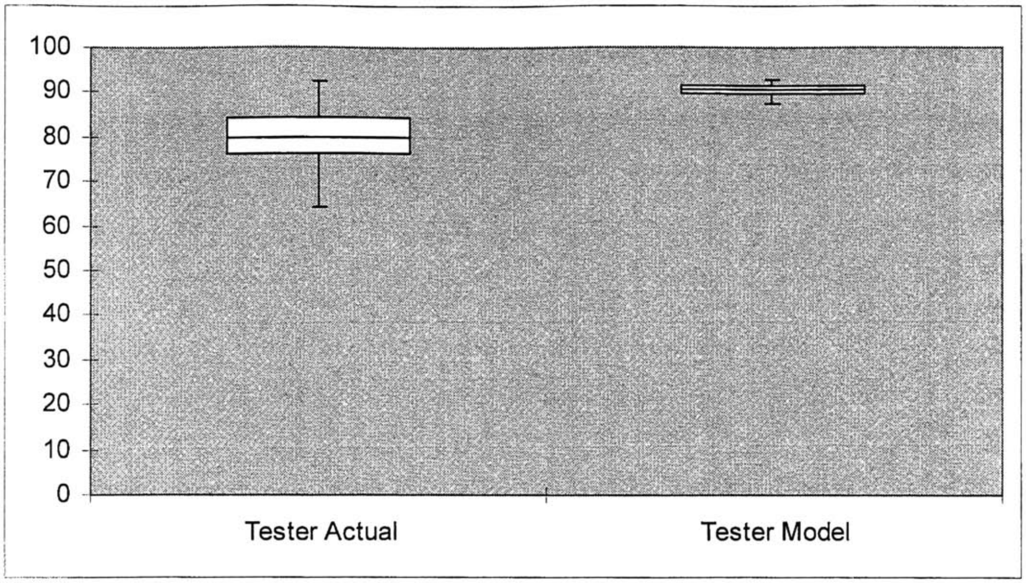

100
90
80
70
60
50
40
30
20
10
0
Deflux Actual
Deflux Model

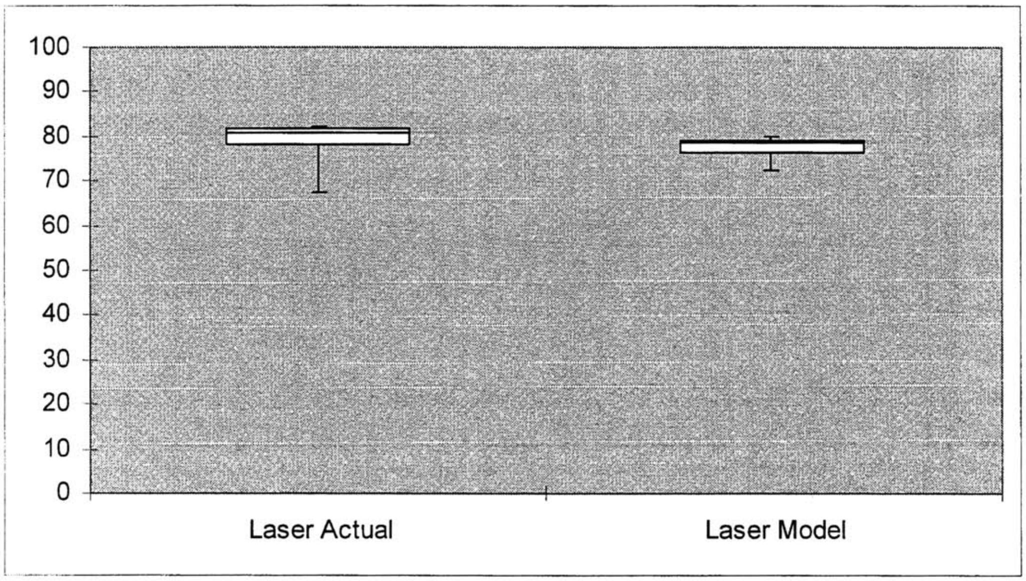

100
90
80
70
60
50
40
30
20
10
0
Epoxy Actual
Epoxy Model

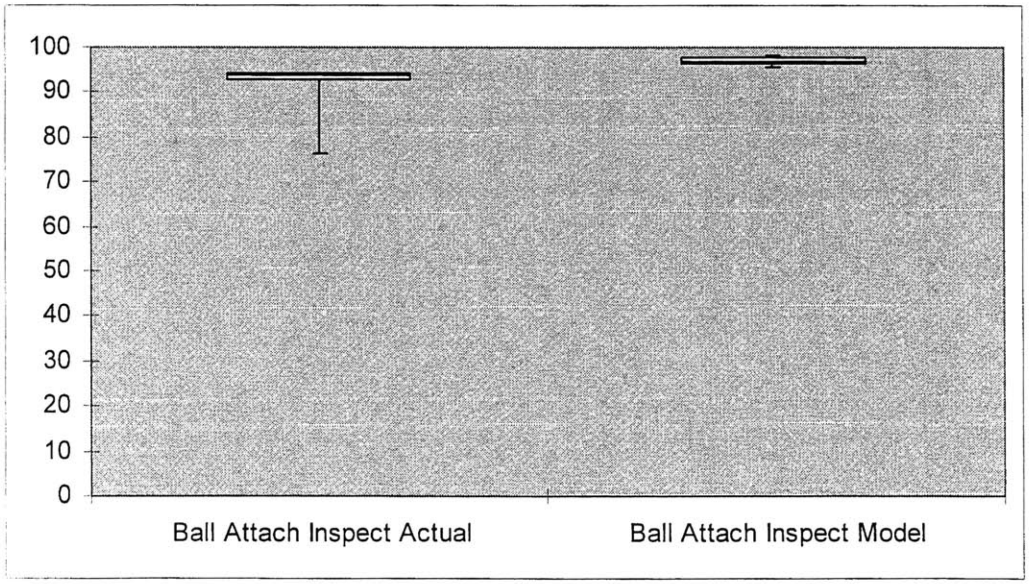

100
90
80
70
60
50
40
30
20
10
0
CTL Actual
CTL Model


100
95
90
85
80
75
70
65
60
55
50
BLU Actual
BLU Model

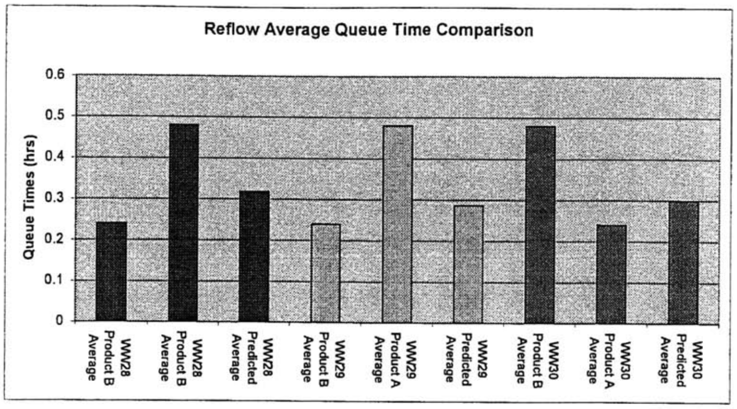

100
90
80
70
60
50
40
30
20
10
0
BIO Actual
BIO Model

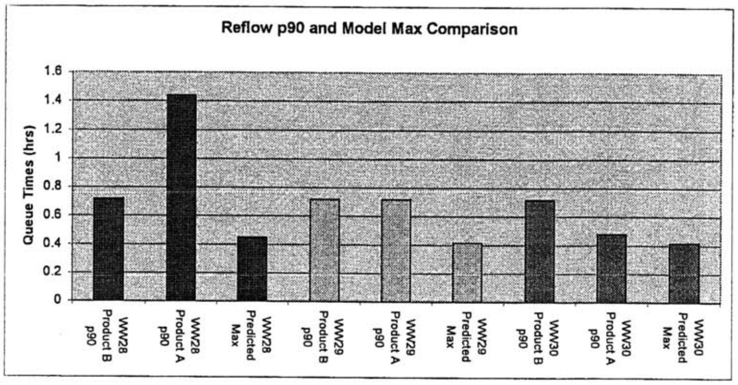

100
90
80
70
60
50
40
30
20
10
0
Tester Actual
Tester Model

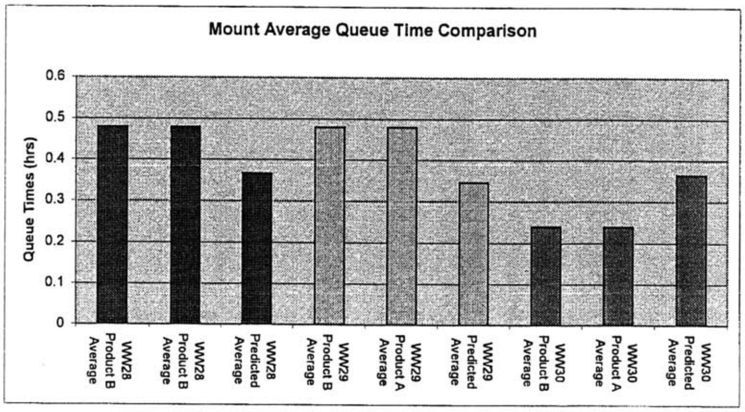

100
90
80
70
60
50
40
30
20
10
0
Laser Actual
Laser Model

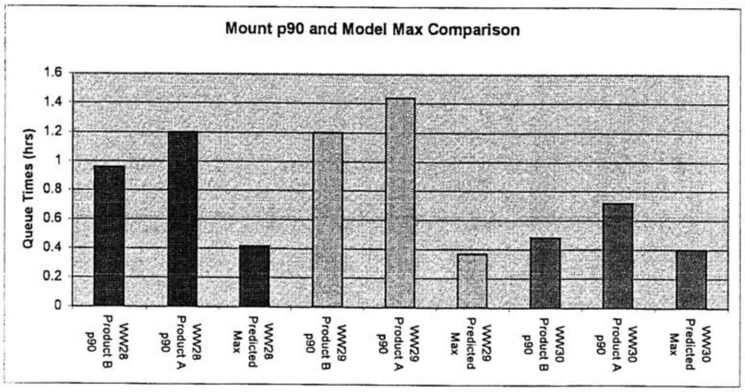

100
90
80
70
60
50
40
30
20
10
0
Ball Attach Inspect Actual
Ball Attach Inspect Model

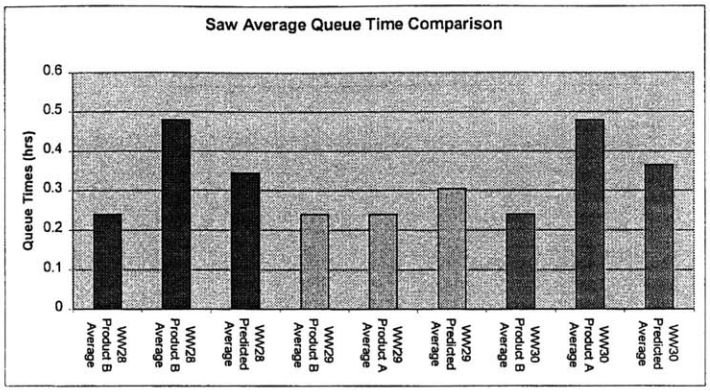
Appendix B: Queue Time Validation Results for WW28-30

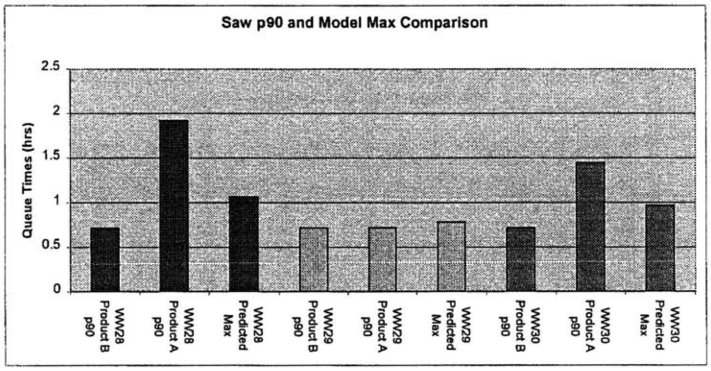


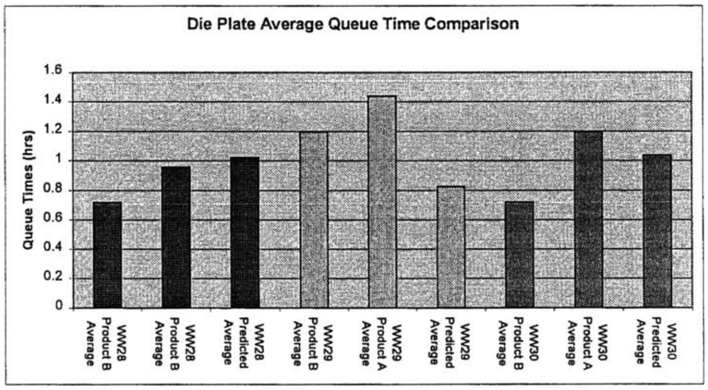


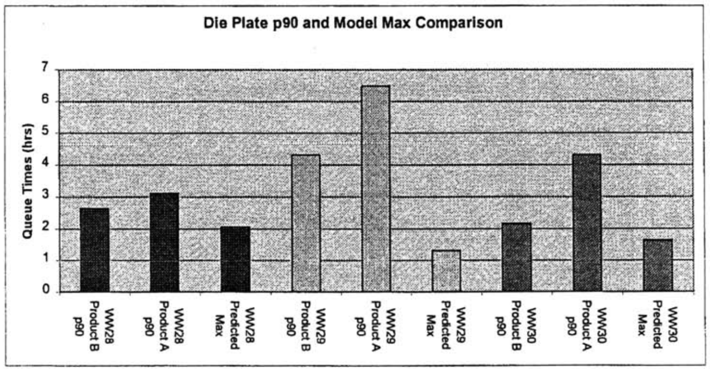


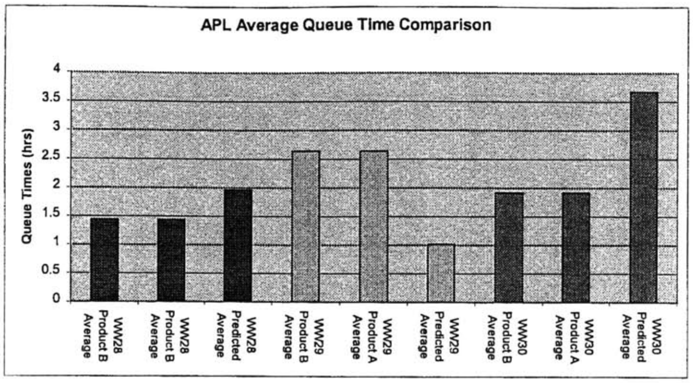


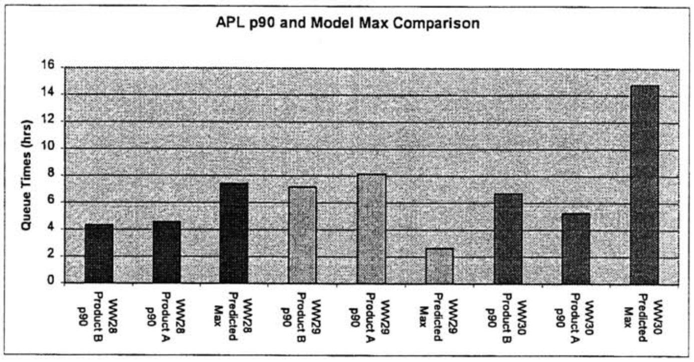

Saw Average Queue Time Comparison
Queue Times (hrs)
0
0.1
0.2
0.3
0.4
0.5
0.6
WW28 Product B Average
WW28 Product B Average
WW28 Predicted Average
WW29 Product B Average
WW29 Product A Average
WW29 Predicted Average
WW30 Product B Average
WW30 Product A Average
WW30 Predicted Average

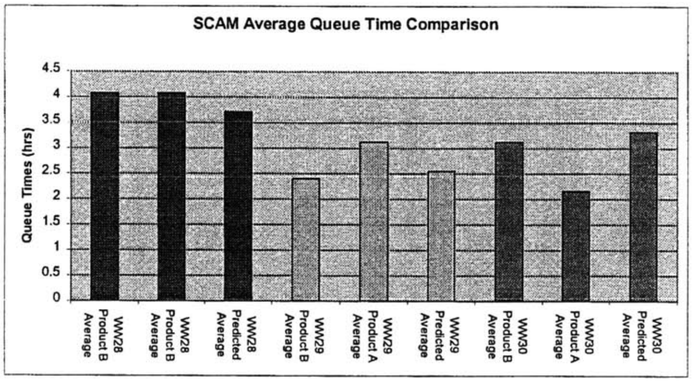

Saw p90 and Model Max Comparison
Queue Times (hrs)
0
0.5
1
1.5
2
2.5
WW28 Product B p90
WW28 Product A p90
WW28 Predicted Max
WW29 Product B p90
WW29 Product A p90
WW29 Predicted Max
WW30 Product B p90
WW30 Product A p90
WW30 Predicted Max

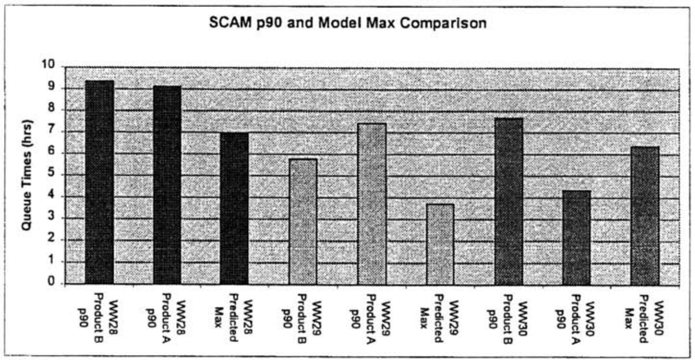

Die Plate Average Queue Time Comparison
Queue Times (hrs)
0
0.2
0.4
0.6
0.8
1
1.2
1.4
1.6
WW28 Product B Average
WW28 Product B Average
WW28 Predicted Average
WW29 Product B Average
WW29 Product A Average
WW29 Predicted Average
WW30 Product B Average
WW30 Product A Average
WW30 Predicted Average

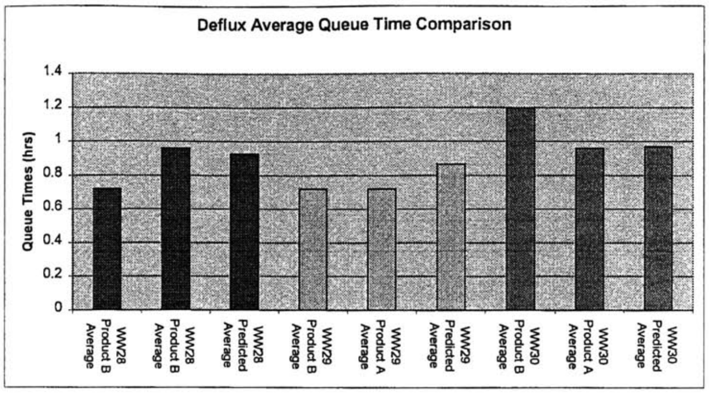

Die Plate p90 and Model Max Comparison
Queue Times (hrs)
0
1
2
3
4
5
6
7
WW28 Product B p90
WW28 Product A p90
WW28 Predicted Max
WW29 Product B p90
WW29 Product A p90
WW29 Predicted Max
WW30 Product B p90
WW30 Product A p90
WW30 Predicted Max

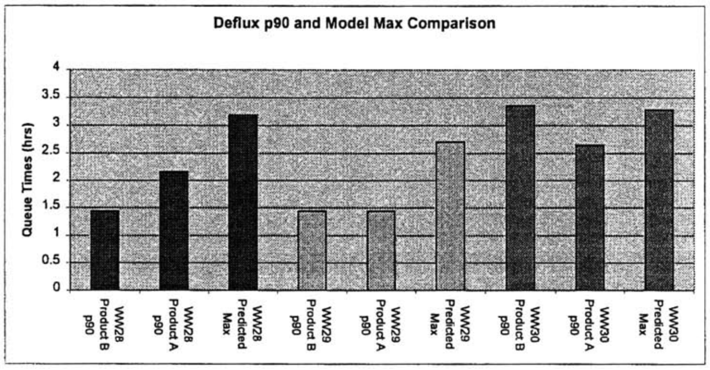

APL Average Queue Time Comparison
Queue Times (hrs)
4
3.5
3
2.5
2
1.5
1
0.5
0
WW28 Product B Average
WW28 Product B Average
WW28 Predicted Average
WW29 Product B Average
WW29 Product A Average
WW29 Predicted Average
WW30 Product B Average
WW30 Product A Average
WW30 Predicted Average

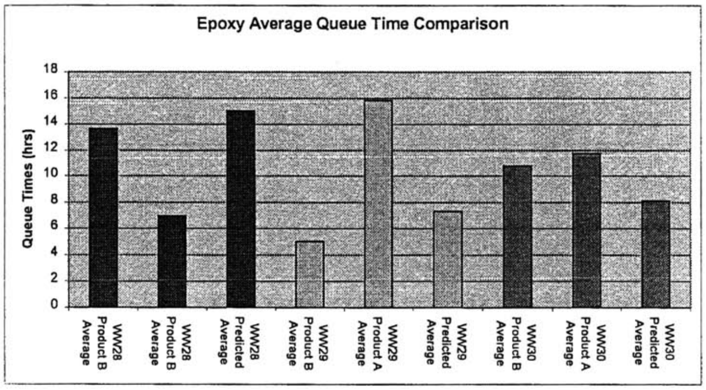

APL p90 and Model Max Comparison
Queue Times (hrs)
16
14
12
10
8
6
4
2
0
WW28 Product B p90
WW28 Product A p90
WW28 Predicted Max
WW29 Product B p90
WW29 Product A p90
WW29 Predicted Max
WW30 Product B p90
WW30 Product A p90
WW30 Predicted Max

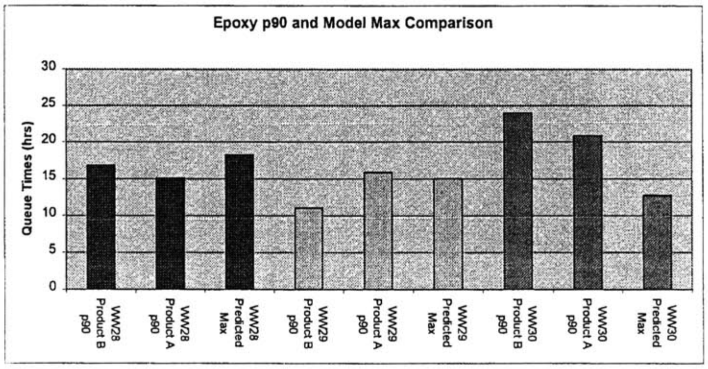

SCAM Average Queue Time Comparison
Queue Times (hrs)
4.5
4
3.5
3
2.5
2
1.5
1
0.5
0
WW28 Product B Average
WW28 Product B Average
WW28 Predicted Average
WW29 Product B Average
WW29 Product A Average
WW29 Predicted Average
WW30 Product B Average
WW30 Product A Average
WW30 Predicted Average

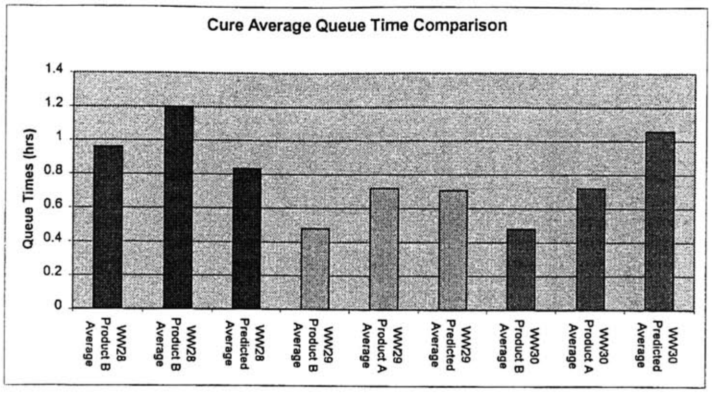

SCAM p90 and Model Max Comparison
Queue Times (hrs)
10
9
8
7
6
5
4
3
2
1
0
WW28 Product B p90
WW28 Product A p90
WW28 Predicted Max
WW29 Product B p90
WW29 Product A p90
WW29 Predicted Max
WW30 Product B p90
WW30 Product A p90
WW30 Predicted Max

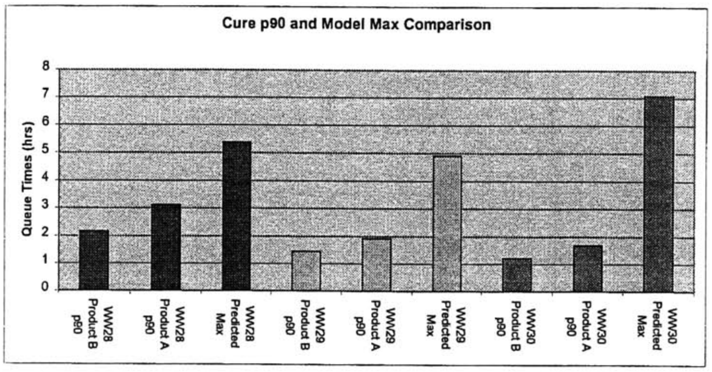

Deflux Average Queue Time Comparison
Queue Times (hrs)
1.4
1.2
1
0.8
0.6
0.4
0.2
0
WW28 Product B Average
WW28 Product B Average
WW28 Predicted Average
WW29 Product B Average
WW29 Product A Average
WW29 Predicted Average
WW30 Product B Average
WW30 Product A Average
WW30 Predicted Average

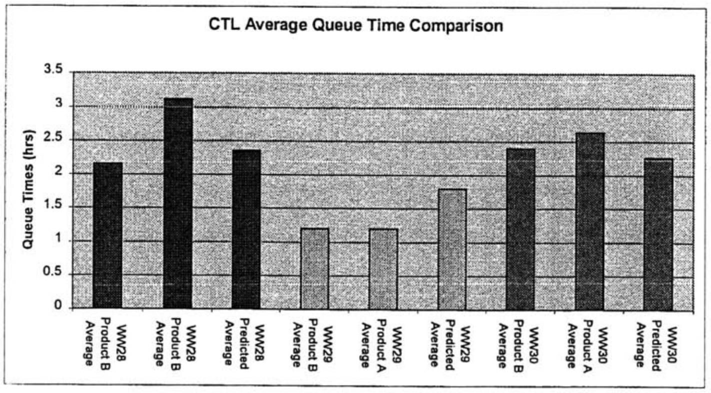

Deflux p90 and Model Max Comparison
Queue Times (hrs)
4
3.5
3
2.5
2
1.5
1
0.5
0
WW28 Product B p90
WW28 Product A p90
WW28 Predicted Max
WW29 Product B p90
WW29 Product A p90
WW29 Predicted Max
WW30 Product B p90
WW30 Product A p90
WW30 Predicted Max

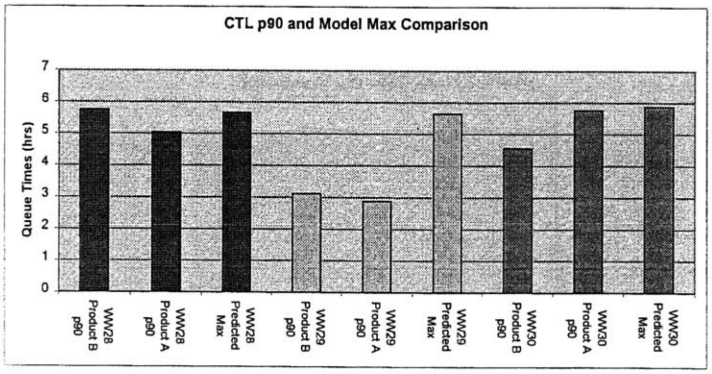

Epoxy Average Queue Time Comparison
Queue Times (hrs)
18
16
14
12
10
8
6
4
2
0
WW28 Product B Average
WW28 Product B Average
WW28 Predicted Average
WW29 Product B Average
WW29 Product A Average
WW29 Predicted Average
WW30 Product B Average
WW30 Product A Average
WW30 Predicted Average

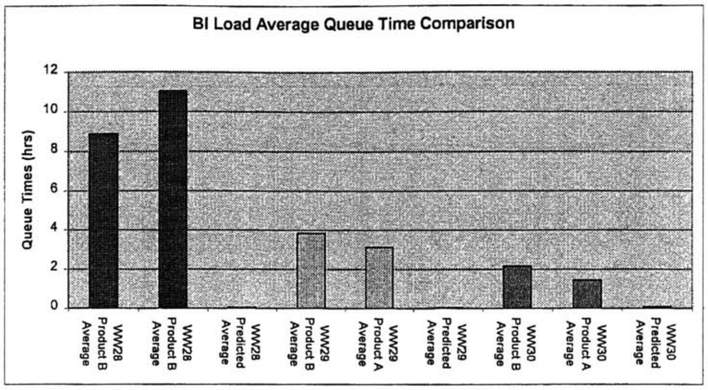

Epoxy p90 and Model Max Comparison
Queue Times (hrs)
30
25
20
15
10
5
0
WW28 Product B p90
WW28 Product A p90
WW28 Predicted Max
WW29 Product B p90
WW29 Product A p90
WW29 Predicted Max
WW30 Product B p90
WW30 Product A p90
WW30 Predicted Max

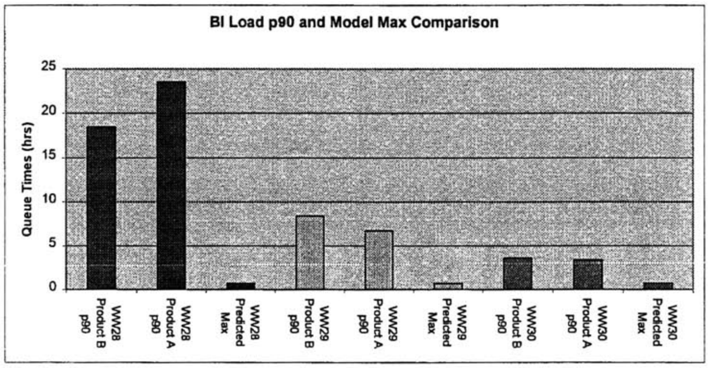

Cure Average Queue Time Comparison
Queue Times (hrs)
1.4
1.2
1
0.8
0.6
0.4
0.2
0
WW28 Product B Average
WW28 Product B Average
WW28 Predicted Average
WW29 Product B Average
WW29 Product A Average
WW29 Predicted Average
WW30 Product B Average
WW30 Product A Average
WW30 Predicted Average

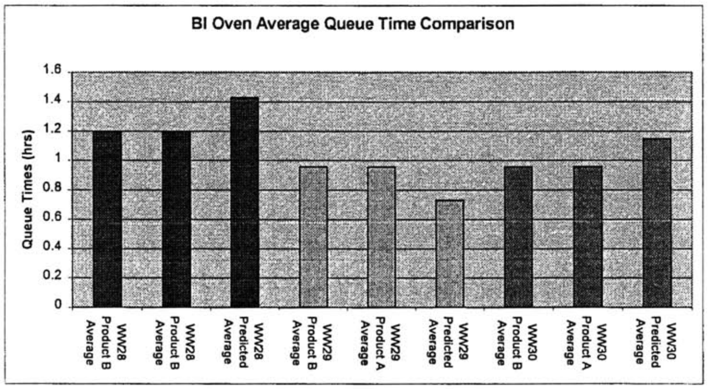

Cure p90 and Model Max Comparison
Queue Times (hrs)
8
7
6
5
4
3
2
1
0
WW28 Product B p90
WW28 Product A p90
WW28 Predicted Max
WW29 Product B p90
WW29 Product A p90
WW29 Predicted Max
WW30 Product B p90
WW30 Product A p90
WW30 Predicted Max

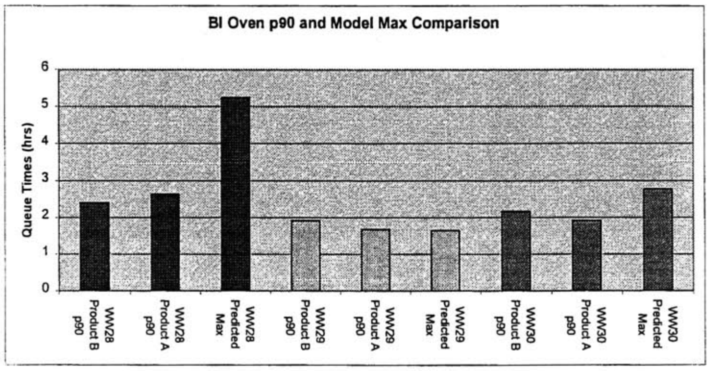

CTL Average Queue Time Comparison
Queue Times (hrs)
3.5
3
2.5
2
1.5
1
0.5
0
WW28 Product B Average
WW28 Product B Average
WW28 Predicted Average
WW29 Product B Average
WW29 Product A Average
WW29 Predicted Average
WW30 Product B Average
WW30 Product A Average
WW30 Predicted Average

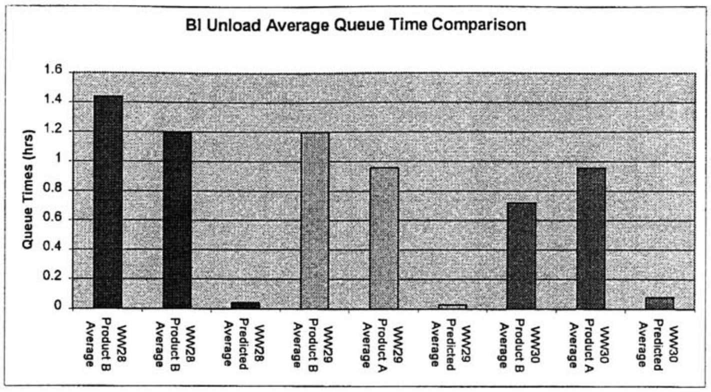

CTL p90 and Model Max Comparison
Queue Times (hrs)
7
6
5
4
3
2
1
0
WW28 Product B p90
WW28 Product A p90
WW28 Predicted Max
WW29 Product B p90
WW29 Product A p90
WW29 Predicted Max
WW30 Product B p90
WW30 Product A p90
WW30 Predicted Max

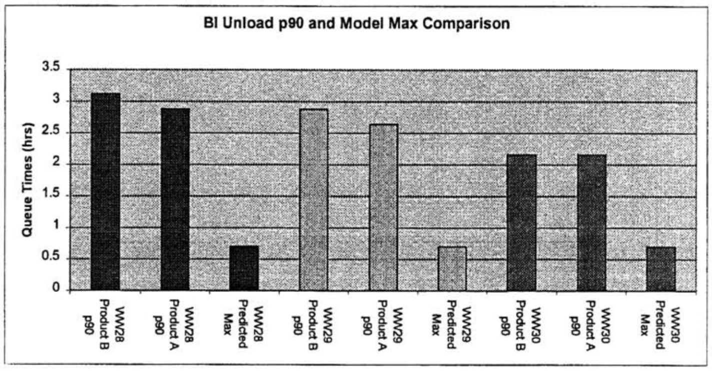

BI Load Average Queue Time Comparison
Queue Times (hrs)
12
10
8
6
4
2
0
WW28 Product B Average
WW28 Product B Average
WW28 Predicted Average
WW29 Product B Average
WW29 Product A Average
WW29 Predicted Average
WW30 Product B Average
WW30 Product A Average
WW30 Predicted Average

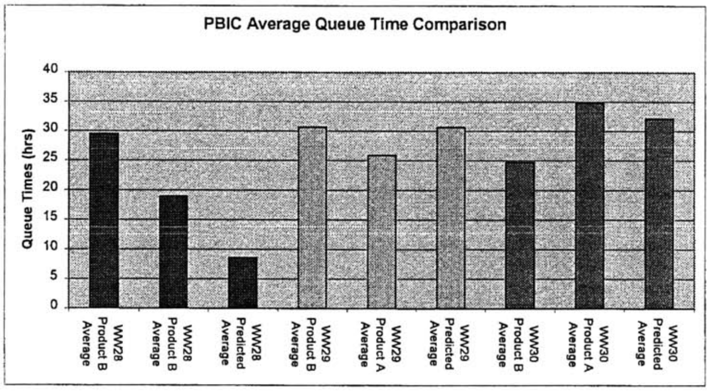

BI Load p90 and Model Max Comparison
Queue Times (hrs)
25
20
15
10
5
0
WW28 Product B p90
WW28 Product A p90
WW28 Predicted Max
WW29 Product B p90
WW29 Product A p90
WW29 Predicted Max
WW30 Product B p90
WW30 Product A p90
WW30 Predicted Max

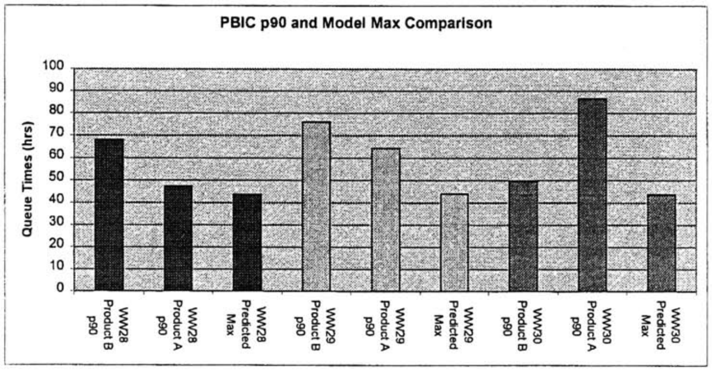

BI Oven Average Queue Time Comparison
Queue Times (hrs)
1.6
1.4
1.2
1
0.8
0.6
0.4
0.2
0
WW28 Product B Average
WW28 Product B Average
WW28 Predicted Average
WW29 Product B Average
WW29 Product A Average
WW29 Predicted Average
WW30 Product B Average
WW30 Product A Average
WW30 Predicted Average

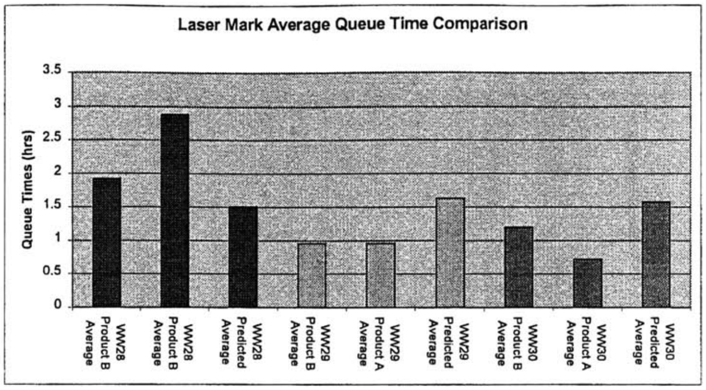

BI Oven p90 and Model Max Comparison
Queue Times (hrs)
6
5
4
3
2
1
0
WW28 Product B p90
WW28 Product A p90
WW28 Predicted Max
WW29 Product B p90
WW29 Product A p90
WW29 Predicted Max
WW30 Product B p90
WW30 Product A p90
WW30 Predicted Max

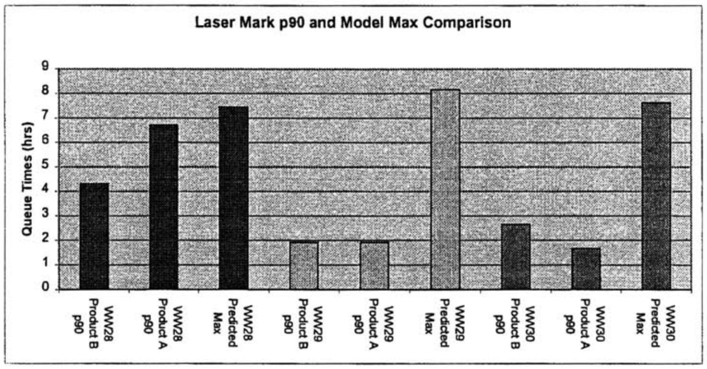

BI Unload Average Queue Time Comparison

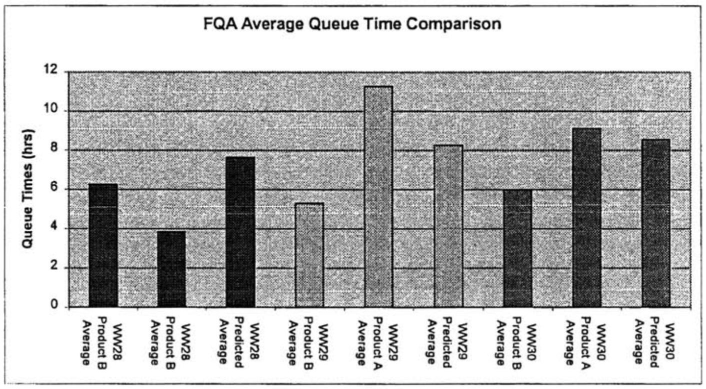

BI Unload p90 and Model Max Comparison

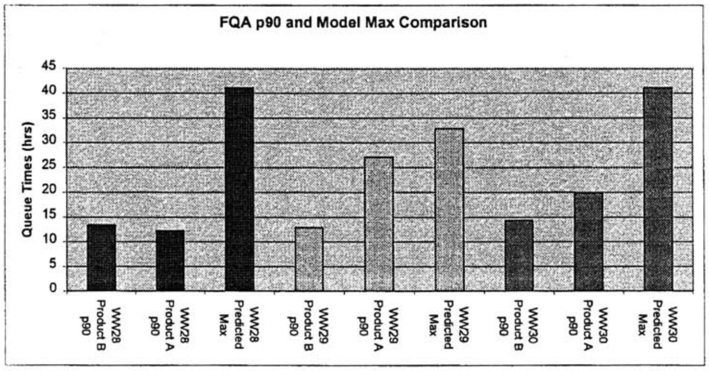

PBIC Average Queue Time Comparison

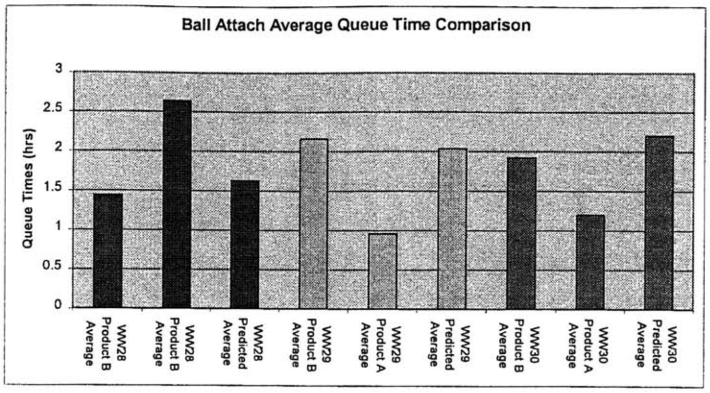

PBIC p90 and Model Max Comparison


Laser Mark Average Queue Time Comparison

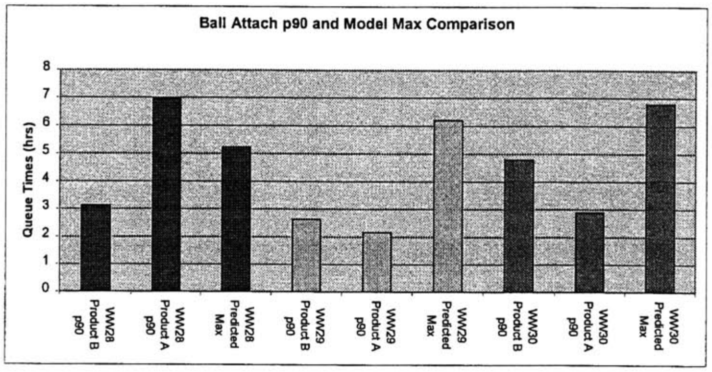

Laser Mark p90 and Model Max Comparison


FQA Average Queue Time Comparison

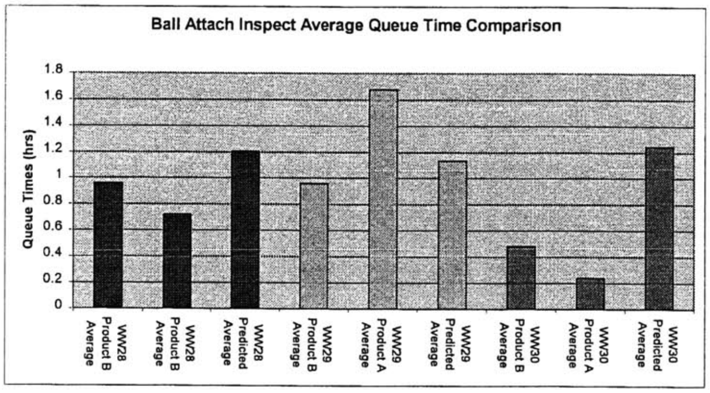

FQA p90 and Model Max Comparison



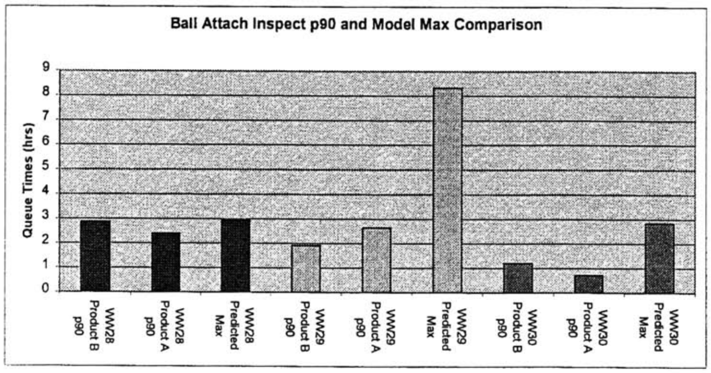
Ball Attach Average Queue Time Comparison



Ball Attach p90 and Model Max Comparison

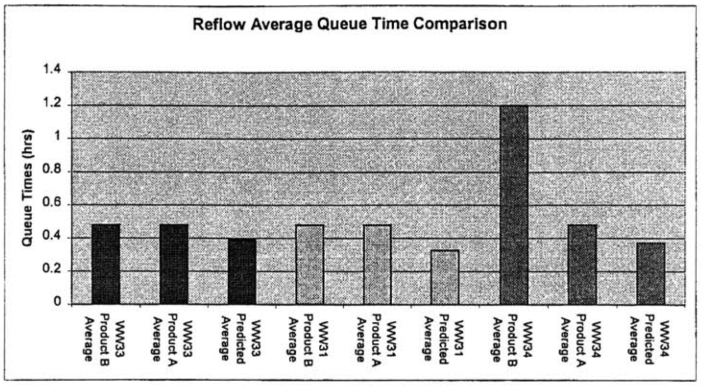


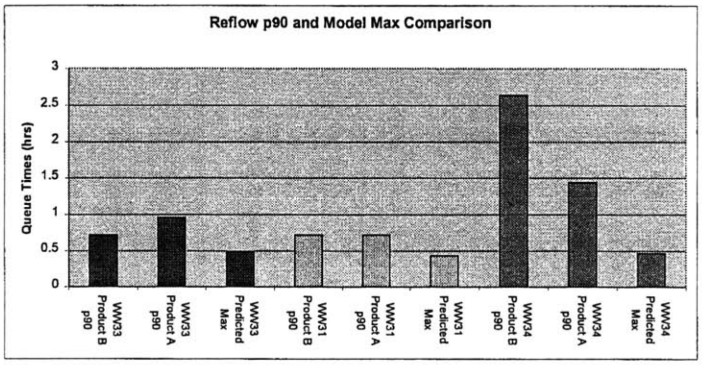
Ball Attach Inspect Average Queue Time Comparison

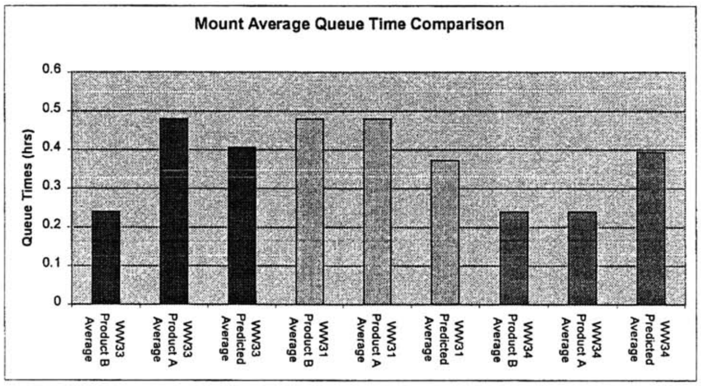


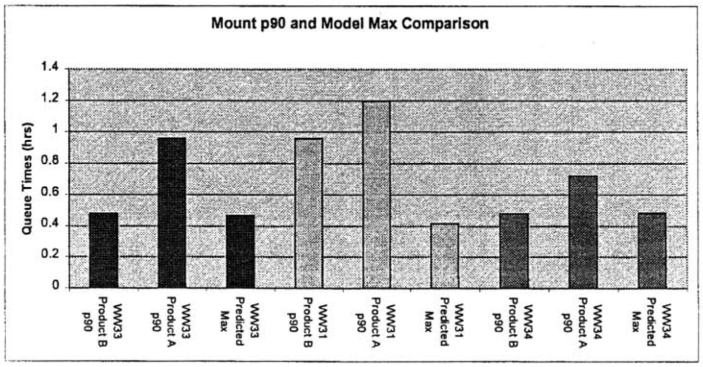
Ball Attach Inspect p90 and Model Max Comparison

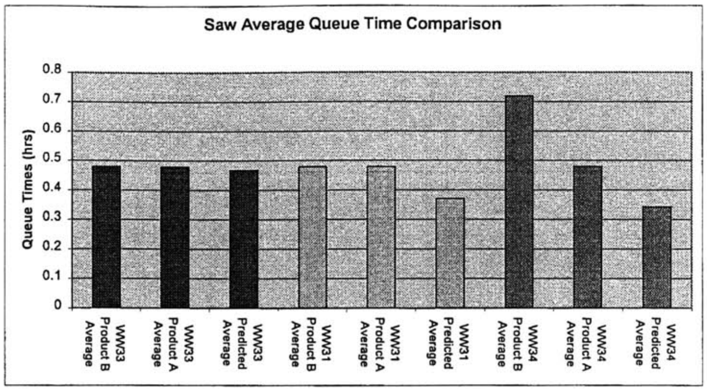
Appendix C: Queue Time Validation Results for WW31, 33-34

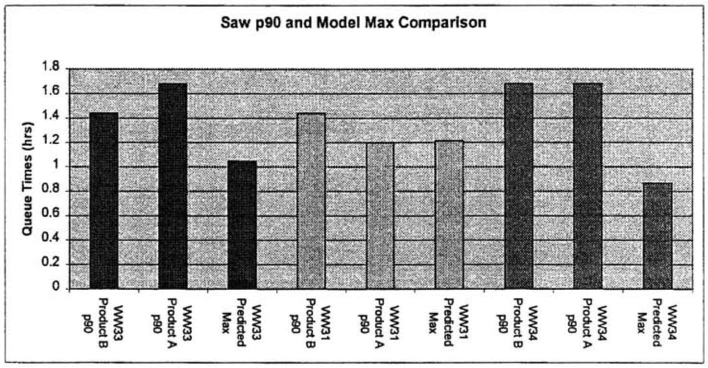


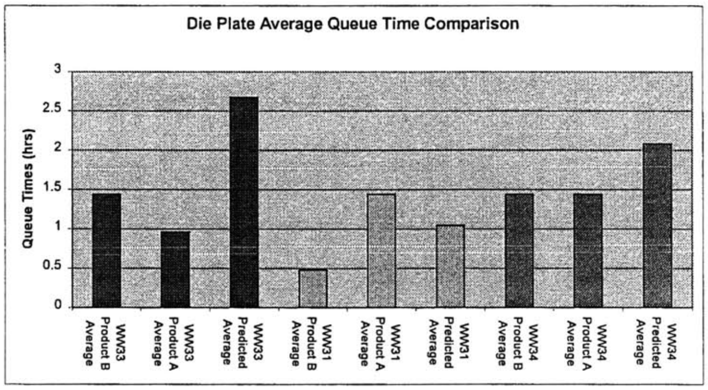


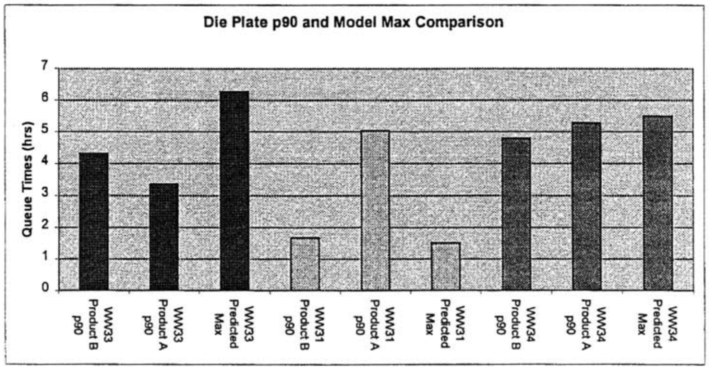


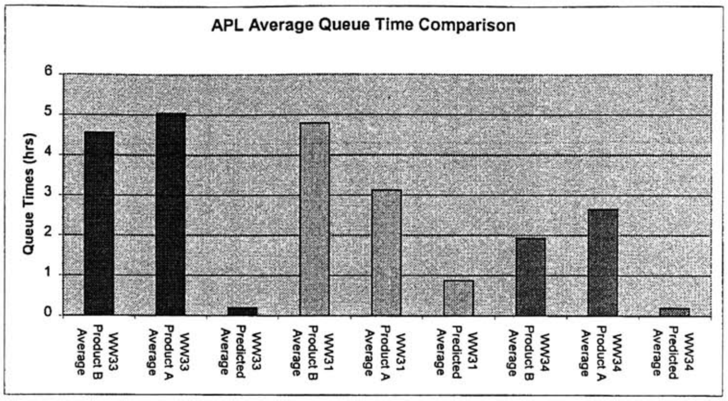


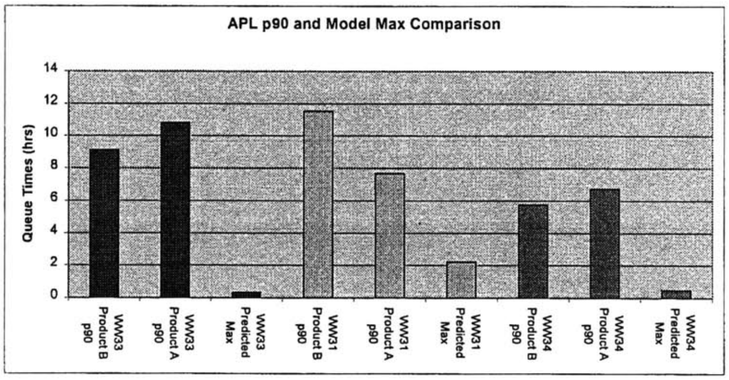

Saw Average Queue Time Comparison

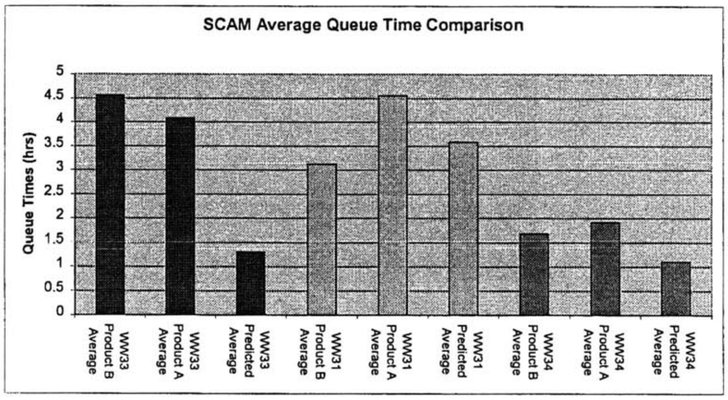

Saw p90 and Model Max Comparison

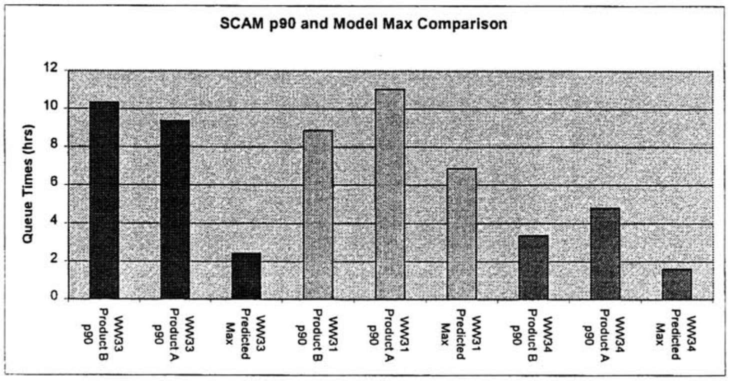

Die Plate Average Queue Time Comparison

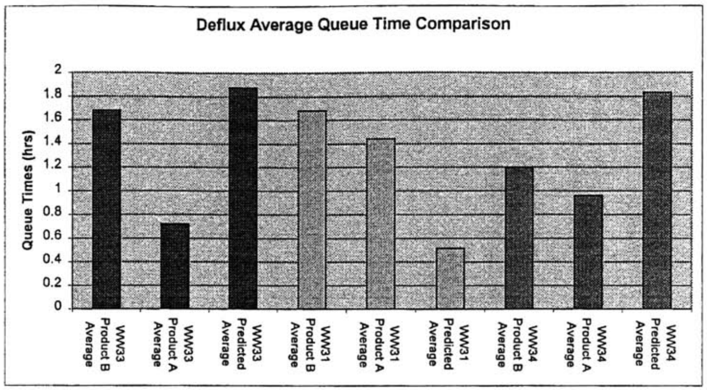

Die Plate p90 and Model Max Comparison

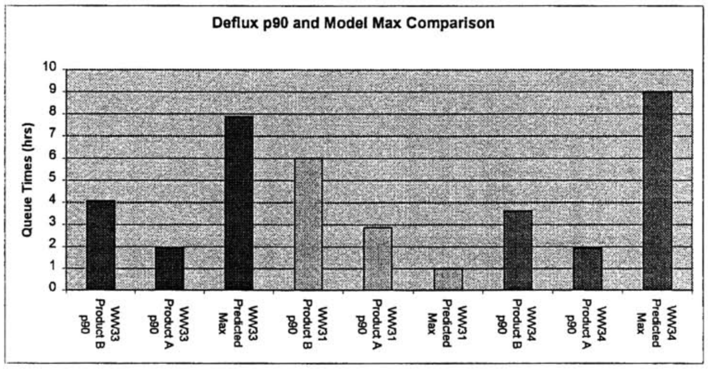

APL Average Queue Time Comparison
Queue Times (hrs)
6
5
4
3
2
1
0
WW33 Product B Average
WW33 Product A Average
WW33 Predicted Average
WW31 Product B Average
WW31 Product A Average
WW31 Predicted Average
WW34 Product B Average
WW34 Product A Average
WW34 Predicted Average

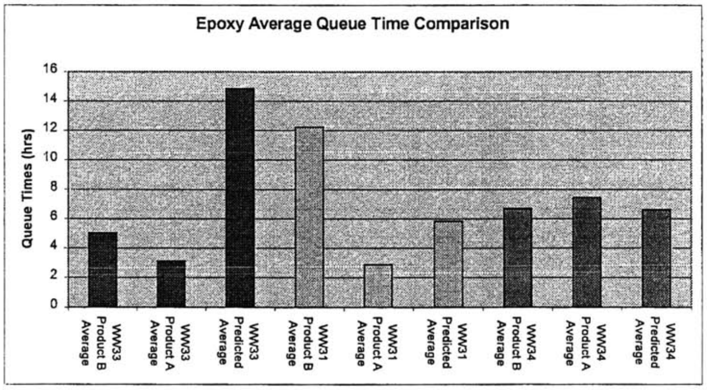

APL p90 and Model Max Comparison
Queue Times (hrs)
14
12
10
8
6
4
2
0
WW33 Product B p90
WW33 Product A p90
WW33 Predicted Max
WW31 Product B p90
WW31 Product A p90
WW31 Predicted Max
WW34 Product B p90
WW34 Product A p90
WW34 Predicted Max

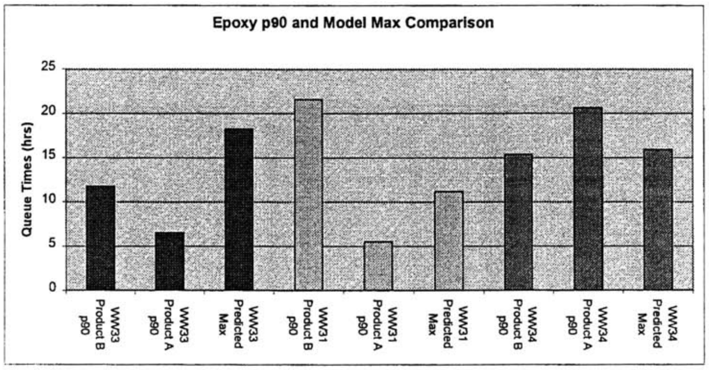

SCAM Average Queue Time Comparison
Queue Times (hrs)
5
4.5
4
3.5
3
2.5
2
1.5
1
0.5
0
WW33 Product B Average
WW33 Product A Average
WW33 Predicted Average
WW31 Product B Average
WW31 Product A Average
WW31 Predicted Average
WW34 Product B Average
WW34 Product A Average
WW34 Predicted Average

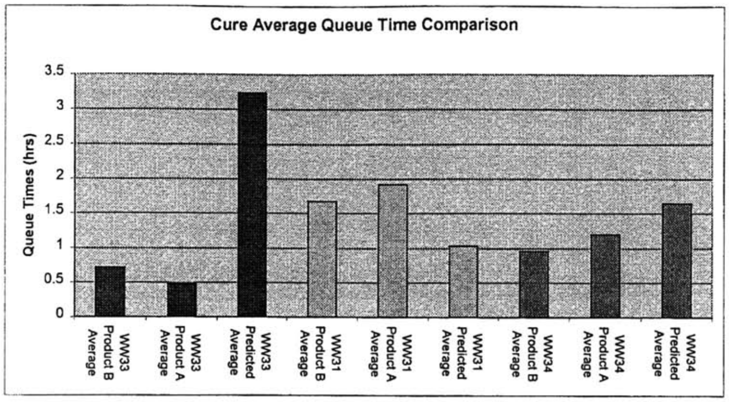

SCAM p90 and Model Max Comparison
Queue Times (hrs)
12
10
8
6
4
2
0
WW33 Product B p90
WW33 Product A p90
WW33 Predicted Max
WW31 Product B p90
WW31 Product A p90
WW31 Predicted Max
WW34 Product B p90
WW34 Product A p90
WW34 Predicted Max


Deflux Average Queue Time Comparison

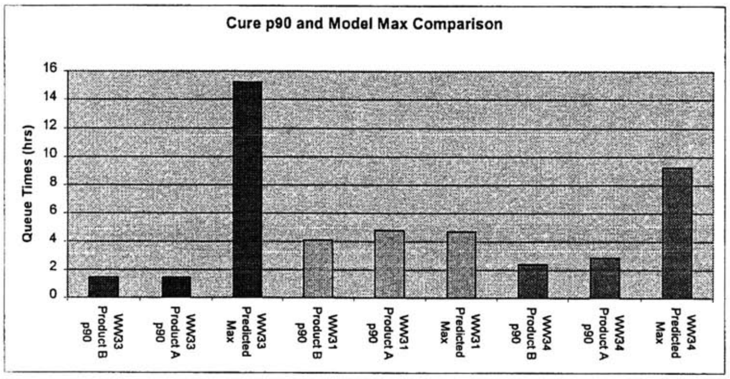

Deflux p90 and Model Max Comparison


Epoxy Average Queue Time Comparison

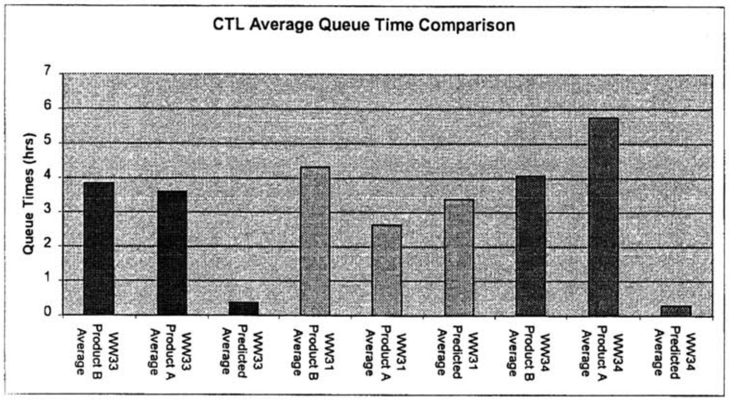

Epoxy p90 and Model Max Comparison



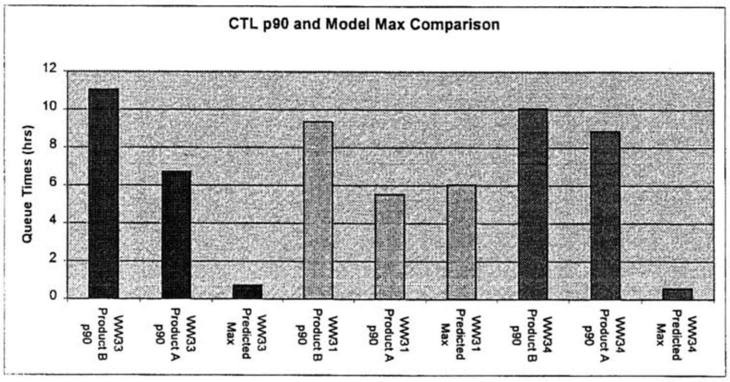
Cure Average Queue Time Comparison



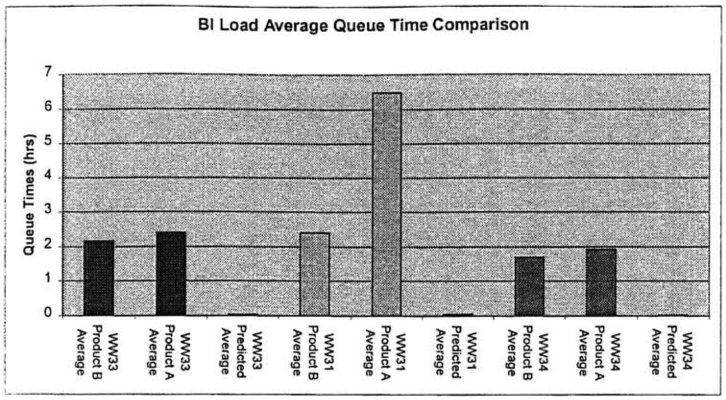
Cure p90 and Model Max Comparison

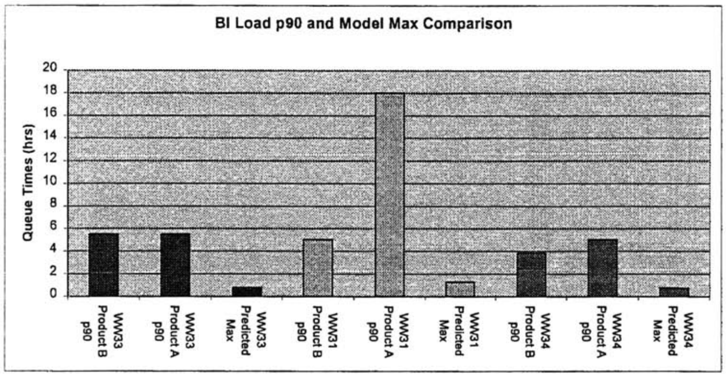


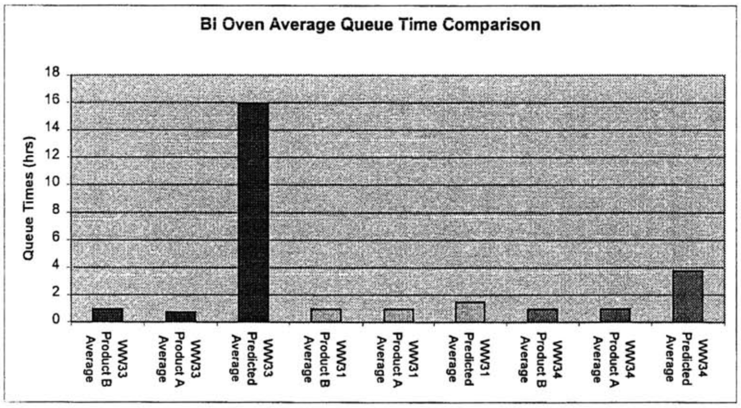
CTL Average Queue Time Comparison

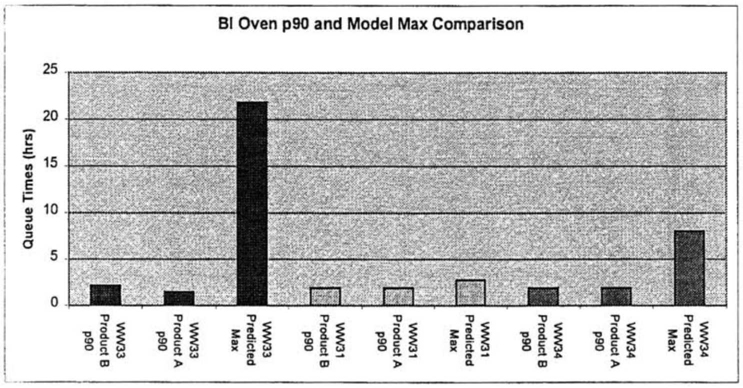


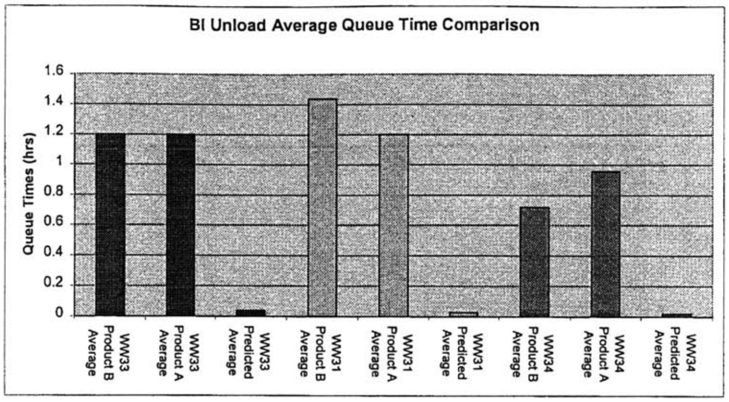
CTL p90 and Model Max Comparison


BI Load Average Queue Time Comparison
Queue Times (hrs)
0
1
2
3
4
5
6
7
WW33 Product B Average
WW33 Product A Average
WW33 Predicted Average
WW31 Product B Average
WW31 Product A Average
WW31 Predicted Average
WW34 Product B Average
WW34 Product A Average
WW34 Predicted Average

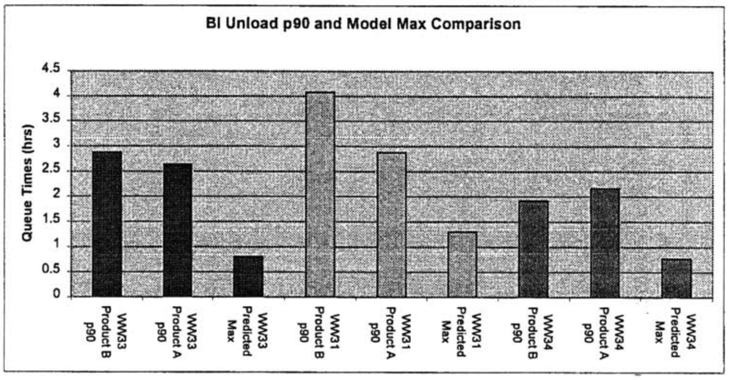

BI Load p90 and Model Max Comparison
Queue Times (hrs)
0
2
4
6
8
10
12
14
16
18
20
WW33 Product B p90
WW33 Product A p90
WW33 Predicted Max
WW31 Product B p90
WW31 Product A p90
WW31 Predicted Max
WW34 Product B p90
WW34 Product A p90
WW34 Predicted Max


BI Oven Average Queue Time Comparison
Queue Times (hrs)
0
2
4
6
8
10
12
14
16
18
WW33 Product B Average
WW33 Product A Average
WW33 Predicted Average
WW31 Product B Average
WW31 Product A Average
WW31 Predicted Average
WW34 Product B Average
WW34 Product A Average
WW34 Predicted Average

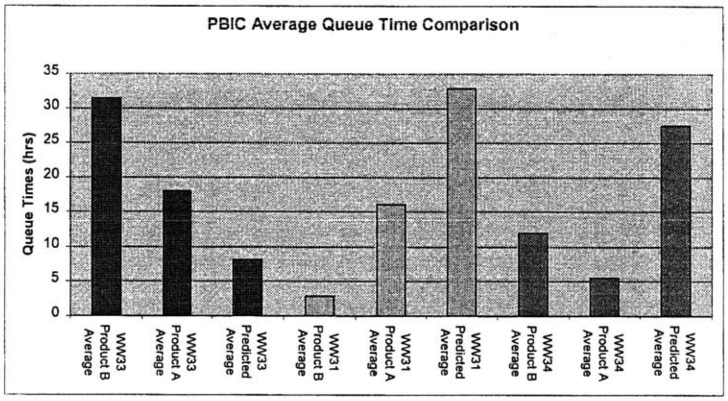

BI Oven p90 and Model Max Comparison
Queue Times (hrs)
0
5
10
15
20
25
WW33 Product B p90
WW33 Product A p90
WW33 Predicted Max
WW31 Product B p90
WW31 Product A p90
WW31 Predicted Max
WW34 Product B p90
WW34 Product A p90
WW34 Predicted Max



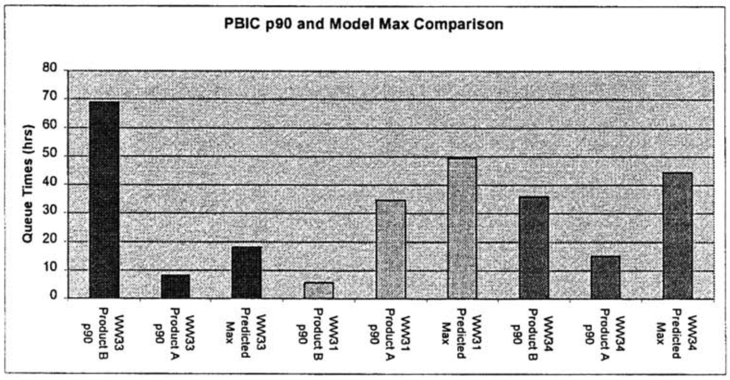
BI Unload Average Queue Time Comparison



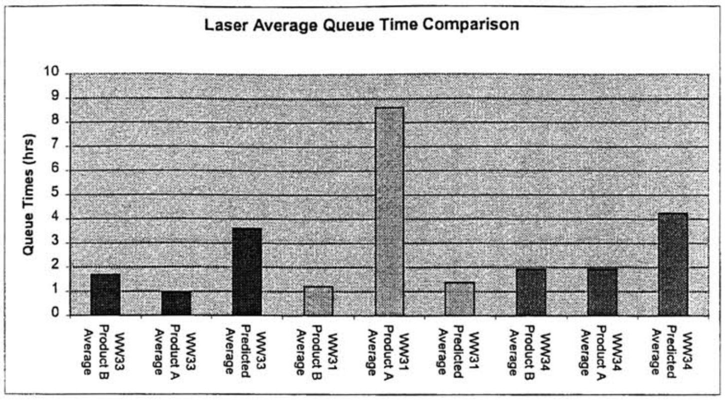
BI Unload p90 and Model Max Comparison

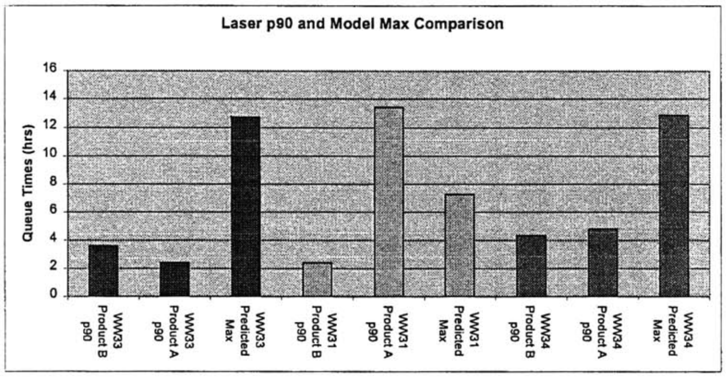


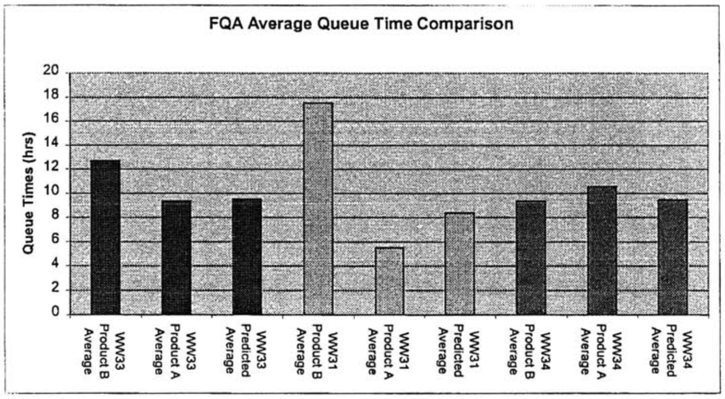
PBIC Average Queue Time Comparison

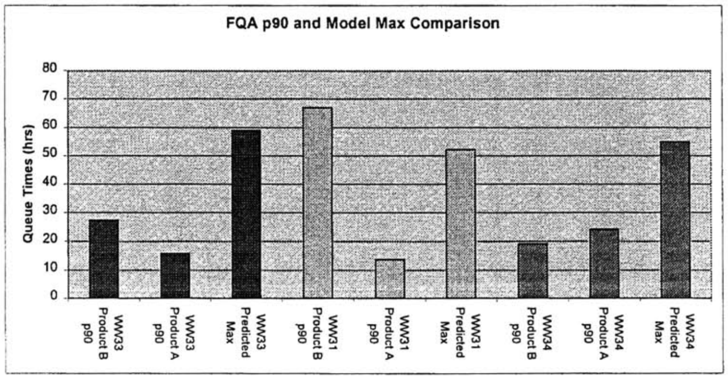


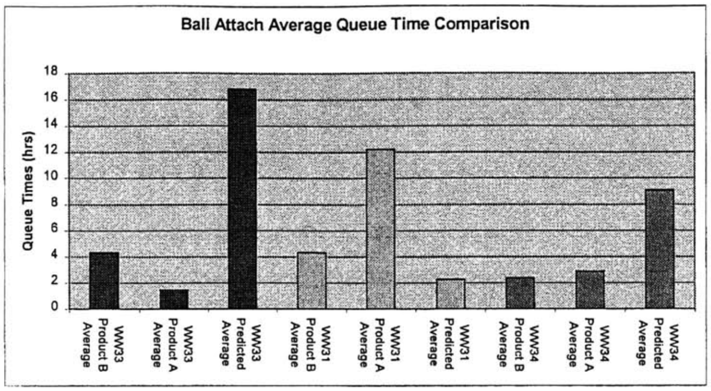
PBIC p90 and Model Max Comparison

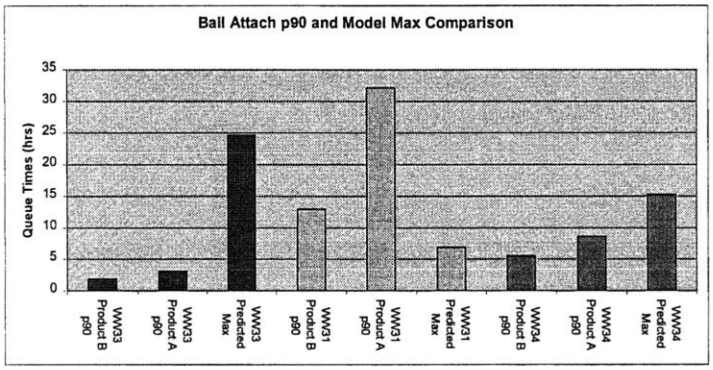

Laser Average Queue Time Comparison

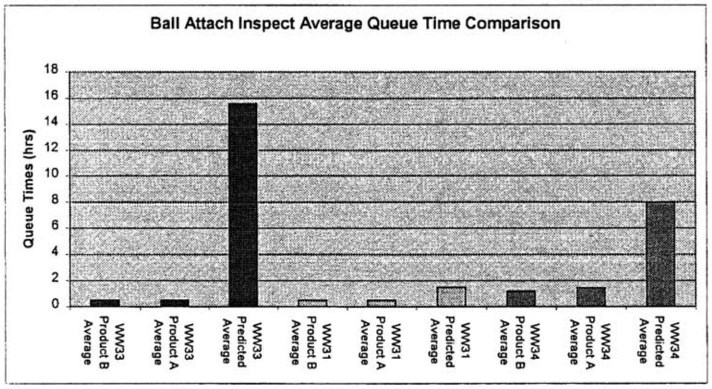

Laser p90 and Model Max Comparison

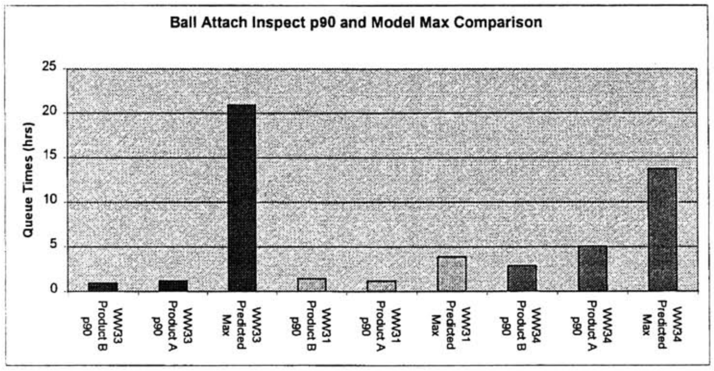

FQA Average Queue Time Comparison


FQA p90 and Model Max Comparison


Ball Attach Average Queue Time Comparison
Queue Times (hrs)
0
2
4
6
8
10
12
14
16
18
WW33 Product B Average
WW33 Product A Average
WW33 Predicted Average
WW31 Product B Average
WW31 Product A Average
WW31 Predicted Average
WW34 Product B Average
WW34 Product A Average
WW34 Predicted Average


Ball Attach p90 and Model Max Comparison
Queue Times (hrs)
0
5
10
15
20
25
30
35
WW33 Product B p90
WW33 Product A p90
WW33 Predicted Max
WW31 Product B p90
WW31 Product A p90
WW31 Predicted Average
WW34 Product B p90
WW34 Product A p90
WW34 Predicted Max


Ball Attach Inspect Average Queue Time Comparison
Queue Times (hrs)
0
2
4
6
8
10
12
14
16
18
WW33 Product B Average
WW33 Product A Average
WW33 Predicted Average
WW31 Product B Average
WW31 Product A Average
WW31 Predicted Average
WW34 Product B Average
WW34 Product A Average
WW34 Predicted Average


Ball Attach Inspect p90 and Model Max Comparison
Queue Times (hrs)
0
5
10
15
20
25
WW33 Product B p90
WW33 Product A p90
WW33 Predicted Max
WW31 Product B p90
WW31 Product A p90
WW31 Predicted Max
WW34 Product B p90
WW34 Product A p90
WW34 Predicted Max

Appendix D: Block Size Optimization Results

WW29 Screen Experiment Results

WW29 Simulation Baseline using Block 1 = Block 2 = 2 days. Note: the TPT adjustment factor of 0.7 days was applied to all TPTs. Note that the output numbers have been disguised.

	2-2-1			
Simulation Week	Product A Output	Product A TPT	Product B Output	Product B TPT
1	316628	4.2	378784	4.2
2	337998	4.3	400150	4.4
3	324398	4.5	380730	4.5
4	361286	4.0	407942	4.0
5	332170	4.3	374898	4.3
6	310788	4.8	361312	4.8
7	334094	4.5	417648	4.5
8	361302	4.0	404040	4.0
Average	334833	5.0	390688	5.0

WW29 Simulation data using Block 1= Block 2 = 1.5 days.

	1.5-1.5-1			
Simulation Week	Product A Output	Product A TPT	Product B Output	Product B TPT
1	324394	3.2	376846	3.3
2	347710	3.4	384610	3.4
3	334118	3.3	388488	3.4
4	338002	3.1	415688	3.2
5	345760	3.2	369078	3.3
6	314670	3.7	361318	3.8
7	324412	3.4	409850	3.5
8	363254	3.1	394314	3.1
Average	336540	3.3	387524	3.4

Results of t-tests between the WW29 output of the 2,2 simulations and the 1.5,1.5 simulations. For both products with α=0.05, $|t| < t_{critical,\ 2\ sided}$, so the null hypothesis that the means of the two samples are equal is not rejected.

1.5-1.5 to 2-2 t-test

Product A Output
t-Test: Two-Sample Assuming Equal Variances

	Variable 1	*Variable 2*
Mean	334832	336540
Variance	87017762.86	60868900.86
Observations	8	8
Pooled Variance	73943331.86	
Hypothesized Mean Difference	0	
df	14	
t Stat	-0.20	
P(T<=t) one-tail	0.42	
t Critical one-tail	1.76	
P(T<=t) two-tail	0.85	
t Critical two-tail	2.14	

Product B Output
t-Test: Two-Sample Assuming Equal Variances

	Variable 1	*Variable 2*
Mean	390688	387524
Variance	94481711	88864821
Observations	8	8
Pooled Variance	91673266	
Hypothesized Mean Difference	0	
df	14	
t Stat	0.33	
P(T<=t) one-tail	0.37	
t Critical one-tail	1.76	
P(T<=t) two-tail	0.75	
t Critical two-tail	2.14	

Results of the t-tests between the TPTs of the 2,2 and 1.5,1.5 simulations. For both products with α=0.05, $|t| > t_{critical,\ 2\ sided}$, so the null hypothesis that the means of the two samples are equal is rejected; the TPT improvement is statistically meaningful.

Product A TPT (hours) t-test
t-Test: Two-Sample Assuming Equal Variances

	Variable 1	*Variable 2*
Mean	103.6	78.9
Variance	41.6	24.0
Observations	8	8
Pooled Variance	32.8	
Hypothesized Mean Difference	0	
df	14	
t Stat	8.63	
P(T<=t) one-tail	2.8E-07	
t Critical one-tail	1.76	
P(T<=t) two-tail	5.59E-07	
t Critical two-tail	2.14	

Product B TPT
(hours)t-Test: Two-Sample Assuming Equal Variances

	Variable 1	*Variable 2*
Mean	103.5	80.3
Variance	34.3	27.3
Observations	8	8
Pooled Variance	30.8	
Hypothesized Mean Difference	0	
df	14	
t Stat	8.33	
P(T<=t) one-tail	4.24E-07	
t Critical one-tail	1.76	
P(T<=t) two-tail	8.48E-07	
t Critical two-tail	2.14	

WW28 Simulation Results

Results of t-tests between the WW28 output of the 2,2 simulations and the 1.5,1.5 simulations. For both products with α=0.05, $|t| < t_{critical, 2\ sided}$, so the null hypothesis that the means of the two samples are equal is not rejected. A t-test to determine the significance of the TPT differences was not performed because the similar experiment for the WW29 screening data demonstrated how robust the TPT difference actually is.

t-test with 2-2-1 and 1.5-1.5-1
t-Test: Two-Sample Assuming Equal Variances
Product A Output

	Variable 1	*Variable 2*
Mean	419276	416290
Variance	69606204.410	125016476.9
Observations	13.000	13
Pooled Variance	97311340.654	
Hypothesized Mean Difference	0.000	
df	24.000	
t Stat	0.39	
P(T<=t) one-tail	0.35	
t Critical one-tail	1.71	
P(T<=t) two-tail	0.70	
t Critical two-tail	2.06	

t-Test: Two-Sample Assuming Equal Variances
Product B Output

	Variable 1	*Variable 2*
Mean	310800	315122
Variance	27802019.603	35300850
Observations	13.000	13
Pooled Variance	31551434.756	
Hypothesized Mean Difference	0.000	
df	24.000	
t Stat	1.29	
P(T<=t) one-tail	0.10	
t Critical one-tail	1.71	
P(T<=t) two-tail	0.21	
t Critical two-tail	2.06	

WW30 Simulation Results

Results of t-tests between the WW30 output of the 2,2 simulations and the 1.5,1.5 simulations. For both products with α=0.05, $|t| < t_{critical, 2\ sided}$, so the null hypothesis that the means of the two samples are equal is not rejected.

t-Test: Two-Sample Assuming Equal Variances
Product A Output

	Variable 1	*Variable 2*
Mean	408816	408074
Variance	59827354.41	83867089
Observations	13.00	13
Pooled Variance	71847221.55	
Hypothesized Mean Difference	0.00	
df	24.00	
t Stat	0.11	
P(T<=t) one-tail	0.46	
t Critical one-tail	1.71	
P(T<=t) two-tail	0.91	
t Critical two-tail	2.06	

t-Test: Two-Sample Assuming Equal Variances
Product B Output

	Variable 1	*Variable 2*
Mean	323950	322002
Variance	56794619.77	49236382.3
Observations	13.00	13
Pooled Variance	53015501.04	
Hypothesized Mean Difference	0.00	
df	24.00	
t Stat	0.34	
P(T<=t) one-tail	0.37	
t Critical one-tail	1.71	
P(T<=t) two-tail	0.74	
t Critical two-tail	2.06	