Systems Analysis and Optimization through Discrete Event Simulation at Amazon.com

by

Cameron S. Price

B.S. Environmental Engineering, United States Military Academy 1996

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

Master of Business Administration
and
Master of Science in Ocean Systems Management

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

Signature of Author ______________________________
MIT Sloan School of Management
Department of Ocean Engineering
May 7, 2004

Certified by ______________________________
Jeremie Gallien, Thesis Advisor
J. Spencer Career Development Professor of Operations Management

Certified by ______________________________
Thomas Roemer, Thesis Advisor
Robert N. Noyce Assistant Professor of Operations Management

Accepted by ______________________________
Henry S. Marcus, Thesis Reader
Professor of Marine Systems

Accepted by ______________________________
Margaret Andrews, Executive Director of Masters Program
Sloan School of Management

Accepted by ______________________________
Michael S. Triantafyllou, Chairman of the Graduate Committee
Department of Ocean Engineering

Systems Analysis and Optimization through Discrete Event Simulation at Amazon.com

by

Cameron S. Price

Submitted to the Sloan School of Management and the Department of Ocean Engineering on May 7, 2004 in partial fulfillment of the requirements for the degrees of

Master of Business Administration
and
Master of Science in Ocean Systems Management

ABSTRACT

The basis for this thesis involved a six and a half month LFM internship at the Amazon.com fulfillment center in the United Kingdom.

The fulfillment center management sought insight into the substantial variation across the entire fulfillment center. This variation manifested itself primarily in two areas: the individual process path productivities of picking, pre-sort, re-bin and packing, and the overall system productivities of fulfillment center cycle time and throughput. Employee productivities, within picking, pre-sort, re-bin and packing, varied substantially, with this variation having a significant effect on throughput.

This thesis uses discrete event simulation and the program SIMUL8 to model the overall system and analyze the effect of this variation. It discusses the design and development process for the model, proposes key questions relative to variation, analyzes different scenarios, and recommends specific actions for implementation. The overall goals of the internship included increasing fulfillment center throughput and decreasing cycle time.

Thesis Advisors
Jeremie Gallien
J. Spencer Career Development Professor of Operations Management

Thomas Roemer
Robert N. Noyce Assistant Professor of Operations Management

Thesis Reader
Henry Marcus
Professor of Marine Systems

Acknowledgements

First and foremost, I would like to thank my wife, **Cam Marie**, for believing in my vision for our future and supporting my desire to attend graduate school. Her steadfast support never wavered through two years of business school applications, one pregnancy, and 8 moves within three years – spanning no less than 13,170 miles. I would also like to thank my daughter, **Malia Lynn** for her unbelievable resilience and good nature through the entire ordeal.

I want to thank my mother and father, **Jack and Sam,** for raising me to believe that anything was possible. I owe much of who I have become to their constant love and to the opportunities that they laid before me. And I want to thank my sister, **Megan,** for her constant support, her endless love, and her belief that she has the greatest brother in the whole world ☺

I also want to thank my aunt and uncle, **Ginny and Jim Taylor**, for always being there for me. Thanks for all the memories through four years at West Point and two at MIT! Thanks **Kim, Keith, Chris, Brooke & Jay**!

I also want to thank my past bosses for all of the lessons throughout the years…

Steve Young – It all started at Southern California Jewelry and Loan. Thanks for the lessons about life! And thanks for always being there when you were needed.

Ted Johnson – Thanks for helping me understand the meaning of passion and the importance of rallying the troops.

Nate Allen – Thanks for teaching me the power of delegation, the importance of thinking first, the need to hold people accountable, and your undying belief in developing the team.

Stu Faulk – Thanks for believing in me and allowing me the freedom to execute.

Jay George – Thanks for another lesson in delegation and for emphasizing the importance of implementing systems.

Tom Barth – Thanks for setting a wonderful example and for Best Ranger 2001!

Mark Shanker – Thanks for introducing me to the civilian world, believing in my abilities, and teaching me what a manufacturing environment can and should be.

I also want to thank the Leaders for Manufacturing Program at MIT…

Don Rosenfield – Thanks for the phone call and being the glue that sticks LFM together.

Bill Hanson – Thanks for believing in leadership and teaching me vision.

Class of 2004 – Wow! What a group of amazing people. Thanks for dragging me along!

Thomas Roemer – Thanks for being the voice of reason and understanding the personal issues of an LFM internship.

Jeremie Gallien – Thanks for the help with SIMUL8 and Visual Logic. I hope that I never have to use it again ☺

Finally, I would like to thank the entire operations team at Amazon for all their help and support – especially **Mark Mastandrea** for his understanding and commitment to the project. Thanks also to **Ben, Zvonko** and **Gary** for keeping me sane in Milton Keynes.

Table of Contents

Chapter 1: Description of Amazon's Operations

1.1 What the Customer Doesn't See...

While the customer is used to seeing the website, the order entry process, and packages arriving in the mail, there is a substantial amount of activity that happens behind the scenes at Amazon. Amazon offers over two million unique products, processes hundreds of thousands of multiple item orders and ships hundreds of thousands of packages through different national and international postal companies daily. Their websites cater to markets around the world, and their postal carriers ship customer orders anywhere that they can be delivered.

As a result, one of Amazon's core competencies is the ability to assemble, process, and ship multiple item orders from their extensive product line and get them to their customer's preferred address when promised or before.

1.2 Amazon's Operations over Time

As a company, Amazon currently operates three different types of fulfillment centers (FC): automated, hybrid (half automated and half manual) and manual.

In the beginning, Amazon started out carrying very little in the way of inventory. In fact, in 1995, Amazon offered over 1 million book titles of which only 2000 were carried in company inventory. The rest of the titles were shipped from publishers or wholesalers to their FC in Seattle.[1] As the company and business model evolved, Amazon opened their second FC in 1997 in the state of Delaware as a way of speeding up deliveries to the east coast.[2] These two FCs were completely manual with the entire inventory being processed by hand. As customer orders increased, their operations were forced to scale dramatically. As a result, the company sought greater efficiencies through the opening of new automated facilities. These facilities sought to process all customer orders with as little human touch as possible. Some of the advances included: huge conveyor sortation systems that enabled thousands of orders to be picked by hand and sorted by machine, packing systems that enabled orders to be packaged completely by machine, and miles of conveyors that moved inventory and WIP throughout the FCs.

As the company continued to expand, they sought to build out additional automated capacity. As Amazon struggled to get profitable, the huge infrastructure costs of the automated facilities started to come into question. At the same time, Amazon was experimenting with a new design which they chose to call a hybrid FC. These hybrid FCs incorporated the best of the automation from the automated FCs with more human labor for the sortation processes. To date, these hybrid FCs perform as well, if not better, than the automated facilities.

1.3 Amazon's Operations in the United Kingdom

[1] Harvard Case Study 9-798-063 Leadership Online: Barnes & Noble vs. Amazon.com (A) pg 8

[2] Harvard Case Study 9-798-063 Leadership Online: Barnes & Noble vs. Amazon.com (A) pg 14

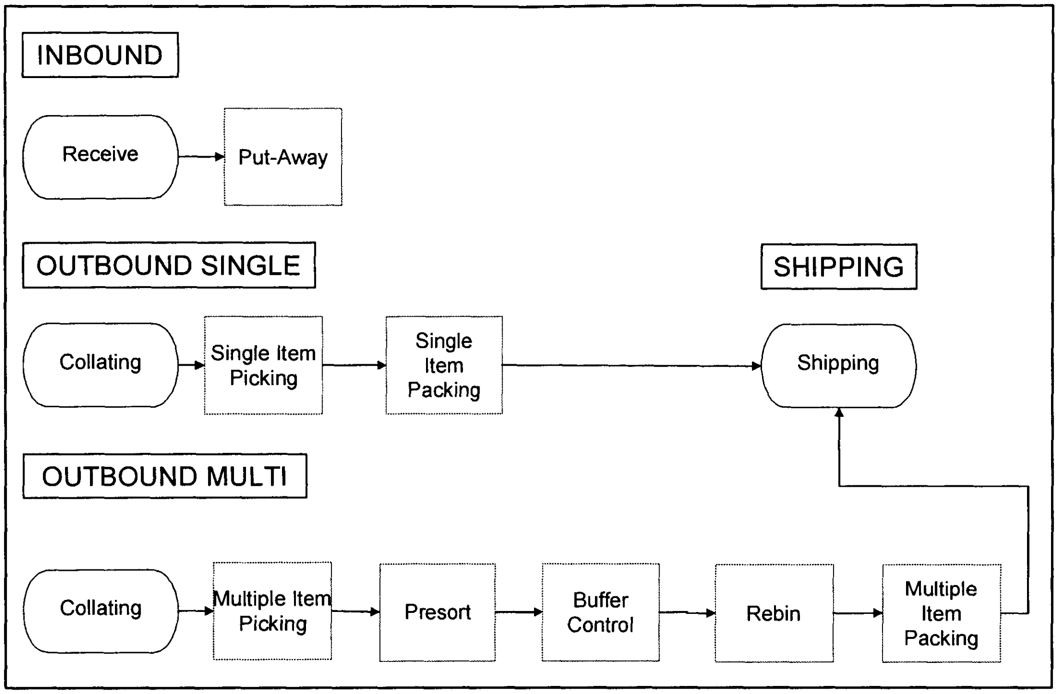


Figure 1.1 – Summary of the Portion of the Fulfillment Operation Under Study

As one of the company's two hybrid operations, Amazon's United Kingdom FC offers a process similar only to the German FC. At the back end of the system, inventory is received on the loading docks of the warehouse. The inventory comes into the building as normal pallets off of tractor trailers from the various regional suppliers. Each pallet consists of roughly 5 to 40 boxes or totes, each of which contains individual items that belong to either specific customer orders or are deemed to be items which Amazon's fulfillment specialists want to have in stock.

After arrival into the warehouse, the individual items of inventory are processed at **receive**. Receive consists of an Amazon associate removing inventory items from the incoming boxes or totes, scanning them into new Amazon inventory totes and placing them on the conveyor system to the next step in the process which is **put-away**.

At put-away, the items are scanned out of the inventory totes and onto Amazon's inventory shelves. In a nutshell, this is the extent of Amazon's **inbound** operations.

The majority of Amazon's **outbound** operations consist of two parallel processing paths: outbound single item orders and outbound multiple item orders.

The **single item order path** is relatively straightforward. Customer orders are aggregated into large groups of items (the process of **single item order collating**) and single item pick-lists are generated. There is one **pick-list** for each of Amazon's different inventory zones that were included in the order aggregation. Each pick-list includes all of the single item orders whose inventory is found within a particular inventory zone.

Associates take a pick-list and a single cart of totes and move around the inventory locations picking each of the single item orders on the pick-list. As they pick each item of inventory, they scan it into the specific inventory tote on their cart. Once an inventory tote is full, it is placed on the conveyor system. The conveyor system reads the tote identification, recognizes that it is a tote filled with single item orders and routes the inventory tote to a single item order packing line. At **single item packing**, the packer takes the individual items out of the tote, packs them into the appropriate size package, and places them back on the conveyor system. The conveyor system then routes the item to shipping, where it is automatically assigned a shipping label and postage and sent to a bin for the appropriate postal carrier.

The **multiple item order path** is substantially more complicated. Customer orders are once again aggregated into large groups of items, but this time it is done twice. This first time, orders are aggregated into **batches** (the process of **multiple item order collating**). One batch consists of enough orders to fill a **re-bin** station. Each re-bin station has a certain number of slots, which can hold on average 5 different items; however, each slot may contain, at most, only one customer order. As a result, any given batch will have a varying number of items in it. The next step in the multiple item order collating process is aggregating batches into **waves**. The diagram below has a graphical representation of this process.

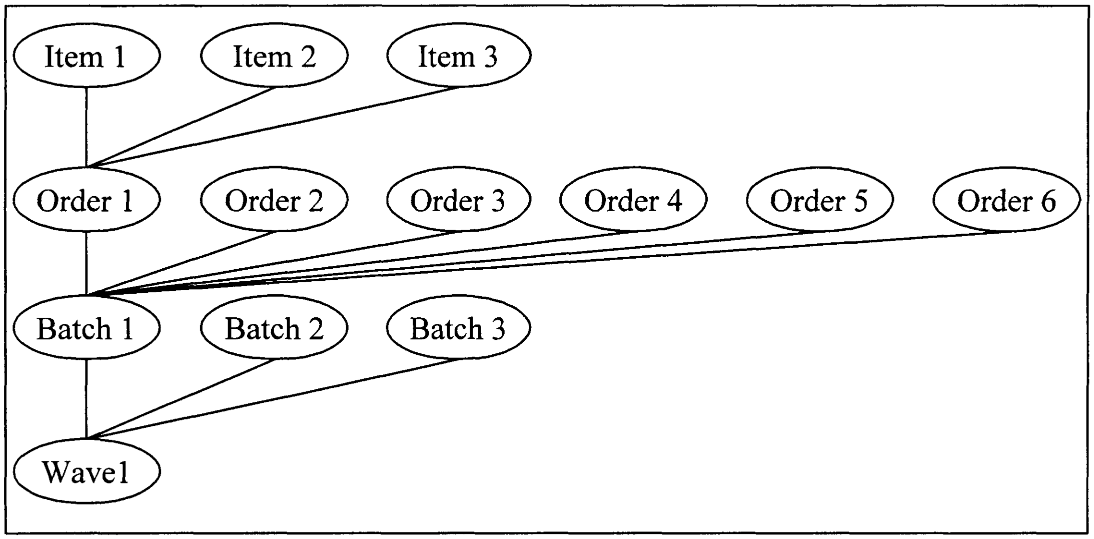


Figure 1.2 – Multiple Item Collating Process: The level of aggregation for items, orders, and batches varies, but this diagram provides a good general idea of what happens.

At this point, **pick-waves** are generated for each of the inventory zones from which the orders were aggregated. Each of these pick-waves includes all of the inventory items from the wave that are located in each of the inventory zones.

The next step in the multiple item order path is multiple item picking. Amazon associates receive a pick-wave and a wave picking cart. They move to the inventory zone where the items are located and pick the items off of the inventory shelves and place them onto the wave picking cart. Once they have picked all of the items from the wave-pick-list, they take the wave picking cart and the pick-wave to the next step in the process, **pre-sort**.

At pre-sort, another associate takes the wave picking cart, which includes all of the items from the pick-wave, and they separate the items from the pick-wave into their prospective batches. This is all accomplished at the pre-sort station. The associate takes one item off the wave picking cart, scans the item, and places it into the appropriate inventory tote. They continue to do this as long as there are still items on the wave picking cart. At the end of the process, the cart is empty, and the customer items have been separated into inventory totes. Each of these totes corresponds to the items, which happened to be within that inventory zone, from the original wave from collating. It is important to note that this is happening for each wave at multiple inventory locations. And, by the same logic, there are portions of each of the batches at each of the inventory locations. At this point, the inventory totes are moved to the next step in the process, **buffer control**.

The process of buffer control helps to synchronize the entire system. At buffer control, there is a team who ensures that each of the waves across all the inventory zones is complete before it is allowed to flow downstream in the multiple item order path. By computer, they track each of the waves within the system. As the pick-waves are completed at pre-sort, they show up as complete for that inventory zone. Once the pre-sort stations within all of the inventory zones have completed a particular wave, the buffer control people allow all of the completed totes from each of the pre-sort stations into the conveyor system. The conveyor system recognizes which totes belong to which batch, and it routes all of the totes from a particular batch into a single lane within the **batch accumulator.**

While the batch accumulator isn't depicted on the diagram above – as it is part of the conveyor system – it plays an integral part in the overall system processing. In essence, the batch accumulator aggregates all of the totes released from buffer control – from each of the different inventory zones – into their prospective batches. The diagram below depicts the picking, pre-sort and batch accumulation process.

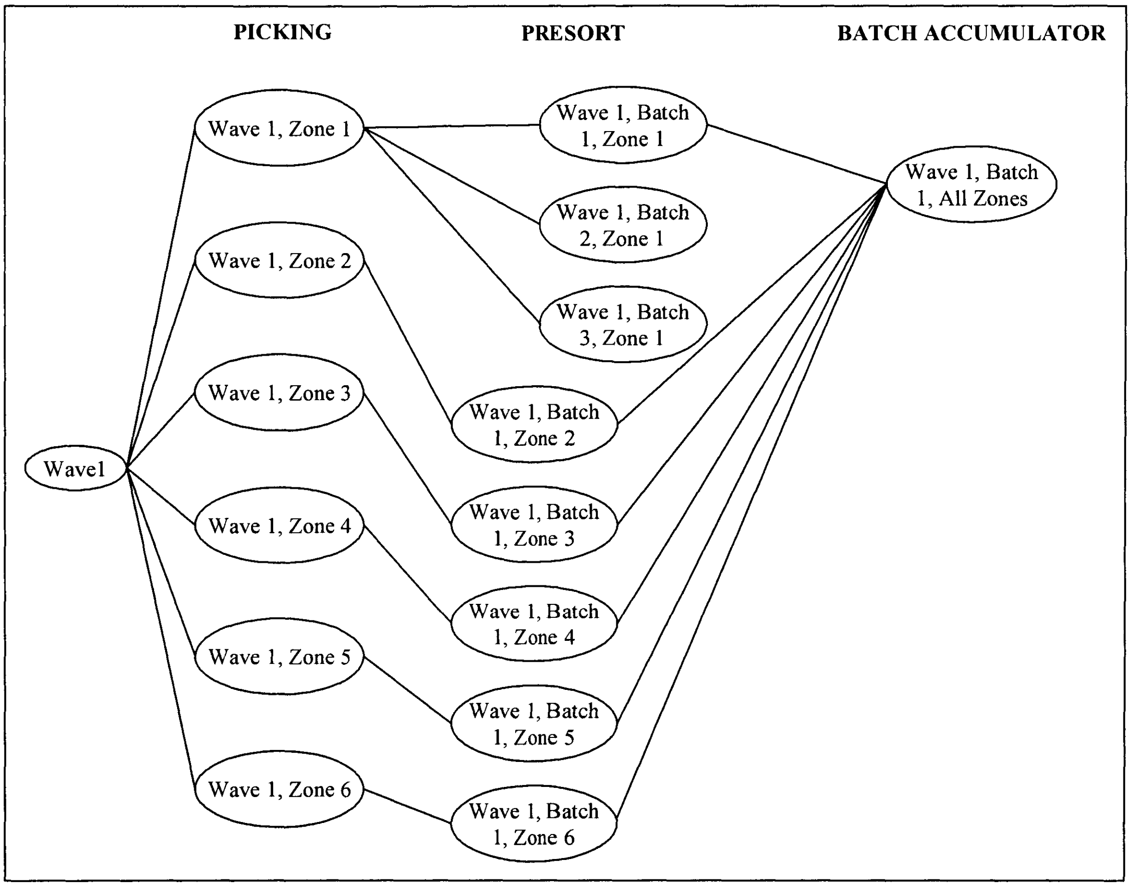


Figure 1.3 – Picking, Pre-Sort and Batch Accumulation Processes: The number of zones and batches will vary.

Once a batch is complete, with all of the totes that comprise a part of that batch from each of the inventory zones, it is on call to be released to the re-bin stations as they become empty. **Re-bin** is the next step in the process.

At re-bin, Amazon associates separate one complete batch into the individual customer orders. Every re-bin station has a certain number of slots, each of which houses a small black tote. When the station is empty, an Amazon associate will press a button which signals the buffer to send one complete batch to the station. The batch arrives in the form of totes full of inventory. The associate will pull one item at a time from the large totes, scan it, and place it in the appropriate re-bin slot. At the end of the process, all of the items have been moved from the large totes into the small totes of the re-bin rack. A depiction of the re-bin process is included below.

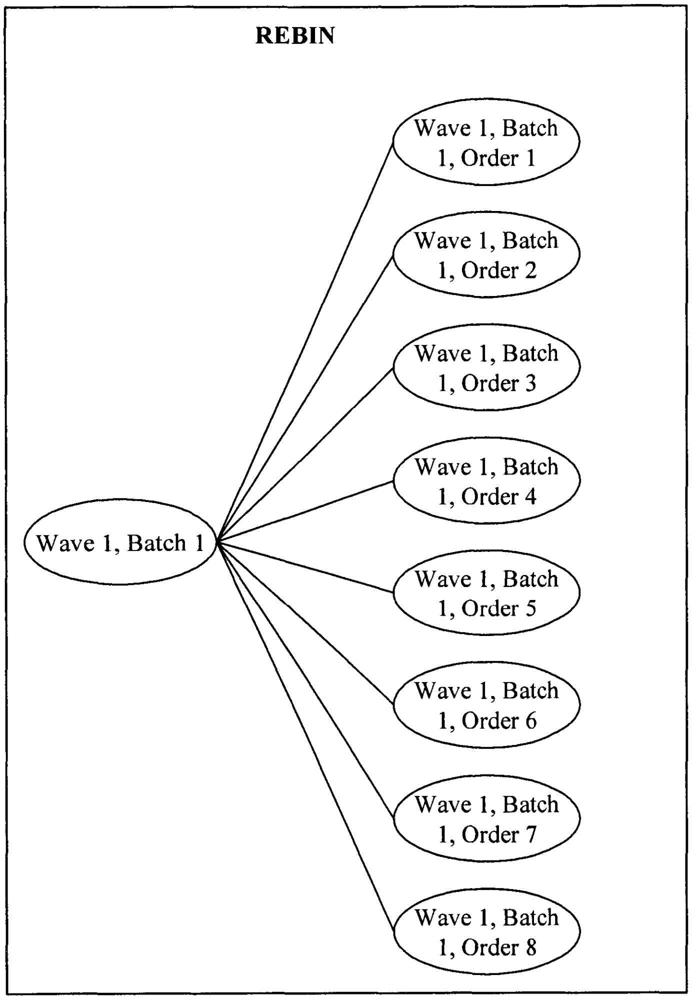


Figure 1.4 – Re-Bin Process: The number of orders in any batch will vary.

Once all of the items are in the re-bin rack, the associate will push all of the re-bin totes through the rack to a **multiple item packing** station.

At multiple item packing, an associate pulls one complete order from the back side of the re-bin rack, selects the appropriate size box and packs the order into the box. The associate then places the box onto the conveyor. The conveyor system then routes the item to shipping, where it is automatically assigned a shipping label and postage and sent to a bin for the appropriate postal carrier.

Chapter 2: Operational Issues at the United Kingdom Fulfillment Center

2.1 Problem Statement

When I first arrived on site, in the middle of April, I spent a weekend with the senior management of the facility. I spoke with the general manager, the senior operations manager, the human resources manager, and the engineering manager, and I spent about 12 hours on the FC floor. I could not have imagined a better support group for the internship. The main problem that the management team faced involved understanding the extensive variation within the system and across the different process paths. They were struggling with developing a tool that could provide management insight into where they should focus their limited resources to in order to create a more efficient process.

Within Amazon's UK Fulfillment Center, the two primary business metrics are system throughput and cycle time. During the 12 months preceding my internship, these metrics experienced significant fluctuations day to day. The operations team wanted to isolate the key drivers behind throughput and cycle time, and they sought insight and guidance into the significant variation that the fulfillment center was experiencing.

As such, I settled on the following problem statement:

Analyze the significant variation in the system, uncover the key drivers of throughput and cycle time, and develop specific recommendations for improvement of these two business metrics.

2.2 Problem Symptoms

2.2.1 Inventory Flow Problems

During my six months at Amazon, I observed several areas of flow problems within the FC. These areas primarily involved the buildup of WIP at certain points (picking queue, pre-sort queue, buffer control queue, and the re-bin queue) in the system and the starvation of different processes within the system (re-bin and packing).

2.2.2 Productivity Variation

On the micro scale, productivities within each process path (picking, pre-sort, re-bin and packing) vary dramatically from employee to employee. The combined effect of these variations has a significant effect on the total system performance.

2.3 Managerial Areas of Influence

In order to address the problem stated above, I first attempted to determine which operational levers the management team was capable of influencing. These managerial levers included fulfillment center design, fulfillment center control, and human resource policies – relative to training and incentive programs.

2.3.1 Fulfillment Center Design

Fulfillment center design should be based on the theory of constraints. The system bottleneck should be chosen by management, and the rest of the process capacities should be balanced and measured relative to that chosen bottleneck. This enables the operations team to focus their resources on the bottleneck process to ensure that if is fully utilized throughout the operations time period.

During my six months at Amazon, I spoke with many people across the operations team, at every different level of management. It became clear that there was no consensus on exactly where the bottleneck in the UK fulfillment center actually lay. Some people thought that it was in packing, some people felt that it was in re-bin, and still others felt that it was in pre-sort. As a result, I felt that it was necessary to locate the system bottleneck, and this became my first issue for analysis.

2.3.2 Fulfillment Center Control

Given the facility that currently exists at Amazon UK, the primary lever that the operations management team controls on a day to day basis is the amount of inventory present at different places in the system. This work in process resides in the queues that form in front of the individual processes of picking, pre-sort, buffer control, re-bin and packing. The operations team adjusts the level of inventory by initially allocating labor to the individual processes within the system and by adjusting the amount of labor throughout the day.

Again, through my six months at Amazon, I realized that there was no consensus within the operations team on the optimal buffer size of these queues during startup and at different times throughout the day. As a result, I felt that gaining insight into the effect of the work in progress and identifying the optimal buffer size at startup was important, and this became my second issue for analysis.

2.3.3 Human Resources: Workforce Training and Incentives

Understanding the impact on the primary business metrics (throughput and cycle time) of increasing mean productivities and decreasing the standard deviation of productivities will enhance management's ability to dictate human resource policies. As a result, I chose to focus on uncovering the effect on cycle time and throughput of changing the productivity mean and standard deviation.

2.4 Summary

Amazon's UK fulfillment center's key goals are increasing throughput and decreasing cycle time. They can affect these two business metrics through system design, management policies, and human resource training and incentives. The three main areas of research going forward are:

- Identifying the system bottleneck from picking, pre-sort, buffer control, re-bin and packing.
- Understanding the impact of work in progress and identifying the optimal buffer size for picking, pre-sort, batch accumulation, re-bin and packing.
- Uncovering the effect of changing the productivity mean and standard deviation.

Chapter 3: Design of Experiments, Results and Analysis

3.1 Graph Interpretation

Many of the graphs that are included below are inventory graphs over time. Because these graphs all show the same output under different simulation conditions, I made the decision to scale them all exactly the same. This dramatically enhances one's ability to compare the results across all of the different experiments – each of which is based on different model conditions.

The vertical axis units are batches (for buffer control, the batch accumulator, re-bin and packing) or pick-waves (for picking and pre-sort), and they are all scaled from 0 to 200. The horizontal axis unit is time in minutes, and it is scaled from 0 to 373 units. One result of this is that some of the experiments run for less time and the output is cut short. Another impact is that some of the graphs are more compact vertically than would be ideal. However, I believe that the benefit of comparing the graphs across experiments far outweighs the cost of not having an expanded graph.

In addition, the graph titles show the experiment number – relative to the actual thesis section – as well as the following key (% of actual productivity / % of actual standard deviation / arrival rate in units – startup picking buffer in pick-waves / startup pre-sort buffer in pick-waves / startup re-bin buffer in batches / startup packing buffer in batches). Finally the legend at the bottom shows which marks belong to which inventory graph.

A final note is that I included a red line at 36 on each of the graphs. This red line depicts the minimum acceptable level of inventory for both re-bin and packing. If the re-bin inventory or the packing inventory falls below this line, there will be an immediate decrease in total throughput for the day. In reality this red line just aids the viewer in interpreting the graphical data.

3.2 Fulfillment Center Design – Identification of the Bottleneck

Within any operations or manufacturing environment, it is crucially important for the operations team to understand which process within the overall system is the bottleneck. This enables the team to focus their resources on that process to ensure that it is fully utilized throughout the operations time period. In addition, it allows them to balance the system relative to the bottleneck.

3.2.1 Process Followed

In order to start my system analysis, I chose to do a simple capacity study using the mean productivity data and actual number of processing stations within the Amazon UK facility. The calculations that I conducted involved multiplying the total number of stations for each of the process paths by the mean productivity. The percent capacity results of these calculations are included below, and the capacities shown are relative to the highest capacity process, which turned out to be picking.

	Capacity
Pick	100%
Pre-sort	53%
Re-bin	45%
Pack	51%

Table 3.1 – Capacity Analysis

Looking at this data, it is immediately clear that picking is not the system bottleneck. On the other hand, pre-sort, re-bin and packing all represent possible system bottlenecks as their productivity data is really close.

The next step in the process was to actually run the simulation under several different scenarios to gain further insight into the process capacity of pre-sort, re-bin and packing. For these experiments, I enabled each of the process workstations to have their own labor source, thereby ensuring that they could process inventory if it was available. A chart depicting the different experiments is shown below.

	Base #1	Base #2	Exp #1	Exp #2	Exp #3	Exp #4
Statistical Measures						
% of Actual Standard Deviation of Productivity	100%	100%	0%	100%	100%	100%
% of Actual Mean of Productivity	100%	100%	100%	100%	100%	100%
% of Actual Arrival Rate (Units/Hr)	139%	139%	139%	139%	111%	100%
Startup Queues						
Pick Buffer (Wave Picklists)	175	175	175	150	150	150
Pre-Sort Buffer (Wave Picklists)	55	55	55	55	55	30
Rebin Buffer (Batches)	60	60	60	85	85	85
Packing Buffer (Batches)	36	36	36	36	36	36
% Capacity of Actual Facility						
Picking Stations	100%	100%	100%	100%	100%	100%
Pre-Sort Stations	100%	100%	100%	100%	118%	118%
Re-bin Stations	100%	100%	100%	100%	100%	100%
Packing Stations	100%	100%	100%	100%	100%	100%

Table 3.2 – Bottleneck Experiments

Due to the long simulation run time of each of the above experiments, I was unable to test the variability of output across individual experiments using different random number seeds. As a result, confidence intervals for the individual experiments are not included. Looking at the difference between Base #1 and Base #2 gives you a good idea of the impact of a different random number seed on the same experiment; while the trends are the same, the actual minute by minute results are different.

3.2.2 Bottleneck Base #1: Amazon's Actual Operations (1)

For the first experiment, I wanted to get a baseline inventory graph that reflected Amazon's actual productivities and startup conditions.

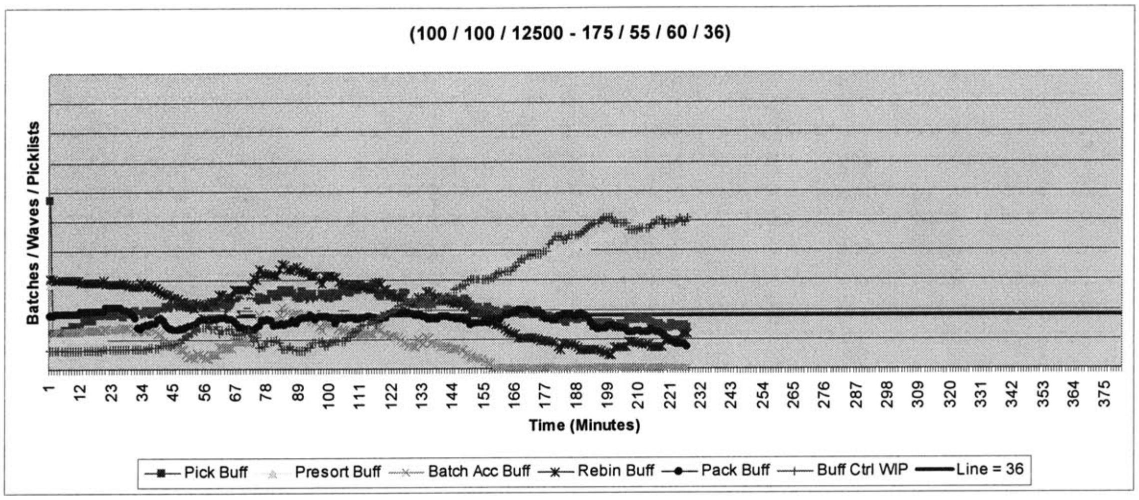


Figure 3.1 – Bottleneck Base #1

Looking at this graph, it is readily apparent that the pre-sort queue and the buffer control queue continue to trend upward. In addition, the re-bin and packing queue both dip below the red line – in other words, they are underutilized. This suggests that pre-sort may be the bottleneck. In addition, it also reflects the environment that I experienced at Amazon's fulfillment center.

3.2.3 Bottleneck Base #2: Amazon's Actual Operations (2)

For my next experiment, I chose to run this same experiment over – with a different random number seed - to get a feel for whether the results would be similar and to get another data point for Amazon's actual operations. The inventory graph is depicted below.

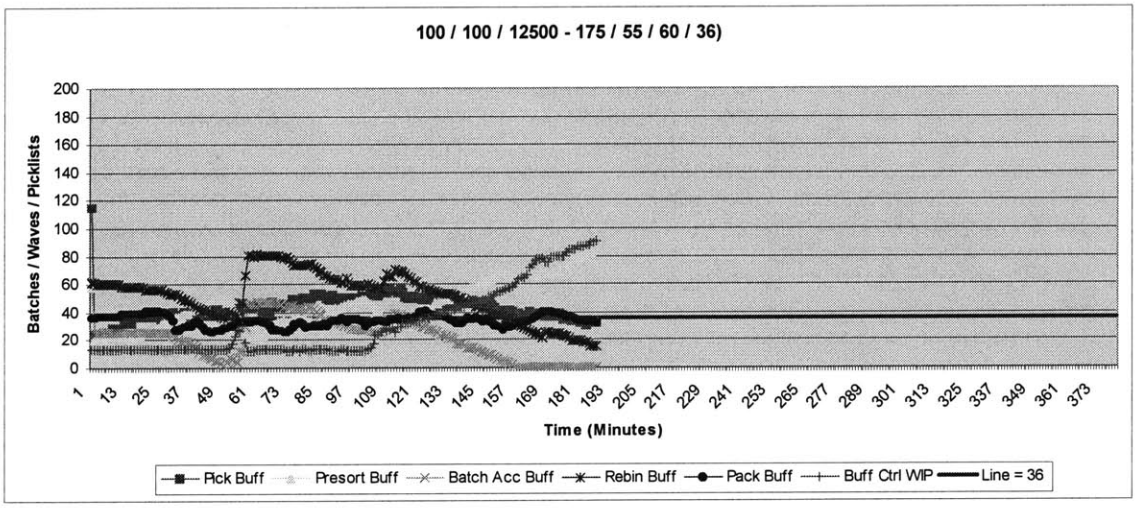


Figure 3.2 – Bottleneck Base #2

Looking at this inventory graph, we see very similar output results. The pre-sort queue and the buffer control queue are trending upward, and again, the re-bin and packing queues are both trending downward and underutilized at the end of the simulation. This graph also suggests that the system bottleneck lies in pre-sort and again it reflects the reality of what actually occurred at Amazon during the peak days when I was present in the facility.

3.2.4 Bottleneck Experiment #1: Standard Deviation Reduced to Zero

For the next experiment, I chose to decrease the standard deviation to zero to see if this would uncover more information about the bottleneck. The results of the inventory graph are depicted below.

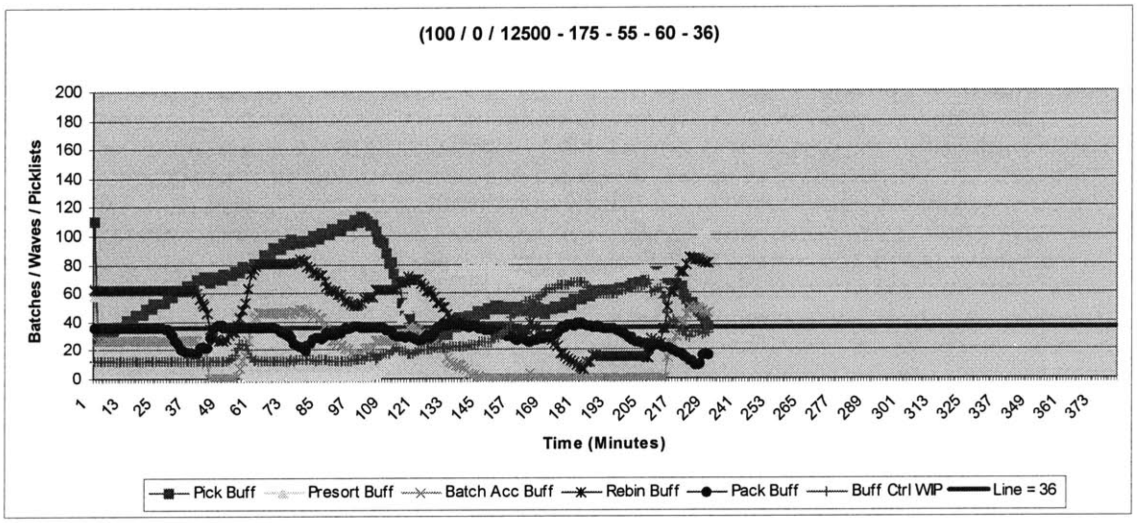


Figure 3.3– Bottleneck Experiment #1

Looking at this experiment, it is interesting to note that the system seems to become less stable / more erratic with zero standard deviation in the productivity values. This is the case because the only variation in the system is the variation in the order sizes, which results in variation in the amount of items in batches, waves, and pick-waves. This is a direct result of the fact that all of the workers start work at the same time. So, 100 minutes after the system start, most of the pickers are finishing their work, which explains the huge dip in picking inventory. About 60 minutes after the system start, all of the pre-sorters are finishing their work, which explains why the pre-sort inventory dives at 60 minutes. The same is true of re-bin and packing. As the simulation time advances, these fluctuations become smoother because of the compounded variability in the sizes of the batches and pick-waves.

Another observation is that because of the limited variation in processing times, the queue for buffer control is remarkably low relative to the other simulation experiments – under 30 batches after almost 4 hours. Again, like in the other two experiments, the pre-sort queue seems to be trending upward. In addition, at the end of the experiment, the re-bin queue also seems to be trending upward. This difference in re-bin is due to the low inventory in buffer control. Because the buffer control inventory is low, that means that the inventory has passed through buffer control to re-bin. Finally, from time 170 to 220, the re-bin inventory dips below the red line, and at the end of the simulation, the packing buffer is also below the red line, underutilized and declining.

3.2.5 Bottleneck Experiment #2: Change in Inventory Startup Buffers to Improve Re-bin and Packing Utilization

For this experiment, I adjusted the picking startup inventory down to decrease the amount of inventory in the system, and I adjusted the re-bin inventory up to see if I could keep it fully utilized throughout the simulation.

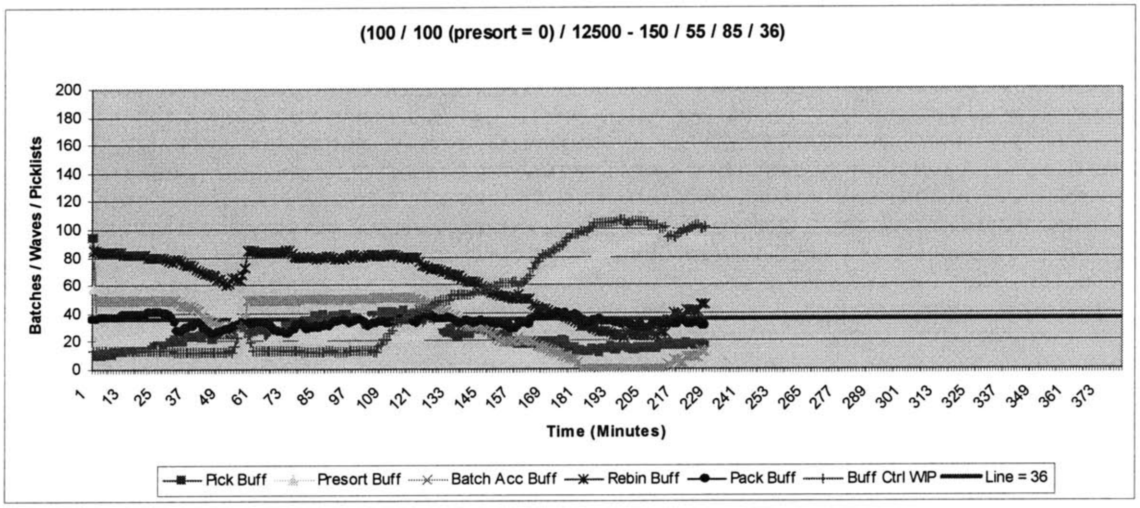


Figure 3.4 – Bottleneck Experiment #2

Again, the first thing that is clearly evident is that the pre-sort and buffer control inventory levels are trending upward for the majority of the simulation. However, with this experiment, the re-bin and packing inventory levels stay above the red line or just under the red line. So, while the system is fully utilized, we still have the problem of inventory buildup in front of pre-sort. However, for the final minutes of the simulation, it appears that the pre-sort and buffer control inventory levels may be stabilizing.

3.2.6 Bottleneck Experiment #3: Increase in Pre-sort Processing Capacity

For this experiment, I chose to increase the pre-sort processing capacity to determine if this process represented the system bottleneck. I increased it 18% by adding additional pre-sort stations within the SIMUL8 model.

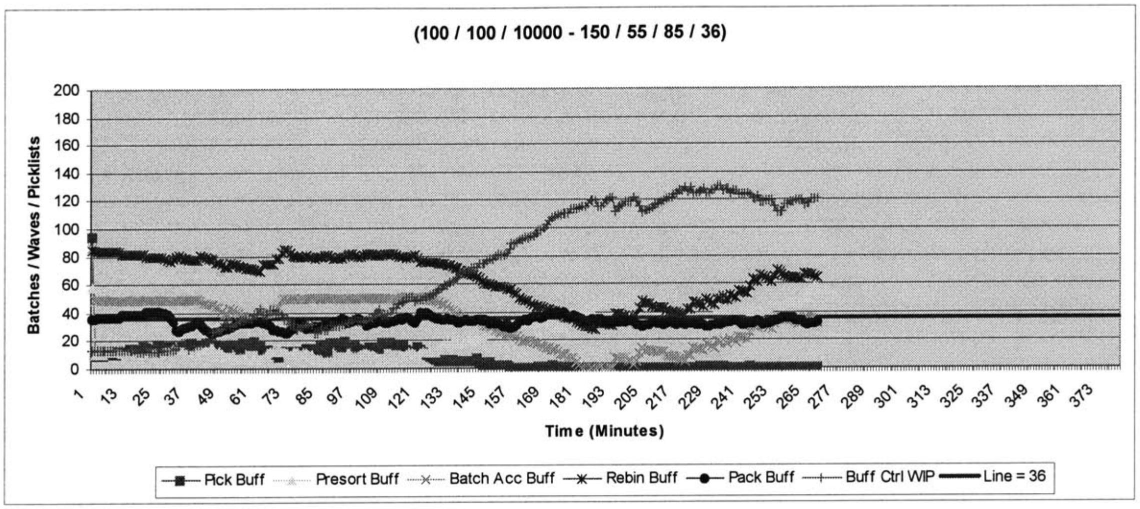


Figure 3.5– Bottleneck Experiment #3

This graph clearly shows that, at the current productivities, pre-sort is the bottleneck. However, as you increase the number of pre-sort stations, the pre-sort buffer inventory drops, and the re-bin queue becomes the new bottleneck. In addition, it seems that the buffer control queue stabilizes around 3 hours of simulation time. Also, the packing buffer is very stable with this iteration at right around 36 batches, which should translate to maximum throughput for the production period. One troubling note is that the re-bin buffer seems to be trending upward toward the end of the simulation.

3.2.7 Bottleneck Experiment #4: Increase in Pre-sort Processing Capacity

My final bottleneck experiment aims to take a little deeper look at the re-bin queue further into the simulation. In addition, I have adjusted the pre-sort starting buffer down to 30 pick-waves in order to keep the inventory in the model as low as possible. The graph of the last experiment is below:

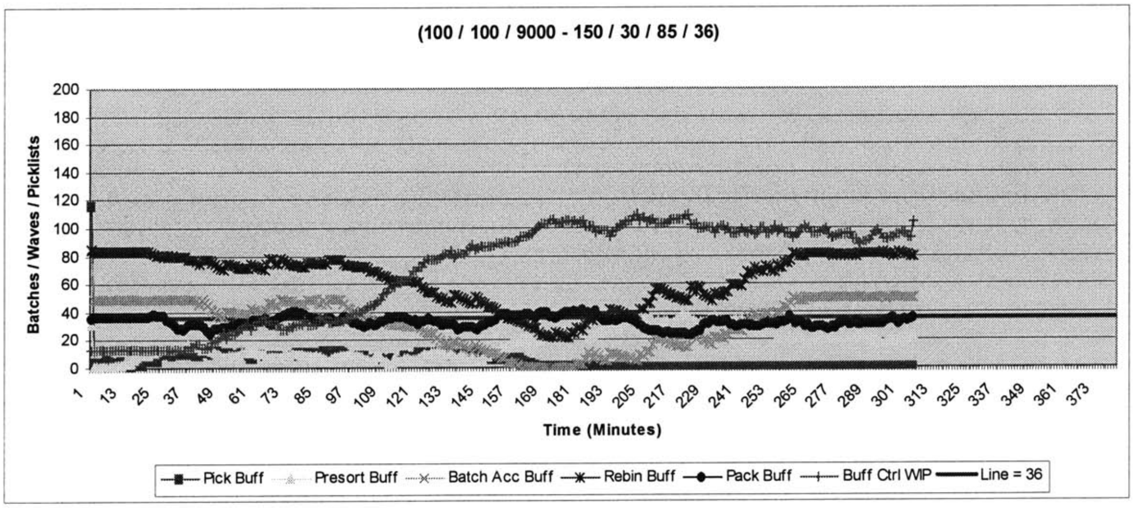


Figure 3.6 – Bottleneck Experiment #4

Analyzing this graph, you see that the re-bin inventory is trending upward and actually runs up against its full capacity at the end of the simulation time. This clearly demonstrates that re-bin is a secondary bottleneck, just behind pre-sort.

3.2.8 Conclusions

Given the infrastructure of Amazon's fulfillment center, the current bottleneck at the given worker productivities is pre-sort. This bottleneck may be relieved by adding additional pre-sort processing stations. In addition, directly behind the pre-sort bottleneck is the re-bin bottleneck. This bottleneck may also be relieved by increasing additional re-bin capacity within the facility. Finally, packing also represents another potential bottleneck within the system. As a result, it is important to ensure that packing capacity is increased along with pre-sort and re-bin capacity.

3.3 Identifying the Optimal Buffer Size

During my 6 months at Amazon, there was a lot of discussion among the operations managers on the floor relative to the impact of buffer sizes at the shift startup. Most operations managers felt that coming into a shift with full buffers across the system was the best way to have a great day of production. However, the general manager of the facility continually focused on the cost of additional WIP (excess buffers) relative to cycle time and ease of management. Understanding the relationship of buffers on throughput and cycle time is crucial to getting a feel for the pulse of the operations. Reaching a conclusion on the optimal buffer size, relative to these business metrics, would enhance the operations managers' ability to run a successful and efficient shift.

3.3.1 Process Followed

In order to test different buffer condition, I ran several experiments. I started with an experiment that held the initial startup buffers at zero. Then I ran two experiments with the buffers at maximum capacity. Finally, for the next three experiments, I adjusted the individual buffers up and down to get a feel for whether I could improve over the other conditions.

There are several different variables that go into the optimal buffer configuration. First, there are the business metrics of throughput and cycle time. In addition, there are ease of management issues: Are the buffers stable or erratic? Is the work in progress level high or low? Some of these issues are concrete and some are subjective – but I have chosen to address both in the experiments that follow.

The table below details the key inputs to the simulations that I ran for each of the experiments. The initial buffer inventory levels represent the amount of inventory at startup before any inventory flows into the individual work stations. Depending on the amount of people working in the individual process paths, these startup buffers will decrease dramatically at the beginning of the shift.

	Exp # 1	Exp # 2	Exp # 3	Exp # 4	Exp # 5	Exp # 6
Initial Buffer Inventory Levels						
Picking (Pick-wave)	148	300	296	150	150	150
Pre-sort (Pick-wave)	52	104	104	30	30	30
Re-bin (Batch)	36	85	85	85	85	55
Packing (Batch)	36	36	36	36	36	36
Statistical Measures						
% of Actual Arrival Rate	100%	100%	100%	100%	100%	100%
% of Actual Prod Mean	100%	100%	100%	100%	100%	100%
% of Actual Prod St Dev	50%	100%	100%	100%	50%	50%
Results						
Time (Minutes)	275	117	117	275	275	275
Inventory (Units)	35,925	59,619	57,959	34,226	37,142	36,205
Cycle Time (Hours)	3.99	6.62	6.44	3.8	4.13	4.02
Throughput (Percent of Exp #1)	100%	44%	44%	109%	110%	99%

Table 3.3 – Buffer Experiments

Due to the long simulation run time of each of the above experiments, I was unable to test the variability of output across individual experiments using different random number seeds. As a result, confidence intervals for the individual experiments are not included. Looking at the difference between experiments #2 and #3 gives you a good idea of the impact of a different random number seed on the same experiment; while the trends are the same, the actual minute by minute results are different.

In addition, you will notice that the Percent of Actual Productivity Standard Deviation changes from 50% to 100% and back to 50% within the series of experiments. This was an unintended change and really results in a decrease in the amount of WIP held in front of buffer control.

3.3.2 Buffer Experiment #1 – Zero Buffers at Startup

The first scenario that I ran involved startup conditions where there were zero buffers across the entire system. In essence, each of the available work stations has one unit (batch or pick-wave) of inventory. The inventory graph is depicted below.

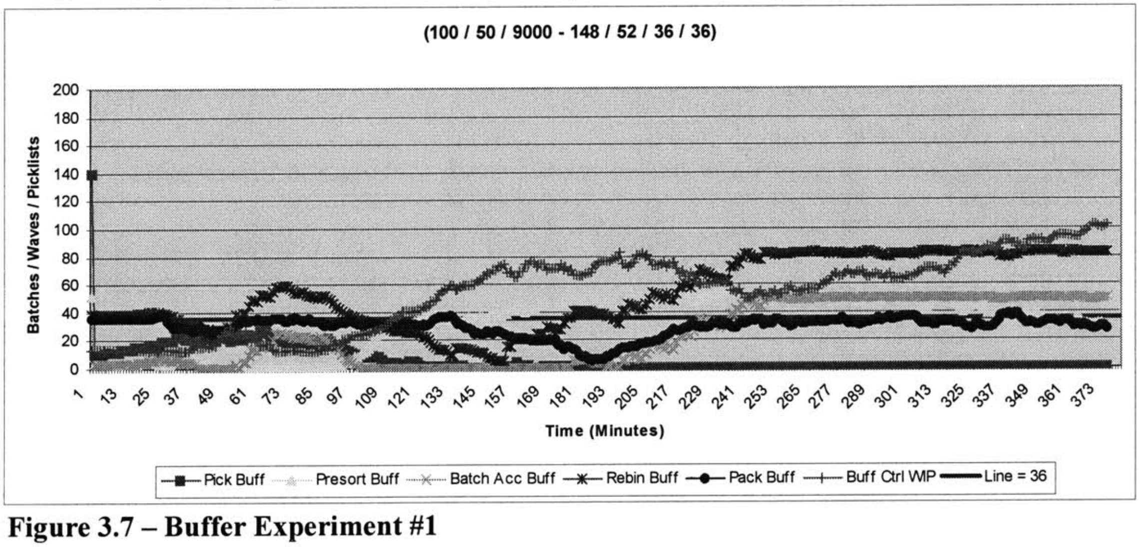


Figure 3.7 – Buffer Experiment #1

Looking at this graph, it is immediately evident that the inventory levels are very low – which translates to an easier job managing the operation. However, in the middle of the simulation run, from 130 to 245 minutes, the packing buffer is underutilized. And at the end of the simulation run, you can see that the batch accumulator buffer is filled to capacity at 49 batches and the re-bin buffer is also filled to capacity. Finally, you see the inventory at buffer control building toward the end of the simulation. The cycle time at 275 minutes is 3.99 hours; I chose 275 minutes because this allows me to compare experiments across most of the inventory graphs for this section.

3.3.3 Buffer Experiments #2 and #3: Full Inventory Buffers

The next two scenarios that I ran involved setting the buffers much higher at startup, in effect allowing for every picker to have two pick-waves, every pre-sort station to have two pick-waves, a full batch accumulator buffer, and a batch at every re-bin and packing station. The inventory results for two separate runs are below.

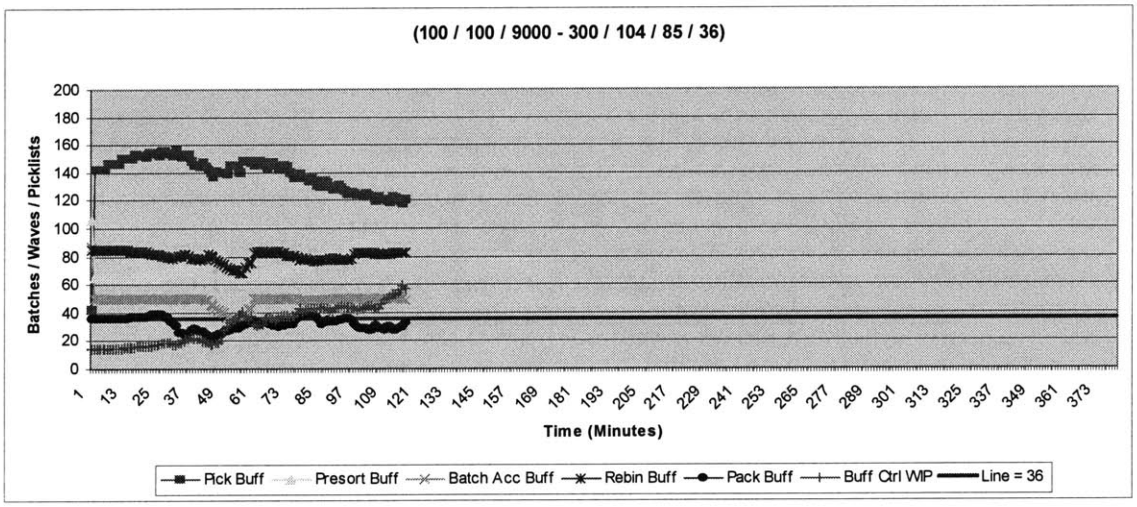


Figure 3.8 – Buffer Experiment # 2

Looking at the graph above, it is immediately clear that the inventory buffers are very high across the system. In addition, the re-bin queue and the batch accumulator queue are filled to maximum capacity. Finally, the buffer control queue is definitely trending upward. For these two experiments, I gathered the cycle time and throughput data from 117 minutes because the system crashed so early. In this case, the cycle time was 6.62 hours.

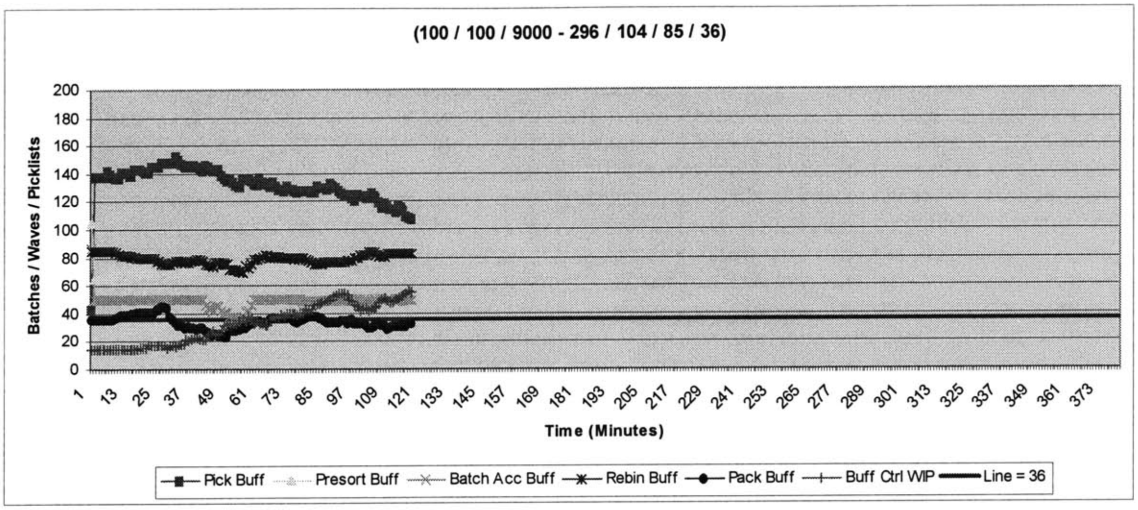


Figure 3.9 – Buffer Experiment #3

The observations for this graph are exactly the same as the last one. In addition, the cycle time at 117 minutes was 6.44 hours.

The system in real life, with the inventory buffers like the ones depicted here would be extremely difficult to manage. To begin with, because the picking buffer is so large and this process capacity is much larger than the bottleneck process capacity, you would end up with lots of additional inventory in the system. Furthermore, the batch accumulator is filled to capacity from startup – this would cause an upstream blockage for pre-sort and buffer control, which would translate to a lot of inventory stacked on the operations floor after pre-sort.

Another observation that should be noted from these two experiments is the short length of simulation time. Because of the full buffers, the system is less stable within the simulation program. As a result, the system crashed on both of these iterations at right around 2 hours of simulation time.

3.3.4 Buffer Experiment #4: Pre-sort Queue Lowered and Re-bin Queue Increased

For the next experiment, I chose to increase the re-bin queue to see if I could prevent the underutilization of packing (from the first experiment), and I decreased the pre-sort queue because it seemed fairly stable in the last several runs.

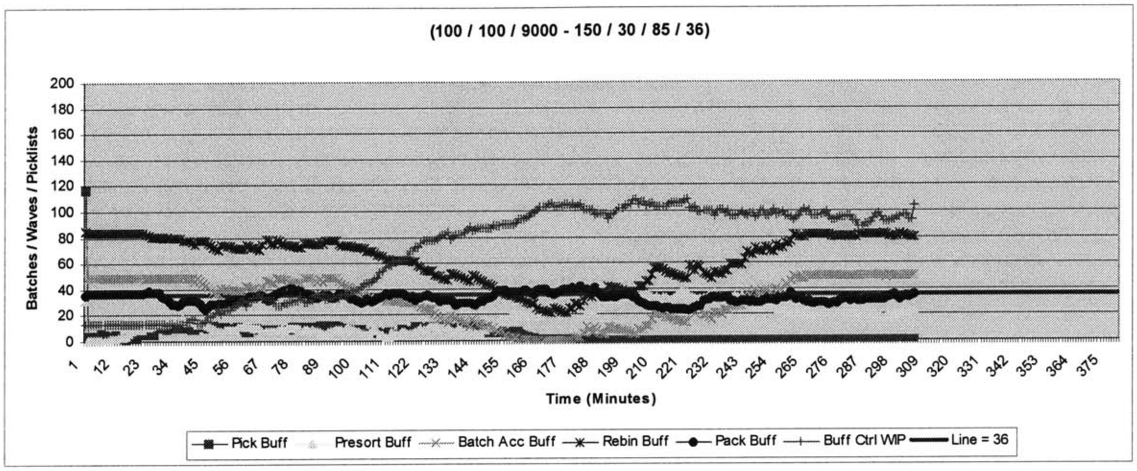


Figure 3.10 – Buffer Experiment #4

This experiment actually looks pretty good. The picking and the pre-sort inventory stays rather low within the system, and the packing process is utilized almost at 100%. In addition, it seems like the buffer control inventory is stable at around 100 batches. However, at the end of the simulation time, you see the batch accumulator and the re-bin buffers filled to capacity. The cycle time at 275 minutes is 3.8 hours. This is substantially better than the first experiment, in terms of business metrics and in terms of ease of management.

3.3.5 Buffer Experiment #5: Repeat of the Above Experiment with the Productivity Standard Deviation Set to 50% of Amazon's Actual Productivity Numbers

For this experiment, I chose to repeat the last experiment, with the exception of the productivity standard deviation which I set to 50% of Amazon's actual productivity numbers. The inventory graph is depicted below.

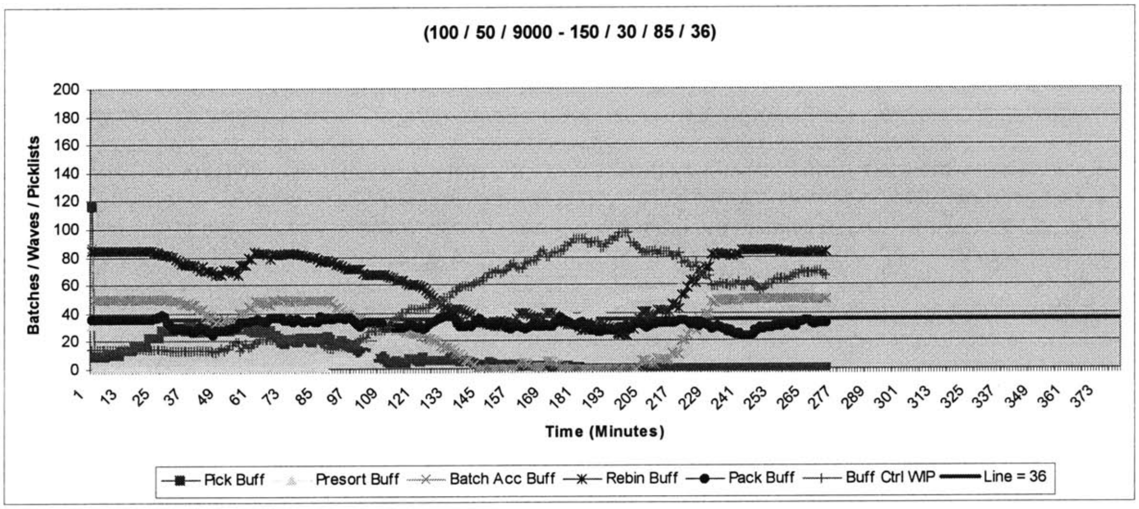


Figure 3.11 – Buffer Experiment #5

Again, this experiment looks very good. The primary difference between this graph and the last one is that the buffer control queue stabilizes around 60 batches which is substantially easier to manage than if it were at 100 batches. The cycle time at 275 minutes is 4.13 hours. This is the best result yet! However, one troubling note is that the re-bin buffer and the batch accumulator buffer are both filled to capacity at the end of the simulation.

3.3.6 Buffer Experiment #6: Decreased Re-bin Queue

Based on the experiment above, I chose to decrease the re-bin queue to see if I could cause the model to stabilize at a re-bin queue value of less than full capacity. The inventory results of this experiment are below.

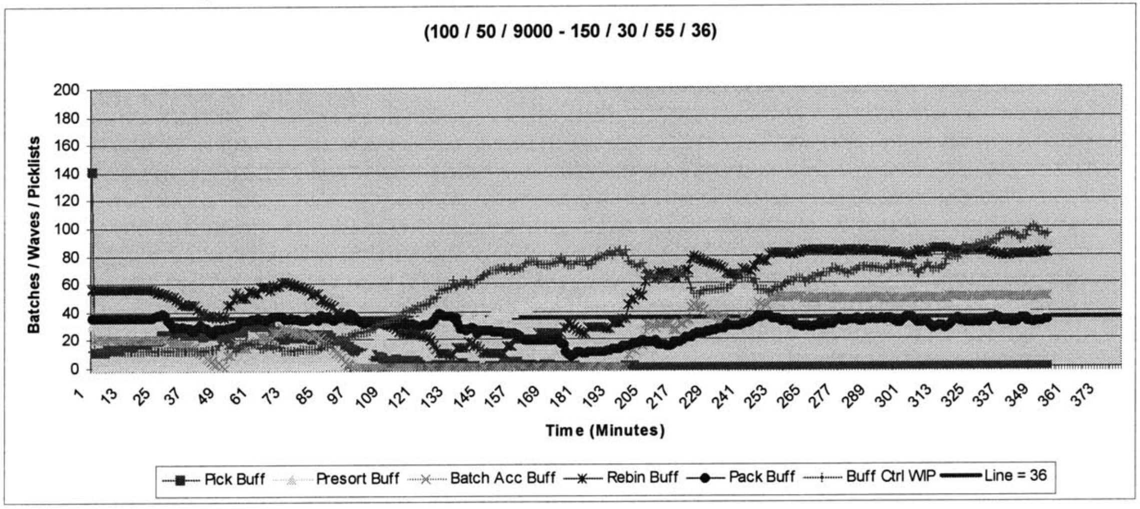


Figure 3.12 – Buffer Experiment #6

In the middle of this simulation run, you see the packing buffer start to decline, which would have an immediate effect on throughput, and the re-bin and batch accumulator buffers still seem to max out at about the same time. In other words, it seems best to keep the re-bin buffer higher at startup. For this experiment, at 275 minutes, the cycle time is 4.02.

3.3.7 Conclusions

In conclusion, I would recommend keeping the picking and pre-sort buffers as low as possible – in other words, starting the system with one pick-wave per processing station. By matching the amount of inventory coming into the system with the bottleneck process (at this point re-bin, as I've increased pre-sort capacity), you could maintain the picking and pre-sort inventories very close to zero. In addition, I would attempt to start the system with a full buffer at the batch accumulator and a complete batch in front of each of the packing and re-bin stations.

3.4 Human Resources: Workforce Training and Incentives: System Effect of Changing the Productivity Mean and Standard Deviation

Uncovering the effect of productivity mean and standard deviation relative to key business metrics is critical for any operation. If Amazon does not fully grasp the key drivers of throughput and cycle time, it is very difficult to design human resource policies to enhance operations and improve the business metrics.

Some of these human resource policies might include pulling out your top performers to be trainers for the slowest performers, thereby decreasing the productivity standard deviation. Or they might include pulling out your bottom performers for indirect labor to increase the productivity mean. Or they might include setting an incentive for a particular productivity to decrease the standard deviation. Or they might include setting an incentive that continues to pay additional compensation as the productivity improves, in order to increase the productivity mean.

3.4.1 Process Followed

I analyzed these two variables by running a number of experiments where I varied the work entry rate as well as the mean and standard deviation of the productivity. I varied the work entry rate to match the change in productivity when the mean was increased or decreased. In addition, because I felt that this might introduce a third variable into the experiment, I also varied the work entry rate while keeping the mean and standard deviation the same. A chart of the different experiments run is included below:

	Base	Exp # 1	Exp # 2	Exp # 3	Exp # 4	Exp # 5
% of Actual Mean Productivity	100%	100%	100%	150%	133%	66%
% of Actual St Dev Productivity	100%	50%	0%	50%	50%	50%
% of Actual Entry Rate(Units / Hr)	100%	100%	100%	150%	133%	100%
	Exp # 6	**Exp # 7**	**Exp # 8**	**Exp # 9**	**Exp # 10**	**Exp # 11**
% of Actual Mean Productivity	80%	100%	100%	100%	100%	100%
% of Actual St Dev Productivity	50%	50%	50%	50%	50%	50%
% of Actual Entry Rate(Units / Hr)	100%	139%	111%	89%	128%	100%

Table 3.4 – Productivity Experiments

In analyzing these different scenarios, I first looked at the impact of changing productivity mean and standard deviation on the business metrics of throughput and cycle time. Then, I went into several of the most enlightening inventory graphs for each of the experiments to see the impact of changing these variables on the actual buffers in the system over time.

Due to the long simulation run time of each of the above experiments, I was unable to test the variability of output across individual experiments using different random number seeds. As a result, confidence intervals for the individual experiments are not included. Similar to the last two sections (3.2 and 3.3), the results across a single experiment with different random number seeds seem to uncover overall trends while having different minute to minute results.

As you look at the graphs of inventory over time with each of these scenarios, you will notice that some of the experiments have wildly fluctuating buffers, some have buffers filled completely to capacity, and some have buffers that remain relatively steady over time. As I get to the individual experiments, I will discuss each in turn.

3.4.2 Throughput Results as a Function of Changing Arrival Rate and Productivity Standard Deviation

Before I analyzed these experiments, I thought about what should happen in each of these cases. I felt that productivity standard deviation should have little impact on throughput results and that productivity mean should have a direct impact on throughput results. Comparing the results from the base scenario and experiments 1, 2, 7, and 8 resulted in the graph below.

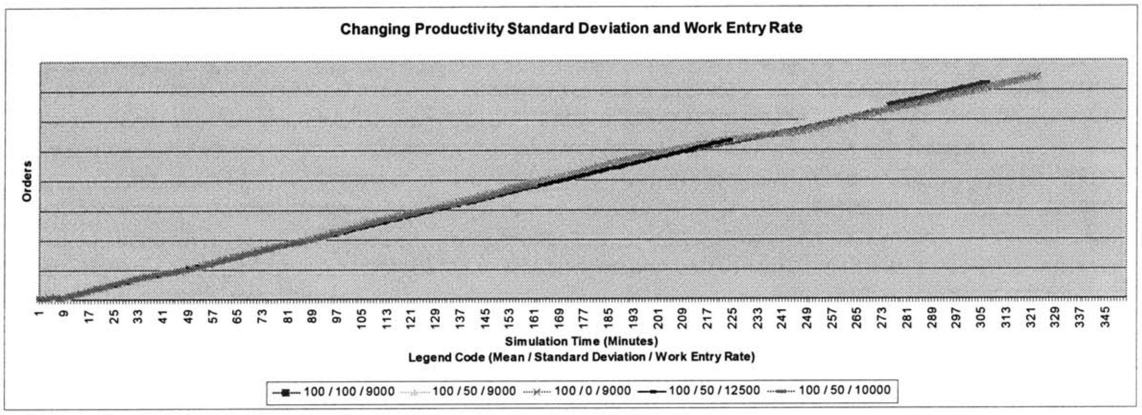


Figure 3.13 – Changing Productivity Standard Deviation and Work Entry Rate

The first observation that jumps out of this graphical presentation is the fact that changing the standard deviation and the work entry rate have virtually zero impact on the throughput. This is the case for productivity standard deviation because, as long as the system is stable and not losing inventory, over an extended period of time the throughput will be the same regardless of the variation. In addition, as long as the work entry rate is not the bottleneck, it shouldn't have any effect on throughput. However, as I will show later, these variables do have a dramatic impact on the buffer stability throughout the system.

3.4.3 Throughput Results as a Function of Changing Arrival Rate and Productivity Mean

The second analysis that I conducted was looking at the impact of changing the mean productivity for the different process paths. To do this, I compared the throughput results for the base scenario with experiments 3, 4, 5, and 6; the graphical comparison is below. Because I was changing the mean productivities in these experiments, I calculated the theoretical throughput of the system based on average productivity to come up with the appropriate work entry rate for the experiment.

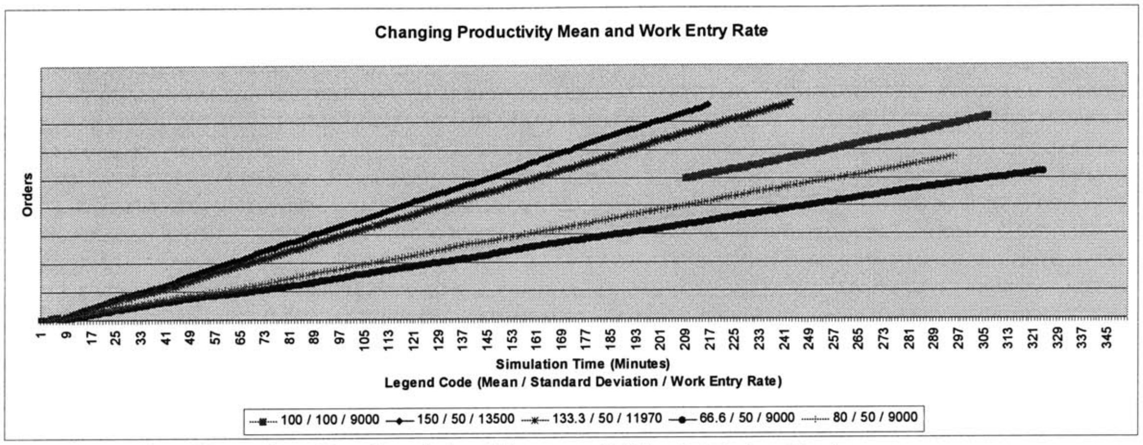


Figure 3.14 – Changing Productivity Mean and Work Entry Rate

It is clear from the graph above that changing the average productivity of the processes has a direct impact on the actual throughput of the system.

Now we have confirmed three things:

- Increasing the productivity mean for the different processes clearly increases system throughput
- Changing the standard deviation of the productivity for the different processes does not impact system throughput.
- Changing the work arrival rate does not affect the system throughput as long as the system doesn't starve.

3.4.4 Cycle Time as a Function of Changing Productivity Standard Deviation

The next analysis that I chose to conduct involved discerning the effect on cycle time of changing the productivity standard deviation. In order to do this analysis, I used Little's Law (Cycle Time = Inventory / Arrival Rate). To ensure that I was conducting a proper comparison, I set everything other than the productivity standard deviation constant. I chose to look at three different experiments: Base, 1 and 2. Because experiment #1 ended at 272 minutes, I took the data from 272 minutes for each of the experiments to develop the table below.

Experiment	St Dev (% of Actual)	Throughput (% of Base)	Arrival Rate	WIP (% of Base)	Time	Cycle Time
Base	100%	100%	9000	100%	272	3.91
Exp # 1	50%	102%	9000	105%	272	4.13
Exp # 2	0	98%	9000	107%	272	4.19

Table 3.5 – Cycle time by changing productivity standard deviation

Looking at these results, it is clear that the standard deviation does not impact the cycle time significantly. In the world of Amazon's fulfillment center, 20 minutes worth of cycle time difference is insignificant. However, it is interesting to note that the lowest value for cycle time and inventory comes from the base experiment – where the standard deviation is actually the highest.

3.4.5 Cycle Time as a Function of Changing Productivity Mean

Realizing that changing the productivity standard deviation had little impact on cycle time, I wanted to see if there was a correlation between productivity mean and cycle time. In order to do this, I again used Little's Law to calculate the cycle time. To ensure a proper comparison, I set everything other than the productivity mean and the arrival rate constant. I chose to look at three different experiments: 1, 3 and 4. Because experiment #3 ended at 215 minutes, I took data from 215 minutes for each of the experiments to develop the table below.

Experiment	Mean (% of Actual)	Throughput (% of Exp #1)	Arrival Rate	WIP (% of Exp #1)	Time	Cycle Time
Exp # 1	100%	100%	9000	100%	215	3.71
Exp # 3	150%	146%	13500	118%	215	2.91
Exp # 4	133%	132%	11970	113%	215	3.15

Table 3.6 – Cycle time by changing productivity mean

Here you see a dramatic impact on cycle time. By increasing the mean productivity, the facility experiences a significant decline in cycle time – over 20% with experiment #3 and over 15% with experiment #4. However, you also see an increase in inventory in the system.

3.4.6 Base Scenario – Amazon's Actual Productivity Mean and Standard Deviation

Given the insights above, I wanted to take a look at the inventory graphs for some of the experiments above. To start with, I wanted to look at the inventory graph for Amazon's actual productivity numbers and work entry rate. This following graph is for the base experiment.

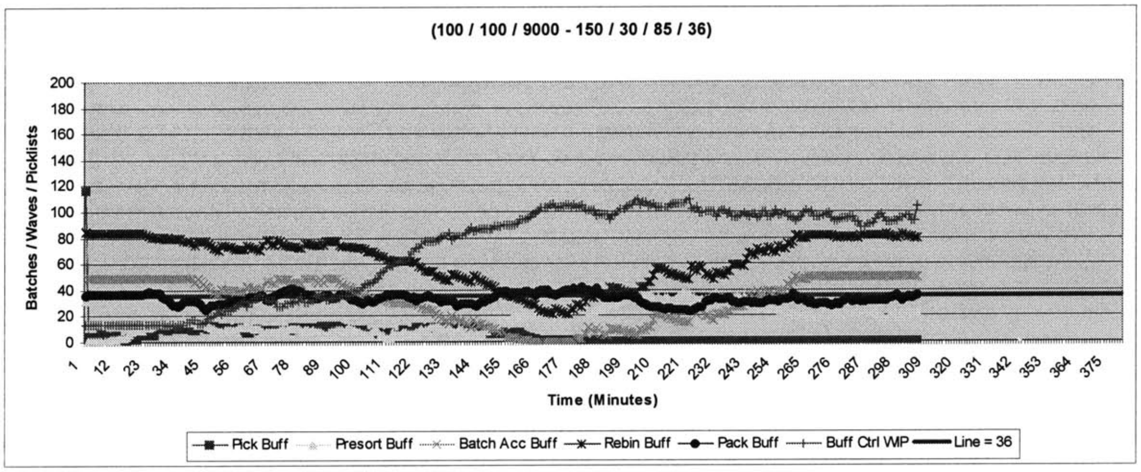


Figure 3.15 – Productivity Base Experiment

Within this scenario, the buffers are actually remarkably stable over time. This is important as wildly fluctuating buffers in the system are extremely difficult to manage. The packing buffer looks good and seems to be trending right along the red line, which translates to maximum throughput for the system. However, the re-bin buffer and the batch accumulator are both full at the end of the model, and the buffer control inventory is also very high – at around 100 batches. Lots of buffer control inventory is important because there is really no inventory management process for the items that are in the buffer control queue. In essence, this equates to a lot of inventory piling up around the pre-sort stations (already processed by pre-sort) that still needs to enter into the conveyor system.

The next experiment that I would like to show is experiment #1, which has 50% of the actual productivity standard deviation.

3.4.7 Productivity Experiment # 1: Amazon's Mean Productivity with 50% of Actual Standard Deviation

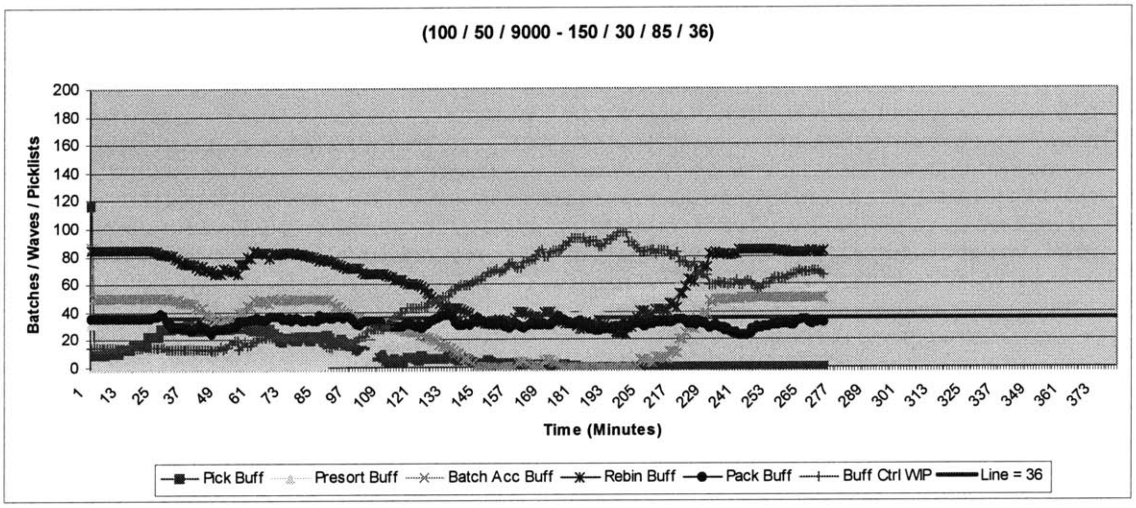


Figure 3.16 – Productivity Experiment #1

Comparing this experiment to the previous one, we see clear gains by cutting the standard deviation in half. While this does not seem to affect throughput (the brown buffer line still trends right along the red line), it seems to be very stable with the inventory at buffer control holding at around 60 batches compared to 100 batches from the base scenario. However, again we see the re-bin buffer and the batch accumulator buffer reaching peak capacity at 226 minutes.

The next experiment that I would like to show is experiment #2, which has no variation in the productivity values because the standard deviation is set to zero.

3.4.8 Experiment #2: Amazon's Mean Productivity with Zero Standard Deviation

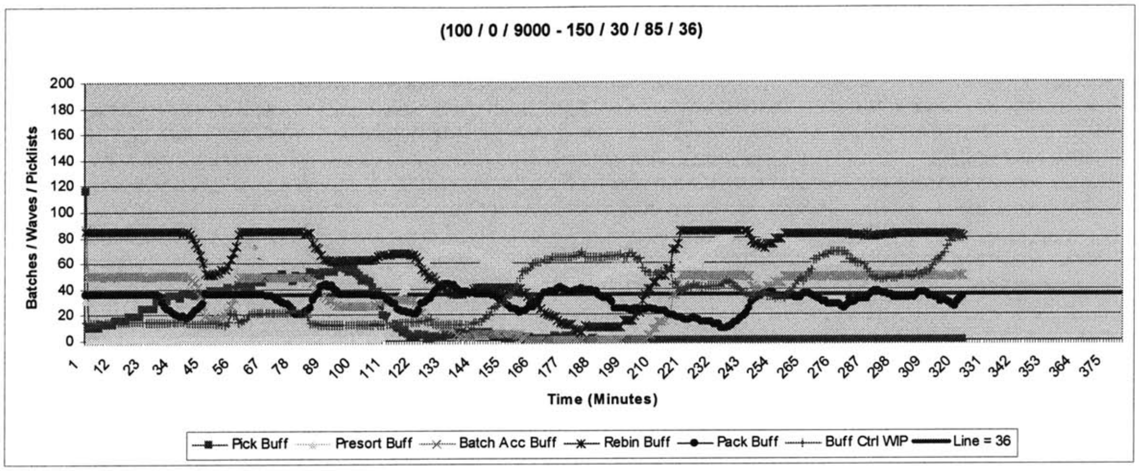


Figure 3.17 – Productivity Experiment #2

The first thing that you notice is wildly erratic fluctuations in the buffers. These fluctuations are less pronounced as you go along because of the variation introduced through order size and, subsequently batch and wave size. From 160 to 211 minutes, the re-bin buffer is dry – this equates directly to a bunch of re-bin operators standing around. In addition, for a large portion of the time, the packing buffer is below 36 batches, again equating to a bunch of packing operators standing around. In the beginning of the model, the pre-sort buffer is dry – also a bunch of people standing around. In a real world situation, this would be a disaster. However, it is also interesting to note that the queue at buffer control, while erratic, averages much lower than the other graphs at around 50 batches.

3.4.9 Productivity Experiment #5: Optimal Buffers

The final graph that I want to show is experiments #5. This graph represents what I feel would be the optimum buffer inventories over time. It has very low inventory levels relative to similar experiments, and the packing buffer still trends in the vicinity of 36. However, one of the reasons for this differential may be the reduced productivity means. This graph may represent a stretched out version of the other graphs, as they are operating at substantially reduced throughput numbers.

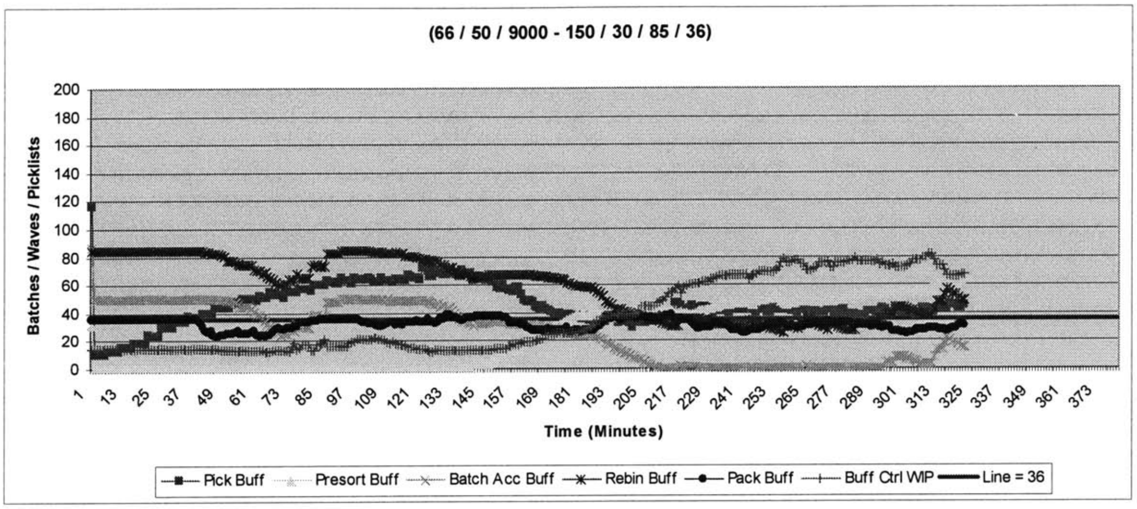


Figure 3.18 – Productivity Experiment #5

The reason that I feel this experiment represents optimal buffers is that it would be the easiest to manage on the operations floor. All of the workers across the system are fully utilized. The re-bin and the batch accumulator buffers are not filled to capacity, and the buffer control queue is stable around 70 batches. Finally, the throughput looks very good as the packing buffer fluctuates right around the red line, which equates to all of the packing stations working for the entire period.

3.4.10 Conclusions

In conclusion, I believe the following findings are important:

- As expected, the main driver of throughput is the productivity mean across the different process paths. Creating or enhancing incentives and training programs to increase the productivity means is a valid goal.
- The work arrival rate and the productivity standard deviation do not substantially affect system throughput – except in the case where the work arrival rate would drop so low that it would starve the system of work.
- The main driver of cycle time is the productivity mean and not the productivity standard deviation. In fact, decreasing the productivity standard deviation seems to have a slightly adverse effect on system cycle time.
- Decreasing the productivity standard deviation yields more manageable buffers for the operations team, until it falls to a point that creates wildly fluctuating buffers. However, in the real system, decreasing productivity standard deviation even 50% would be a stretch goal. In other words, I don't believe that the facility needs to worry about decreasing the standard deviation too far.
- The main driver of inventory in front of buffer control is the productivity standard deviation. As the standard deviation falls, the inventory at buffer control falls dramatically.

As a final recommendation, I would create or enhance human resource policies to focus as much as possible on increasing the mean productivities across all process paths. In addition, if the size of the inventory queue in front of buffer control becomes an issue, I would develop policies to decrease the productivity standard deviation as much as possible.

Chapter 4: Methodology

4.1 Case for Discrete Event Simulation

The two primary drivers for discrete event simulation were the seasonal nature of Amazon's business environment and the extensive variation experienced across every operation within the fulfillment center.

One of the unique aspects of this project involved the seasonal nature of Amazon's business environment. For nine months of the year, Amazon's operations in the UK ran relatively smoothly, and, more importantly, there was never a question of whether or not they could meet the customer promise. During the holiday months of October, November and December, when it was increasingly important to meet the customer promise, it became far more difficult. During these months the facility was operating at more than twice the output of the prior months. In addition, training the management team for the holiday period was virtually impossible during the other nine months of the year. As a result, developing a simulation tool that could replicate the holiday period and uncover key lessons ahead of the holiday rush would be indispensable to the senior management team of the facility.

The second driver for discrete event simulation came from the extensive variation experienced throughout the system. One of the advantages of discrete event simulation was the ability to model the system closer to reality, sampling from different distributions at each step of the process. In the end, through discrete event simulation, Amazon could replicate a shift's worth of production incorporating all of the variation that existed in the real system.

4.2 Case for SIMUL8

After several brainstorming sessions with the management team and my advisors, I made the decision to pursue a replication of the facility using the discrete event simulation program SIMUL8. This decision was made rather quickly, without exploring the capabilities of other discrete event simulation software. One reason for this is that I had been exposed to the SIMUL8 program during my operations studies at MIT. In hindsight, I would have spent a far greater amount of time doing program benchmarking and program selection.

SIMUL8 offers many different capabilities that are essential to replicating Amazon's operations. These capabilities include: the ability to sample from different probability distributions, the ability to incorporate the simple and familiar user interface of Microsoft Excel, and the ability to run trials that will incorporate confidence intervals into the final results. The ability to sample from different distributions makes it possible to sample productivity distributions to determine the processing time within each process path according to the number of items per batch and per pick-wave and to allow for the variation in productivities across the entire base of employees, both permanent and temporary. The ability to incorporate Excel interfaces means that the end model user does not have to be savvy enough to understand the programming language of Visual Logic or the different simulation structures within SIMUL8. Finally, the ability to run trials and develop confidence intervals for results means that the results will be accurate and more widely accepted, believed, and incorporated into system changes within the FC.

4.3 Preparation for Model Build

At the beginning of my time at Amazon, I really wanted to develop a feel for the operations. I accomplished this by spending a couple of days in each of the departments on the operations floor, working as an associate. This two week experience proved invaluable as I sought to gain a better understanding of the overall system. In addition, it provided a significant network of line associates and middle managers to whom I could turn for answers to questions further down the line.

My next step involved creating flow diagrams to ensure that I understood the subtleties involved in each of the individual processes. This step really forced me to understand all aspects of the processes and to realize the implications of upstream mistakes and the needs of downstream customers.

Simultaneously, I started the learning curve for SIMUL8. This learning curve was very long. As soon as you get past the simplest of models, one must understand and code using the programming language of Visual Logic that underlies the entire SIMUL8 platform. In order to learn this programming language, I bought the book, Learning SIMUL8: The Complete Guide, read it, and worked through many of the example models that it contains.

4.4 Model Description

In this section, I will describe the SIMUL8 model in detail including sections on the model structure, the model logic, and the data inputs.

4.4.1 Model Structure

Within the model structure section, I will go over each individual process path and where it lies within the overall model. In addition, I will give a brief description of what actually happens within the model at each new process. To begin with, I wanted to show a snapshot of the SIMUL8 window with all of the individual work stations. As you can see below, each of the individual processes are highlighted and identified by text in the picture.

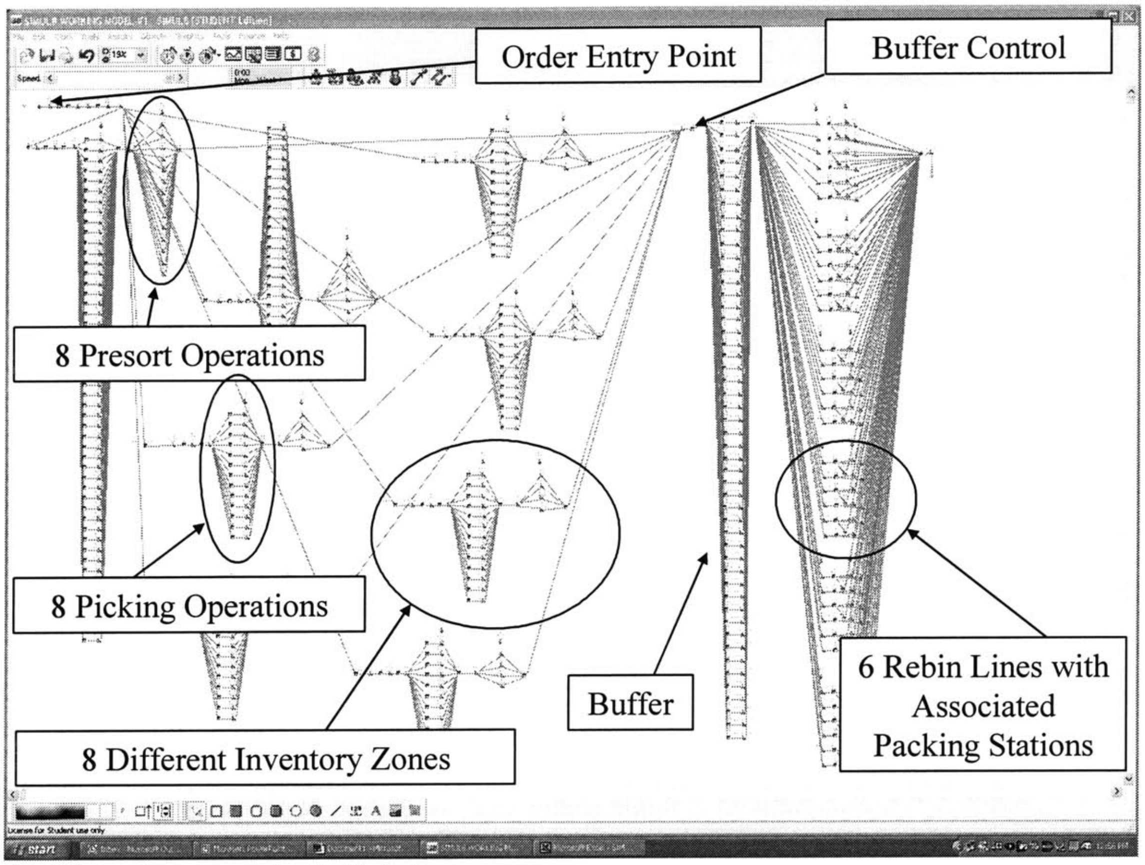


Figure 4.1 - Overview of the Entire SIMUL8 Model

4.4.1.1 Order Entry

The order entry process is relatively simple. There is a throttle on the work entry point that dictates the order arrival rate 'wep multi,' and this rate is input by the user into the Excel spreadsheet. Then the inventory processes through the collating process. First it is grouped into batches based on the accumulation of orders, then it is aggregated into waves based on the accumulation of batches. Finally, the waves are replicated from individual orders into the actual pieces of inventory at Amazon, and the orders are then dispersed across the different inventory zones, based upon a percentage routing out discipline set by the user.

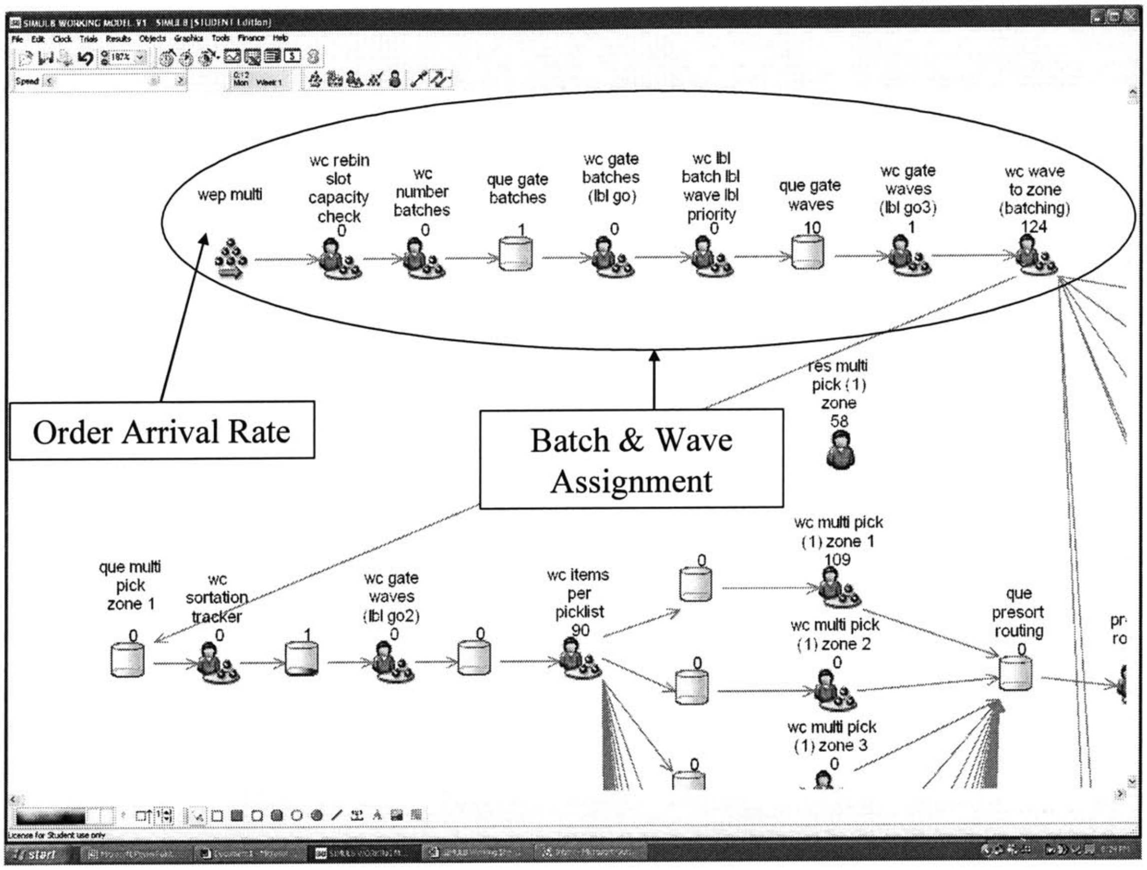
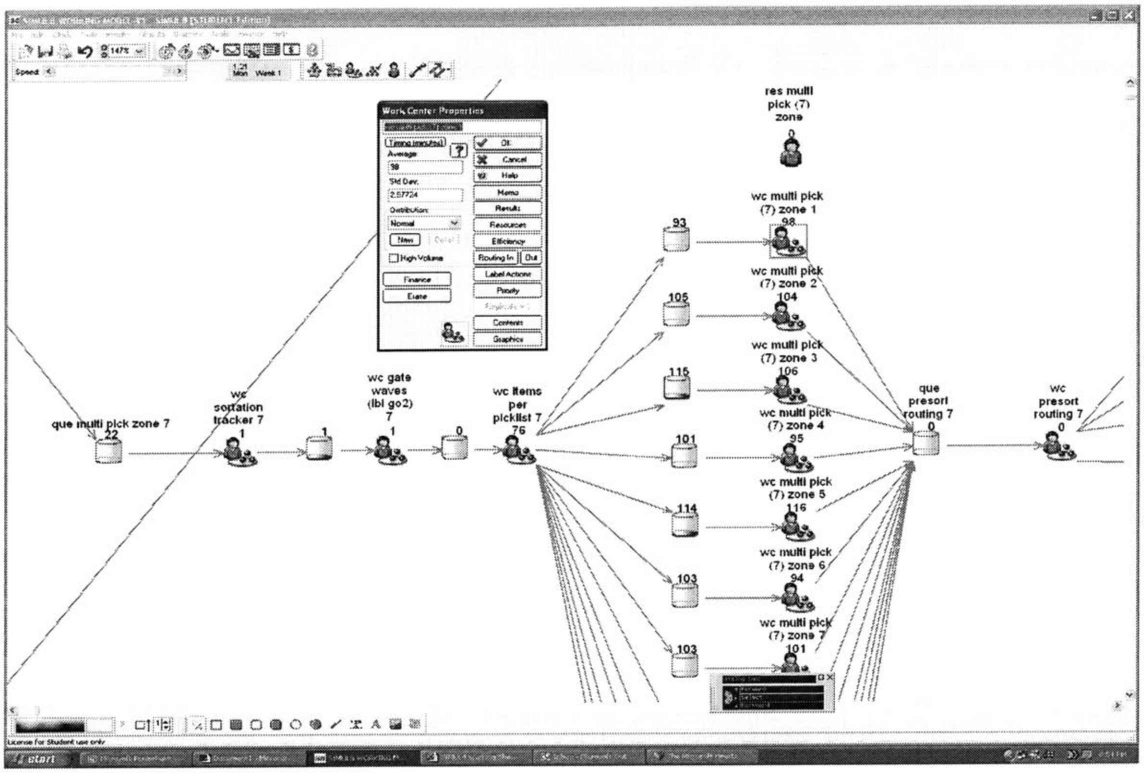

wep multi
wc rebin slot capacity check 0
wc number batches 0
que gate batches 1
wc gate batches (lbl go) 0
wc lbl batch lbl wave lbl priority 0
que gate waves 10
wc gate waves (lbl go3) 1
wc wave to zone (batching) 124
res multi pick (1) zone 58
Order Arrival Rate
Batch & Wave Assignment
que multi pick zone 1 0
wc sortation tracker 0
1
wc gate waves (lbl go2) 0
0
wc items per picklist 90
0
wc multi pick (1) zone 1 109
wc multi pick (1) zone 2 0
0
wc multi pick (1) zone 3 0
0
que presort routing 0

Figure 4.2 – Order Entry and Collating

4.4.1.2 Picking Stations

The picking stations are set within the model based on the actual number of picking stations across the Amazon UK facility. At picking, the model samples from a normal distribution to attain the processing time based on the number of orders within the pick-wave. The inventory processing unit at picking is a pick-wave that includes the number of items within a specific inventory zone from the wave being processed.



Work Center Properties

Timing (minutes)

Average: 38

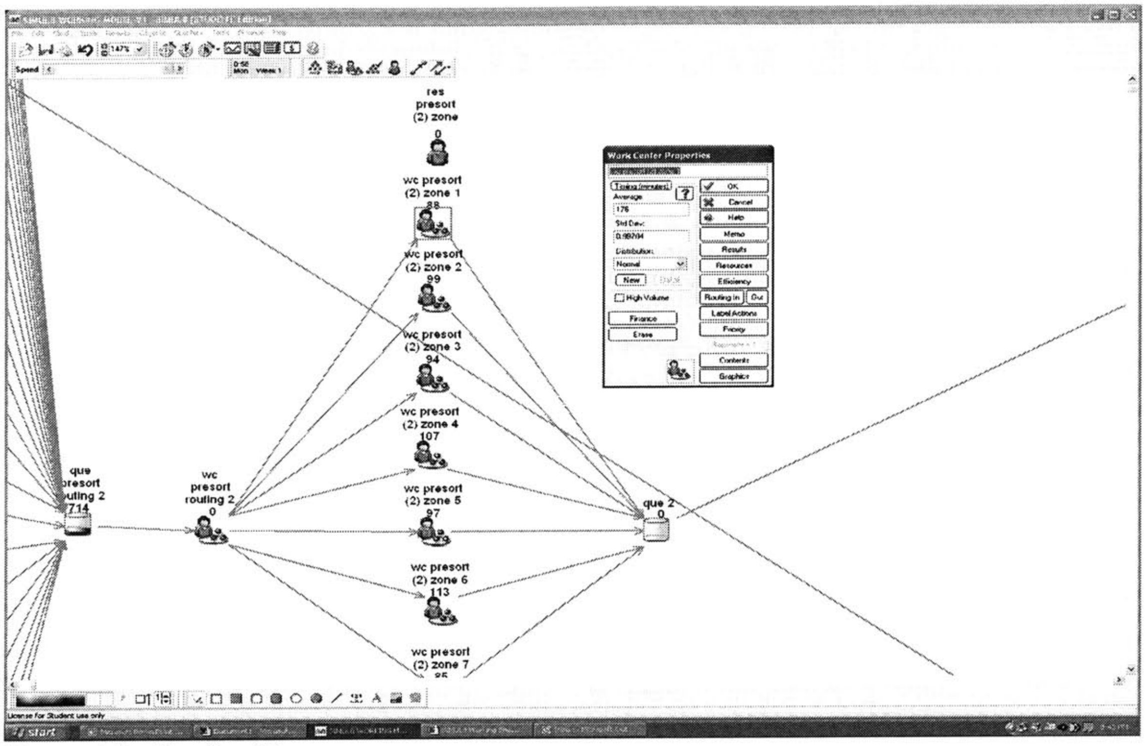
Std Dev: 2.57724

Distribution: Normal

Figure 4.3 – Wave Picking Process

4.4.1.3 Pre-sort Stations

The pre-sort stations are set within the model based on the actual number of pre-sort stations across the Amazon UK facility. At pre-sort, the model samples from a normal distribution to attain the processing time based on the number of orders within the pick-wave. The inventory processing unit at pre-sort is a pick-wave that includes the number of items within a specific inventory zone from the wave being processed.

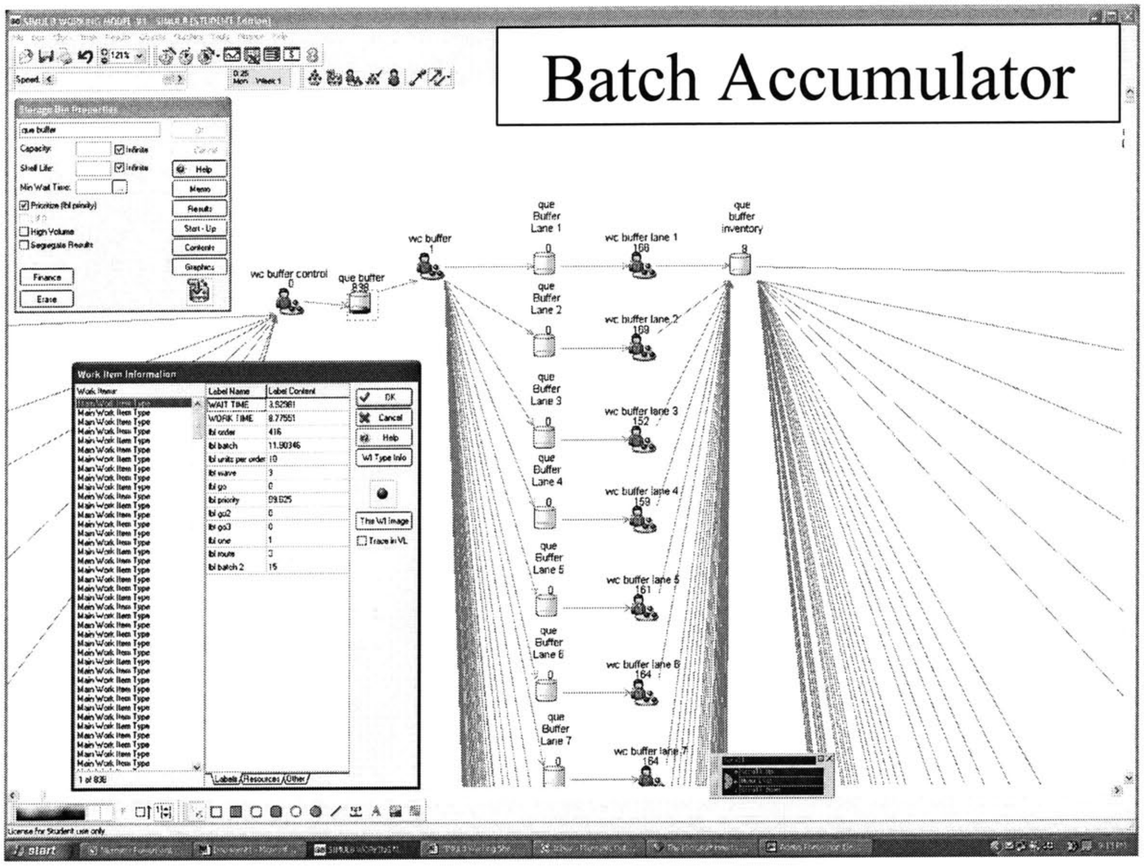


Average
176
Std Dev:
0.99704
Distribution:
Normal
□ High Volume

Figure 4.4 – Pre-Sort Process

4.4.1.4 Buffer Control and the Batch Accumulator

At buffer control, the inventory from all of the pre-sort zones across the model is collected. As complete waves become available at buffer control, the inventory is released to the downstream batch accumulator in batches. At the batch accumulator, the batch accumulator capacity is check, and, if there is room, the inventory processes through to the re-bin queue. As the model is currently set up, there is no wait time to process through the batch accumulator to the re-bin queue.



Storage Bin Properties

que buffer

Capacity: ☑ Infinite

Shelf Life: ☑ Infinite

Min Wait Time:

☑ Prioritize (lbl priority)

□ High Volume

□ Segregate Results

Work Item Information

Label Name	Label Content
WAIT TIME	3.52981
WORK TIME	8.77551
lbl order	416
lbl batch	11.90346
lbl units per order	10
lbl wave	3
lbl go	0
lbl priority	99.625
lbl go2	0
lbl go3	0
lbl one	1
lbl route	3
lbl batch 2	15

1 of 838

□ Trace in VL

Figure 4.5 – Buffer Control and the Batch Accumulator

4.4.1.5 Re-bin and Packing Stations

The re-bin stations are set within the model based on the actual number of re-bin stations across the Amazon UK facility. At re-bin, the model samples from a normal distribution to attain the processing time based on the number of orders within the batch. The inventory processing unit at re-bin is a batch that has been aggregated by buffer control and released by the batch accumulator.

The number of packing stations within the simulation model has also been set according to the actual number of packing stations within the UK facility. At packing, the model pulls complete orders out of re-bin and processes these orders according to the number of items per order. Finally, at packing, the model reassembles the order into an individual item.

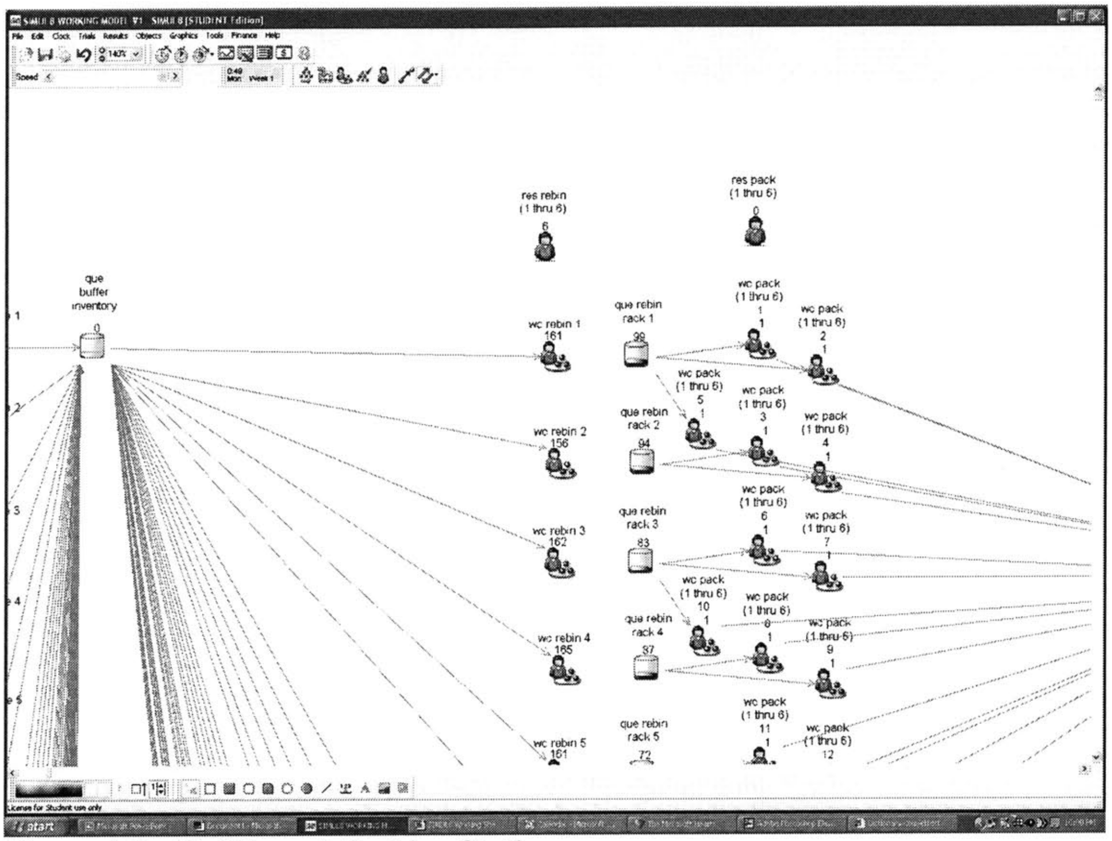


Figure 4.6 – Re-Bin and Packing Stations

4.4.2 Model Logic

The biggest obstacle to replicating Amazon's UK operations involved figuring out how to get batches, waves, and pick-lists to process simultaneously within the system. This was made more difficult because, at each step in the process, the model needed to process on a different level of aggregation. In the beginning of the model at collating, orders needed to process through as complete batches and get aggregated into waves. At the next step, the model needed to separate the wave by inventory zone – at the level of individual items. Then, after picking and pre-sort within each of the inventory zones, the model needed to aggregate from all of the inventory zones back into batches, but only after a specific wave was complete across the system. Finally, at packing, the model need to process items based on the original order size. Compounding the difficulty of this tracking was the fact that every batch, wave, and pick-list was composed of a different number of items.

I've included several snapshots of the SIMUL8 model and annotated where each of the following challenges occurs within the model. The labels on the snapshot pertain to the section numbers below.

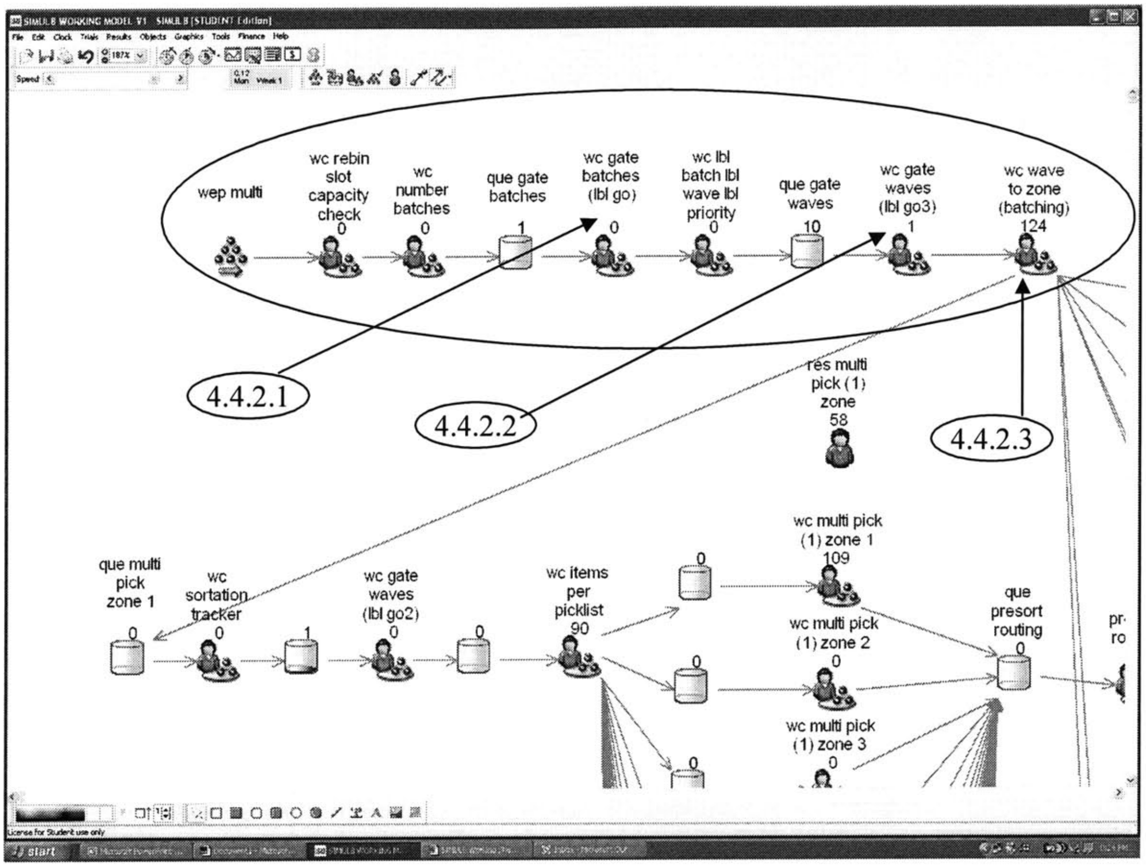

SIMUL8 WORKING MODEL V1 SIMUL8 [STUDENT Edition]
File Edit Clock Trials Results Objects Graphics Tools Finance Help
Speed
wep multi
wc rebin slot capacity check 0
wc number batches 0
que gate batches 1
wc gate batches (lbl go) 0
wc lbl batch lbl wave lbl priority 0
que gate waves 10
wc gate waves (lbl go3) 1
wc wave to zone (batching) 124
4.4.2.1
4.4.2.2
4.4.2.3
res multi pick (1) zone 58
que multi pick zone 1 0
wc sortation tracker 0
1
wc gate waves (lbl go2) 0
0
wc items per picklist 90
0
wc multi pick (1) zone 1 109
0
wc multi pick (1) zone 2 0
0
wc multi pick (1) zone 3 0
que presort routing 0
pr
ro
License for Student use only
start

Figure 4.7 Model Logic Locations 1

4.4.2.1 Aggregating Orders into Batches

Batch size is really driven by the downstream re-bin infrastructure. Each re-bin station on the operations floor at Amazon is comprised of a specific number of slots, each of which has a limited capacity to hold inventory. As orders arrive into my model, they are assigned an order size based off the order composition data above. The order size then determines how many slots each order will take within the re-bin station. Once enough orders have arrived to create a batch and fill up the downstream re-bin station, the batch is complete.

In the model, I accomplish this feat by writing the order size to a spreadsheet, using programming logic to determine the slot charge for each order and keeping track of the cumulative slot charge to determine when a batch is complete. The last order in the batch gets a label value of one (lbl go) to signify that it is the final order in the batch. I then use label batching to enable the model to process the batch as one complete unit.

4.4.2.2 Aggregating Batches into Waves

I use a similar process to aggregate batches into waves. As waves pass through the batch complete work center, they move on to the wave complete work center. Here, the model then looks for a second gating label (lbl go 3) to signify the last order in the final batch of the wave. This process is managed through the work centers label batching option. All of these labels are managed through analyzing data that is input into SIMUL8's spreadsheets. As SIMUL8's spreadsheets become unstable at anything over several thousand lines, it is necessary to periodically clear out the spreadsheet and start over. I do this by dumping several hundred lines of pertinent data into a different spreadsheet, clearing the first spreadsheet, and then dumping the required data back into the first spreadsheet.

When working with SIMUL8's spreadsheets, several tricks worth noting at this point involve the following:

If you need to analyze data on the spreadsheets from one order to the next, a handy way to accomplish this is by adding a line to the spreadsheet with each new order. In other words, you end up with a spreadsheet that scrolls down with each new entry. This also enables you to clear the spreadsheet as mentioned above. Another trick solves the challenge of inventory processing through the system too quickly. You can prevent this from happening by ensuring that there is always one item in the storage bin prior to the work center. This is accomplished by the following code – which I used in many places throughout my model:

```
VL SECTION: wc gate waves (lbl go3) Route-In Before Logic
 IF que gate waves.Count Contents  =  1
  Block Current Routing
```

4.4.2.3 Allocating Waves to Specific Inventory Zones

This step involved the real process of collating more than any other step. In the beginning of the modeling process, I just setup the work center to be a simple route out percent that could be imported from a spreadsheet. This setup caused a huge accumulation of inventory in the zones with fewer processing stations. In the end, I ended up programming a routine that actually collated, by pulling some 5 batch waves from 2 zones, some from 4 zones, some from the front side of the picking stacks, and some from the back side of the picking stacks. In the end, this part of the process very accurately modeled the overall system.

One important distinction still exists between the actual system and my model. At Amazon, the inventory is in the stacks, and the computer pulls the inventory from the inventory zones where it wants to pick. In my model, the user determines where the picks should originate, and then sends the inventory to those zones.

Finally, the last step in the wave allocation process involves replicating the orders into individual items. Because the waves will be split across different inventory zones, the waves cannot be processed as orders. Instead, they are replicated based on the item per order label and then split across the pertinent inventory zones.

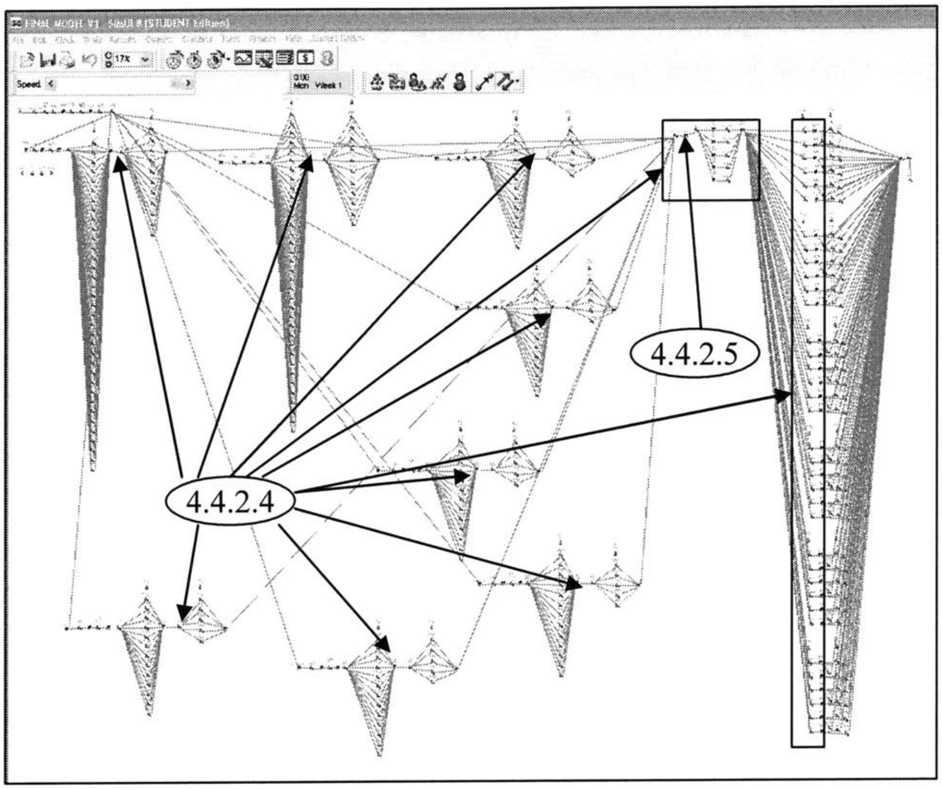


Figure 4.8 – Model Logic Locations 2

4.4.2.4 Processing Variable Sized Batches and Waves Downstream

This step stumped me for quite some time. As the inventory processes through the system, somehow you must bring it back together as waves and batches downstream. However, this is extremely difficult as each of the batches and waves are variable sizes because the order size varies with each arrival. To combat this, I counted the inventory as it flowed through the system at different points. As a result, in the spreadsheet, iss label control, I have data that includes the items per batch and the items per wave and items per pick-wave within each of the different inventory zones. By having this accumulated information, I can find out when an entire batch, wave, or wave-pick-list has processed through a part of the system. In addition, this information allows me to reach out into a queue, find out which batch is first in the queue and collect the exact amount of items within that batch. This is the process used at both the batch accumulator and the individual re-bin stations.

4.4.2.5 Aggregating Batches from Inventory Zones

This section also initially involved a simplifying assumption. When I first built the model, I forced the inventory to process through the batch accumulator chronologically. However, after several trials, I determined that this method was starving the downstream processes of re-bin and pack. As a result, I developed some visual logic code to reach into the queue and look to see which batches were complete. As a result, my model has the ability to process batches out of chronological order to ensure that the downstream processes of re-bin and pack maintain sufficient inventory stock.

4.4.3 Model Data

In the case of Amazon UK, there is a significant amount of data being collected on a daily, weekly and quarterly basis. As a result, collecting data was a relatively painless portion of the overall research project

As I developed modules of my simulation model, the type of data that I needed started to become clear. Furthermore, as I arrived at some of the assumptions in my model, the need for some data disappeared. In the end, I found that I needed much less data than I would ever have anticipated, and most of that data was readily available due to the data collection systems that Amazon had already put in place.

I've included a snapshot of the SIMUL8 model and annotated where each of the following data sets is used within the model. The labels on the snapshot pertain to the section numbers below.

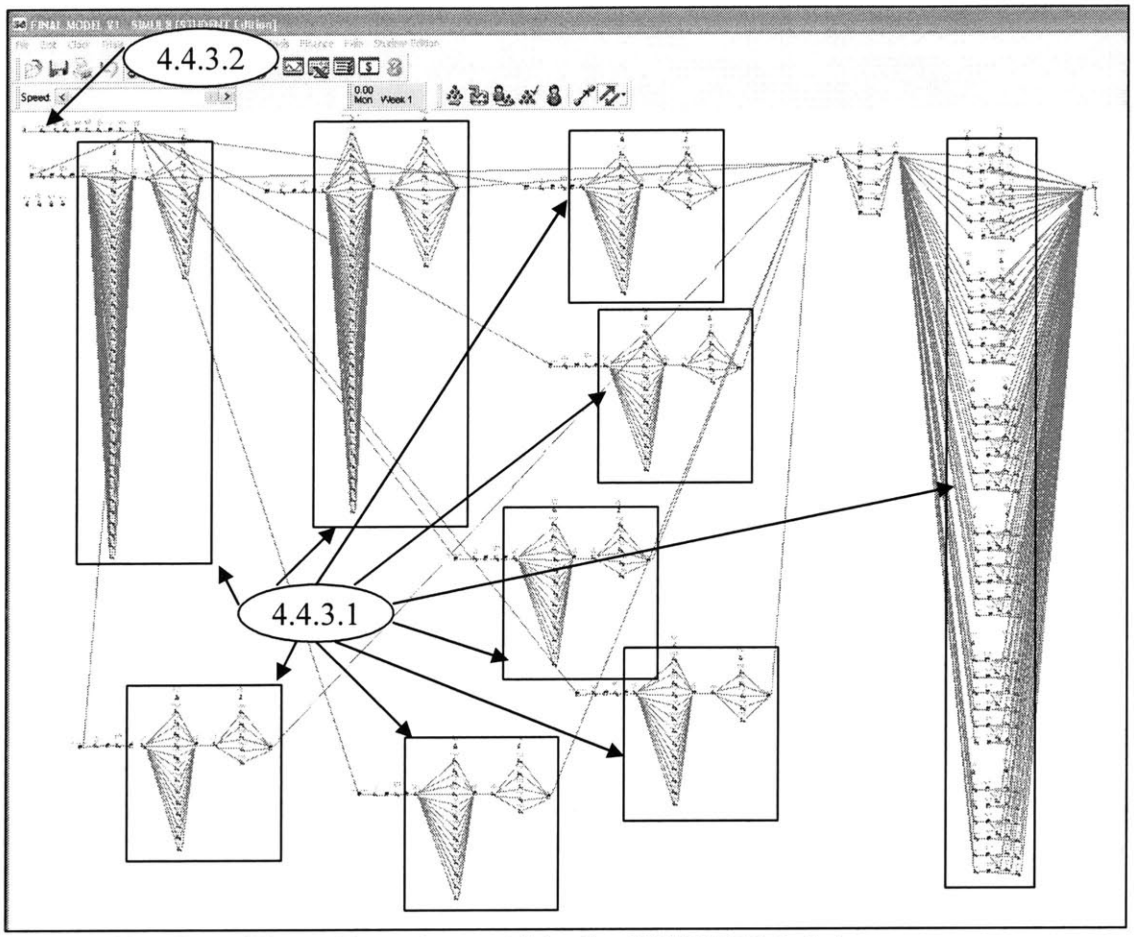


Figure 4.9 – Data Usage Locations within the SIMUL8 Model

4.4.3.1 Mean and Standard Deviation of Process Paths

As one of the main inputs to my simulation model, it was extremely important to get accurate data for the mean and standard deviation of the productivities for the individual process paths. Luckily, Amazon was also concerned with this metric, and, as a result, it was very easy to get pertinent data. I took data from the end of November through the end of December and ensured that it had already been verified by the operations team as accurate. I then sorted the data and threw out any outlying data that could not possibly have been accurate. Finally, I set up a pivot table within Microsoft Excel to arrive at the means and standard deviations of productivity for each of the individual process path. The final output data is included below.

This data set is used to derive process times at every work station throughout the entire model. It is used in picking, pre-sort, re-bin, and packing and is variable based on the number of items in the pick-wave or batch currently being processed.

Task	Statistical Measure	Temp Agency A	Temp Agency B	Amazon Associate	Grand Total
Pack Multis	Sample Size	Amazon Proprietary Information			
	Mean (Units/Hour)				
	StdDev (Units/Hour)				
	Average (Min/Unit)				
	StdDev (Min/Unit)				
Pick Multis	Sample Size				
	Mean (Units/Hour)				
	StdDev (Units/Hour)				
	Average (Min/Unit)				
	StdDev (Min/Unit)				
Presort	Sample Size				
	Mean (Units/Hour)				
	StdDev (Units/Hour)				
	Average (Min/Unit)				
	StdDev (Min/Unit)				
Rebin	Sample Size				
	Mean (Units/Hour)				
	StdDev (Units/Hour)				
	Average (Min/Unit)				
	StdDev (Min/Unit)				

Table 4.1 – Mean and Standard Deviation of Productivity by Process Path

A final note on data collection is that even with the robust metric tracking systems that Amazon UK employs, I am still somewhat skeptical of the underlying data. The reason for this doubt is that many of the individual data entries from which these results were calculated were the same. Given that the individual data entries belonged to unique workers, the probability of 10 workers having the same productivity on the same day is almost impossible. However, given my experience within operations in the UK, I believe that the data above is realistic and in the ballpark, even if it is not perfect.

4.4.3.2 Order Composition

I started with a set of data for order size that comprised a total of over 5 million units. I broke this data set into the amount of single item orders and the amount of multiple item orders. Then I analyzed the percent items within each bucket. The chart below shows the percent of orders within each bucket relative to all the multiple item orders. One of the assumptions in my model is that there are no order sizes over 20 units. As a result, 0.1 % of the multiple item orders have 20 or more units within my model. This represents actual orders from sizes from 19 to 100 units. The true values for the numbers in the chart below have disguised.

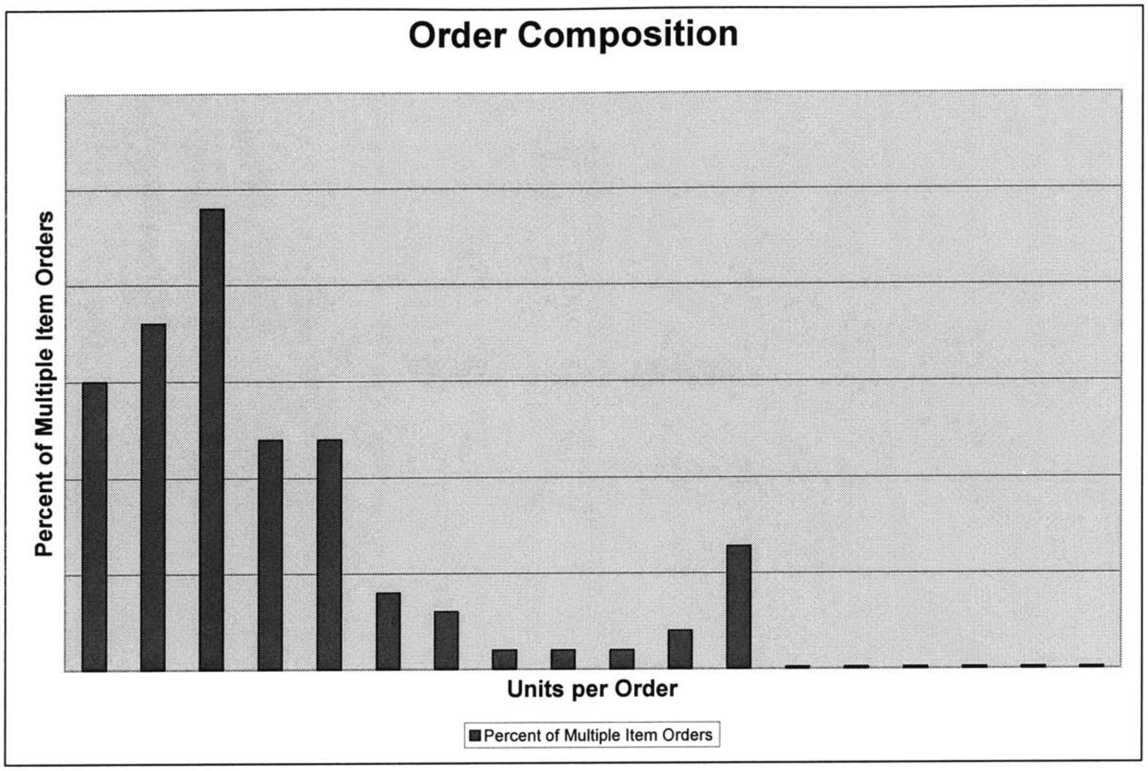


Figure 4.10 – Order Composition Graph: The true values above have been disguised.

4.5 Model Validation

For this simulation model and thesis, I really focused on two different types of validation: accurate replication of inventory flow and accurate replication of results. Relative to inventory flow, it is extremely important to ensure that the manner in which inventory flows through the model parallels the way that it actually flows through the real system. Relative to the accurate replication of results, it is crucial to see whether the results of the model actually reflect real time results from the distribution center.

Relative to inventory flow validation, this model functions very well. Inventory actually moves through the system in a manner that clearly parallels the actual facility. The model incorporates very similar logic to arrive at batch size and wave size and enables the user to dictate the order composition, the number of employees, the collating procedures, and the actual processing time distributions. In addition, the inventory flows through the system exactly as it happens in real life, with the exception of the assumptions highlighted below. It is important to note that this is an entirely qualitative validation.

In order to do the results validation, I chose to focus on the throughput results because I had actual data for these results. As a result, because I didn't have actual data for cycle time results, the cycle time predictions generated by my model and discussed in the experiments sections above have not been validated.

For the throughput validation, I started with the actual number of multiple item units that Amazon wants to process during a 10 hour shift on their peak day. While Amazon strove to meet this figure (let's call it X), their 10-hour shift output at the beginning of December fluctuated between 50-90% of this desired capacity. As a result, I sought to achieve an output in this vicinity through my simulation. In addition, I ensured that the model conditions replicated the actual conditions at Amazon during their peak day trial. This included not only the amount of work stations available across the different process paths, but also the startup inventory in the process path queues, the inventory arrival rate, and the employee staffing.

Due to program instabilities and the complexity of this model, my simulation will only run for several hours of simulation time before crashing. As a result, I have sought validation through running several iterations of one hour's worth of simulation time. The resulting output is exhibited below. This table shows the five different trials used for the validation. The first line in the table notes the trial for each set of data, and it also includes the mean and standard deviation titles. The next line notes the amount of multiple items processed for each trail. This data is normalized relative to the initial trial. In other words, the percent change, positive or negative, in number of multiple items processed is noted for each of the subsequent trials. On the right side of the table, you have the mean and standard deviation for the set of five trials. The mean is a percent change from trial #1 in terms of multiple items, and the standard deviation is a percent change from trial #1 in terms of multiple items.

	Trial #1	Trial #2	Trial #3	Trial #4	Trial #5	Mean	St Dev (% of Trial #1)
Multiple Items Processed (%Δ from Trial #1)	0.00%	-2.12%	-0.13%	-0.07%	+2.26%	-0.01%	+/- 1.55%

Table 4.2 – Validation Data

The next step involved constructing confidence intervals. I chose to use a 95% confidence interval.

With 95% confidence, the single hour throughput is between +/- 1.52% of the demonstrated mean. This value falls within Amazon's 50% – 90% range of X. As a result, I concluded that this model was an accurate representation of Amazon's UK fulfillment center.

Finally, I chose to analyze the relative difference between the output predicted in my simulation experiment and the output for the real system.[3] The relative difference was 14.8%.

[3] Relative Difference = (Simulated Throughput – Actual Throughput) / Actual Throughput

One caveat to this conclusion involves the time period used for validation. Because I only used a single hour worth of production, only the inventory present at packing processed through the model exit point. As a result, this conclusion is really a validation that the packing capacity is reflective of what is seen in the actual system. However, through all of my experiments, the packing capacity is fully utilized over 95% of the time. As a result, I feel comfortable that this validation still holds true.

4.6 Model Assumptions

The assumptions within my model are included below:

- Orders of over 21 items are not included in the model. This assumption can be adjusted through additional and simple Visual Logic coding. This assumption was made to save significant visual logic coding for order sizes from 21 to 250. I believe that the impact of this assumption on the model is negligible.

- There are always enough orders in the system. The system will never experience a point when there are insufficient orders to process. This assumption was made to simplify the model and its effect is negligible for any peak period modeling situation.

- The capital infrastructure of the facility is set within the model. I built the simulation to incorporate all of the stations that were present during the holiday period 2003. While it is relatively easy to change this assumption, it does require some knowledge of SIMUL8 and a cursory understanding of Visual Logic. I made this assumption because I couldn't find a simple method, using visual logic, to allow the user to dictate the number of processing stations for every iteration.

- Though SIMUL8 allows for the introduction of process breakdowns for one reason or another, this model does not incorporate the breakdown of any processes throughout the system. In other words, equipment uptime in this model is 100%. I made this assumption to simplify the model. In effect, no downtime just equates to greater system productivity.

- One of the key issues that Amazon faces is problem solving. Any order in the system that encounters a problem is routed to the problem solve department. Within that department, there is a team of people that fixes the problem and then ships the order out to the customer. In this model, all orders are processed perfectly and there is no double processing. Again, this is a simplifying assumption and amounts to greater system productivity.

- The labor force used in this model works 24 hours a day, seven days a week. This assumption means that the in-depth shift schedules and breaks are not taken into account. I initially modeled the shift schedules and breaks, but it quickly became too complicated because of the plethora of different work schedules during peak holiday period at Amazon. Its effect is negligible because the model does not run past a single shift.

4.7 Model Limitations

I have attempted to list the limitations that I have encountered with SIMUL8, Visual Logic, and Discrete Event Simulation below:

- Some of the limitations of SIMUL8 include the fact that no one currently employed by Amazon understands either SIMUL8 or the underlying programming language of Visual Logic. To my knowledge, there is only one employee within all of Amazon that has extensive experience in simulation – and that employee has only worked with AutoMod.[4]

- Because of the nature of Amazon's sortation operations, it is necessary to model individual items within the model. As a result, I am operating at the limitation of SIMUL8's capabilities. These limitations manifest themselves in very slow processing times for the models, the need to use very advanced computing systems, and the fact that the model will crash after several hours of simulation time. I ran all of my analysis on dual Pentium processer workstations with 1 GB of memory.[5] The processor usage tended to be around 50% (1.4 GH) and the memory usage hovered around 650 MB. Even with this amount of computing power, the simulations take between 30 minutes and several hours to run. Finally, I had the ability to work in a computer lab at MIT where I could run multiple simulations simultaneously on different work stations.

- Because of the slow processing speed of this model, it is impossible to run multiple iterations of a single scenario using the trial function within SIMUL8. This results in having to run multiple scenarios and compare the results manually.

- Because of the way that this model seeds the buffers, it is very difficult to get accurate data for cycle time. This is a result of the fact that all the inventory at startup – even if it is located at packing – starts with a time in system of zero. Another detrimental effect of the buffer startup seeding procedure is that the queue at buffer control starts with a value of under 2000 units, and those units are perfectly matched to proceed through the system. As a result, it takes several hours of simulation run time to actually properly seed the buffer control queue and for this queue to reach equilibrium.

4.8 Breadth Versus Depth: Analyzing the Tradeoff between Model Simplification and the Ability to Model Reality and Accurately Answer Key Questions

[4] This employee is another LFM, Brent Beabout, and AutoMod is another discrete event simulation program.

[5] By using the Ctrl, Alt, Delete function within windows, you can bring up the Windows Task Manager and monitor the performance of your computing system. This application allows you to view the microprocessor usage and memory usage in real time.

Working with SIMUL8 has been both an exhilarating and a frustrating experience. While this simulation of Amazon's operations models the fulfillment center very closely, it is remarkably slow. During the development of this simulation, I often struggled with the decision to simplify the model. In fact, throughout the process, I actually did simplify the model several times. However, with every iteration of simplification, I found that I was running up against the same issues that I faced with the more complicated model. In the end, I kept returning to the more complicated model to do my troubleshooting. Finally, after nine months of development, I finally arrived at the system that I had envisioned from the beginning of the process, and I could not have attained similar results through a more simplified version. In fact, some of the insights that I described in chapter 3 would not have been possible with any of the previous versions.

However, it has been a long, difficult and often frustrating experience. In addition, I do feel that a couple of the insights in the results and analysis section could have been accomplished with a simpler model. The best advice that I can give is that the decision of more detail vs. less detail really comes down to the overall goal of the simulation. In this case, the goal was insight into system variation, and, as a result, I believe this insight could not have been attained with a more simplified version.

Chapter 5: Further Areas for Exploration and Knowledge Development

During my six month internship with Amazon.com, there were many times that we discussed different scenarios facing the operation and brainstormed about the proper course of action. While we could come up with ideas for and against the different scenarios, it was extremely difficult to predict what the proper course of action really was. Some of these ideas are included below. With each of the ideas, I have included my thoughts about how to tackle the proposed question using this model

5.1 Impact of a Change in Order Size

Because of web based nature of Amazon's business, the company has the ability to impact the average size of multiple item orders by shaping customer demand through different pricing structures. As a result, it would be interesting to know if there is an optimal order size, relative to their operations capabilities, that would impact the processing cost of orders. In addition, it would be interesting to know the impact on operations of a significant shift either up or down in the multiple item order size.

Using this model, it is possible to experiment with different order compositions reflecting the changing nature of multiple item orders. By simply changing the order composition within the Excel spreadsheet, the user can run different simulations comparing the different order types. In addition, they can keep the random number seed the same to determine the exact difference between scenarios by keeping all other factors the same.

5.2 Effect of Added Capacity

One of the issues that Amazon is currently facing is the ability to process holiday orders. It is crucial to know the effect of added capacity across the system. In other words, knowing where to increase the capital infrastructure and understanding the impact of an additional processing station is vitally important to the operations and engineering team.

During my capacity analysis, I changed several of these capital infrastructure process stations, but a lot more work could be done. Adding work centers can be easily accomplished by highlighting the desired work center and dragging it to a new location while holding the control key. With some of the work centers in the model, some small pieces of the underlying Visual Logic code must also be changed.

5.3 Slowdown in the Buffer Induction

One of the problems that the UK FC faced during the holiday period was the increase in totes flowing into the buffer. As more batches and waves were being processed simultaneously, the buffer would breakdown because of the additional circulating totes. This caused a huge problem as the fulfillment side of the facility continued to push WIP into the pipeline, and the production side of the operation starved because the buffer was unable to process all of the WIP. Understanding the subtleties of this buffer induction slowdown would greatly enhance management's ability to execute during the holiday period.

One of the capabilities that I built into the simulation is the ability for the user to change the minimum wait time for each of the buffer lanes. This code is located in the 'before reset' section of the Visual Logic code.

5.4 Uneven Distribution of Put-Away Inventory

Another one of the challenges at the UK facility involves the put-away process. If the put-away process is not managed closely, the inventory allocation across different inventory zones becomes uneven. As a result, the pick-waves across inventory zones gets skewed, with some inventory zones having large pick-waves and other inventory zones having very small pick-waves. Understanding the impact on operations of skewed inventory is important. In addition, calibrating collating in order to maintain balance between capital infrastructure, the put-away inventory, and the labor allocation can make or break the day's productivity.

Because my model dictates the allocation of inventory through collating rather than the other way around, the user can force the model to pull inventory from specific inventory zones. This capability means that the user can experiment with different collating procedures as well as different inventory distributions across the zones.

5.5 Impact of Bimodal Productivity Distributions of Associates and Temps

One of Amazon's key challenges during the peak holiday period is managing the productivity of associate workers and temporary workers. In the UK, Amazon generally maintains a steady associate workforce year round. Then, during the holiday period, temporary workers are used to triple the workforce to cope with the increase holiday order volume. As a result, the temporary workers tend to dilute overall process productivities. Understanding the impact of this temp dilution on system throughput and cycle time will help human resources and operations plan the temporary worker induction during subsequent peak periods.

SIMUL8 has the capability to add distributions together and sample from the sum of the two probability distributions. Creating this new distribution and sampling from it to get processing times will enable the user to discern the impact of the temporary worker population

5.6 Relationship between Throughput and Cycle Time

The two main goals of my internship project were to increase throughput and minimize cycle time. Many different variables affect total throughput including labor allocation, capital infrastructure, flow of WIP through the system and buffer size. In the case of cycle time, there are several important variables including buffer size, amount of

WIP in the system, work entry rate and labor allocation across the different processes. Optimizing these different variables will lend insight into how to increase throughput and decrease cycle time while minimizing cost expenditures.

This topic is a bit more problematic. As the simulation is unstable over longer periods of simulation time, it is difficult to completely drain the system of the startup inventory. As a result, getting good numbers for cycle time is problematic. I solved this problem in most of my analysis by using Little's Law.[6] SIMUL8 is continuing to decrease the bugs in their software by introducing new patches of at least a couple every month. In addition, they have a development team who is extremely interested in addressing specific problems and working with more advanced model users. By sending the crash file to SIMUL8, you may get a faster response.[7]

5.7 Impact of Variation in Wave Synchronization

This question is really relevant to buffer control, but it also affects cycle time and throughput. At buffer control, associates attempt to release waves to production in a chronological order. However, there are times when a particular wave is held up either in wave picking or in pre-sort. As a result, the associates at buffer control will release complete waves out of order to continue to feed the downstream production process. Understanding the impact on the system of greater variation in wave synchronization is important.

Looking at the inventory levels across the simulation over time seems to show that this issue is dramatically important. Within my model, the inventory waiting to be released at buffer control (because the waves are incomplete) is truly significant. This makes sense to me, as every 'system record day' that we had ended up with massive amounts of inventory waiting to be processed by the buffer control team. The number of batches open at buffer control / pre-sort within my model is often in the vicinity of several hundred.

Investigating the process of buffer control relative to productivity improvement or additional storage capacity could enable significantly enhanced capacity for a given facility.

5.8 Secondary Impact of Minimizing Cycle Time

As the operations team takes steps to minimize cycle time by adjusting system parameters such as buffer size and wave synchronization, understanding when downstream processes will starve becomes extremely important.

This can be readily investigated by looking at the inventory levels for different scenarios and identifying where downstream resources are under-utilized.

[6] Cycle Time = Inventory in the System / Arrival Rate

[7] support@simul8.com and team7@simul8.com will both be interested in any crash files.

5.9 Identifying the Optimal Number of Batches per Wave

One of the opportunities that Amazon has for process improvement involves changing the number of batches per wave. As a result, quantifying the savings from this effort could lead to significant advances in throughput and cycle time. Furthermore, identifying the optimal number of batches per wave could lead significant insight into the design process for the next generation facility.

The user can alter the number of batches per wave by changing the label control sheet in the Excel workbook.

Chapter 6: Conclusions

6.1 Simulations Development

Throughout the last 10 months, I have learned way more than I thought possible about simulation development. These lessons include the following:

- Before starting any simulation, ensure that you have an in-depth understanding of the processes that you are trying to model. I actually accomplished this by working in each of the departments and creating flow diagrams of the individual processes.

- Find a mentor who understands simulation and consult with him/her at regular intervals. In my case, I had my advisor, Professor Jeremie Gallien, at my disposal, but I probably didn't consult with him often enough. One good example of this is that I ended up scrapping much of the software code that I developed during the first three months and replaced it by simplifying assumptions in the final model.

- Question your assumptions early and often. This really is a double edged sword. First, ask yourself if you can make simplifying assumptions to avoid custom coding, and then ask yourself if some of the assumptions that you are making truly detract from what you are trying to accomplish. I ran into both problems throughout the course of my simulation. If you are going to err on one side or the other, err on the side of simplification, because you can always custom code in the future.

- Be very cognizant of the impact of warm up time. Within this simulation, it took me a long time to realize how much warm up time was actually needed for the simulation to reach equilibrium. This is especially important if you are seeding any queue with substantial startup inventory.

- I recommend that one build their simulation using fictitious data. As you get to the final stages in your model and you are prepared to run trials, then invest the time to get the data that is needed for validation and analysis.

- As you set out to build a very complex simulation model, first lay out a plan of attack on paper. This plan should have detailed information about each of the components of the model that you are planning to build. As you build and code, start with one section at a time and stress that component to ensure that it operates properly. Then, once all of the components of the model are complete, replicate them as needed and connect them to create the model system. After the model system is completely built and interconnected, then stress the entire simulation to ensure that it is operating according to plan.

6.3 Simulation Program Selection

Discrete Event Simulation program selection is a tough subject. In my case, I had only been exposed to one program through MIT, and I heavily relied on the advice of one of my advisors to determine that SIMUL8 was the program of choice. In hindsight, I think that, if you are going to spend any money on consulting, this might be the time to spend it. Without in-depth understanding and past experience in multiple discrete event simulation programs, I think that it is very difficult to make this decision.

While SIMUL8 turned out to be a great solution to this problem, this model is clearly operating at the cusp of SIMUL8's capabilities. Perhaps there is another program out there that would have been better than SIMUL8. Perhaps a more experienced SIMUL8 programmer with a better grasp of Visual Logic would have been able to create a more robust system. If I had to make this decision in the future, I would spend the money to consult a professional before investing the massive time commitment to learn a new program and programming language.

6.3 Recommendations

Having spent the last 10 months on this project, I have the following recommendations for Amazon:

6.3.1 Amazon.com

- Continue to invest in discrete event simulation and build an in-house capability to model real world issues. This investment will enhance your competitive advantage in operations and yield insights that, otherwise, would not have been possible.
- Search for a seasoned professional, with experience in multiple simulation programs to lead this effort, and ensure that he has the resources needed to succeed.

6.3.2 Amazon.co.uk

- Focus on improving mean productivities within each of the process paths.
- If you plan to increase throughput at the UK FC, continue to expand pre-sort, re-bin and packing capacities within your facility.
- If you gain insights from this thesis, get someone on the European team that can understand and continue the work that has already been accomplished.
- Brainstorm the buffer control process. How can you enhance the job that these associates need to do? Do they have sufficient and optimized storage capacity for the amount of inventory that stalls in this process? Scrutinize the wave synchronization issue. Is there a way to minimize the wave synchronization? Is it possible to increase or relieve batch accumulator capacity by changing the way that you run buffer control?
- Link the inventory that enters the system through collating to the bottleneck processes.
- If the queue at buffer control becomes unmanageable, focus on decreasing the productivity standard deviation of the picking and presort processes.

Chapter 7: Appendices

7.1 SIMUL8 Model Documentation

As it is not possible to include electronic files with a master's thesis, if you would like a copy of the simulation file, the Microsoft Excel spreadsheet or the model documentation, please email markmast@amazon.com.

7.2 List of Figures

7.3 List of Tables

7.4 Recommended References and Resources

My advisor, Jeremie Gallien, cites the following textbooks as good sources for simulation research:

- Law, A. and W. Kelton, Simulation Modeling and Analysis, 3rd ed., McGraw-Hill (2000).

- Ross, S., Simulation, 3rd ed., Academic Press (2002).

In addition, he cites the following source as a guide for available simulation software:

- Swain, J., "Power Tools for Visualization and Decision-Making," OR/MS Today, February 2001. Available online at

In addition, Sheila Bragg (LFM '03) cited the following references on warehouse operations in her thesis.

- Reveliotis, Spyros. IE 6202: Warehousing Systems Lecture Notes, Georgia Institute of Technology, Atlanta, Fall 2002.

- Bartholdi, John J. and Hackman, Steve T., Warehouse & Distribution Science Release 0.1.2 May 22, 1998; revised May 24, 2002.

- Maloney, David, October 2002, "Sorting to Success", *Modern Material Handling*, 23-27.

- Bozer, Yavuz A., Quiroz, M.A. and Sharp, G.P., 1988, "An Evaluation of Alternative Control Strategies and Design Issues for Automated Order Accumulation and Sortation Systems" *Material Flow*, 4, 265-282.

- Bozer, Y.A., M.A. Quiroz, and G.P. Sharp, 1985, "An Empirical Evaluation of General Purpose Automated Order Accumulation and Sortation Systems Used in Batch Picking," *Material Flow*, 2, 111-131.

- Choe, K.I., G. Sharp, and R.F. Serfozo, 1992, "Aisle-Based Order Pick Systems with Batching, Zoning, and Sorting," *Proceedings of the International Material Handling Research Colloquium*, 389-420.