Master's Thesis
ISRN LUTMDN/TMFL–10/5086–SE

Modeling a converting line using discrete event simulation

– to evaluate the effects of different production strategies

Jakob Harder, David Lindell

Packaging Logistics
Lund University

Modeling a converting line using discrete event simulation to evaluate the effects of different production strategies

Division of Packaging Logistics

Faculty of Engineering

Lund University

Sweden

ISRN: LUTMDN/TMFL-10/5086-SE

Printed by Media Tryck

Lund 2011

Preface

This Master's Thesis is a part of the examination of a Master's degree in Industrial Engineering, finished at the Department of Design Sciences at Faculty of Engineering LTH at Lund University.

We would like to thank Marc Kickulies, our supervisor, for his help during this project. Without him this project could not have been realized.

We would also like to thank our course supervisor Ola Johansson, PhD at the Division of Packaging Logistics at Lund University, for his advice throughout the project.

Further we also want to thank Jason Lightfoot at Flexsim UK for all the invaluable help solving our programming difficulties. His assistance greatly increased the quality of our work.

Finally we would like to thank everyone at the factory for putting up with us and answering all our tedious questions.

22nd of November 2010, Lund
David Lindell
Jakob Harder

Abstract

Title	Modeling a converting line using discrete event simulation to evaluate the effects of different production strategies
Division	Division of Packaging Logistics, Department of Design Sciences, Faculty of Engineering LTH at Lund University
Authors	David Lindell and Jakob Harder
Tutor	Ola Johansson, PhD , Division of Packaging Logistics
Supervisor	Marc Kickulies, company representative
Key words	Discrete Event Simulation, Production, Logistics, Flexsim, Converting Line
Purpose	The purpose of this Master's Thesis is to create a model of a converting line factory. The model will be used to investigate factory performance when altering the order structure as well as other what-if scenarios. The model should also be easily adaptable so that it can be used to model other factories as well without the need to rebuild it from scratch.
Method	The project follows Bank's (2000) process for conducting a simulation study. The system is modeled using the discrete event simulation tool Flexsim.
Conclusion	The main results of this study show that the throughput of the system is very stable when it comes to changing the order structure. The orders can be shortened without too much losses of throughput, the real performance dip doesn't occur until the orders have been shortened to less than 30%. The baseline scenario for the model has a throughput of 17.14 million meters per month. Doubling the order lengths gives a slight but still significant increase of the throughput by 1.4 million meters. Shortening the order length to 30% decreases the throughput by 0.8 million meters which might be an acceptable solution if the factory needs to run shorter orders. Further decreasing the order lengths causes a steeper decline of performance. At 20% order length the throughput is 14.55 million meters per month, and at 10% it decreases to 11.27. Another conclusion from the output of the model is that the current transportation system is adequate for the factory. Removing the travel times doesn't significantly increase the throughput of the factory. Finally, experiments with FIFO-strategies (First In First Out) showed that the factory's performance suffers when removing the sequencing of orders that currently is in use. Removing buffers between the machines as well creates a drastic drop in throughput.

Sammanfattning

Ett förpackningsföretag som producerar material till livsmedelsindustrin vill undersöka olika produktionsscenarier för att skapa sig en bättre förståelse för de olika faktorer som påverkar prestandan i deras fabriker. Företaget har tidigare använt sig av simulering inom andra produktionsprocesser med goda resultat och är intresserade av att använda sig av denna teknik mer i hela företaget.

Syftet med projektet är att skapa en simuleringsmodell av fabriken för att sedan använda denna till att undersöka vad som händer när olika förändringar görs. Förändringarna som testats i modellen är följande: att variera längden på ordrarna, att ta bort tider för intern materialtransport, att låta produktionsordrar köras i den ordning de läggs och att ta bort alla mellanlager från fabriken.

Projektet som denna rapport beskriver innefattar en studie av produktionsprocessen, en simuleringsmodell, en serie experiment, samt en slutrapport. Projektet följer Banks (2000) process för att genomföra en simuleringsstudie. Modellen är skapad i simuleringsprogrammet Flexsim.

Systemet som modelleras består av ett antal processteg. I denna studie har modellen begränsats till att innefatta de tre stegen tryck, laminering och skär. Modellen är baserad på en existerande fabrik och innefattar den produktionsplanering som används där. Ett system av automatiska truckar som används för materialtransporter inom fabriken är också en del av modellen. Modellen är baserad på data från ett feedback-system på fabriken, samt data från en tidigare simuleringsstudie och vissa egna observationer.

Resultaten från studien visar att produktionen inte är särskilt känslig för variationer i orderlängd. Storleken på produktionsordrarna kan minskas ända ner till 30% utan att drastiskt minska fabrikens produktionskapacitet. Minskning av orderlängd till 30% ger en minskning i produktion från 17,14 miljoner meter i månaden till 16,36. Efter 30% ligger det dock en brytpunkt, och om den passeras märks en kraftig försämring av produktionskapacitet. Vid 20 % ligger kapaciteten på 14,55 och vid 10 % är den 11,27.

Ett annat resultat från studien visar på att transportsystemet som används är fullt tillräckligt, då ett experiment där alla transporttider tagits bort gav samma produktionskapacitet som nuläget.

Experimentet där produktionsordrarna körs i den ordning de läggs, rakt igenom fabriken, visar på en klar försämring i kapacitet. Kapaciteten sjunker till ett genomsnitt med 16,61 miljoner meter per månad. Att dessutom ta bort alla mellanlager resulterar i att kapaciteten sjunker ner till 8,73.

Contents

1 Introduction

1.1 Background

A packaging company produces packaging material used in food industries. In their factories, called converting lines, big rolls of paper are printed, laminated and then slit into smaller rolls.

The packaging company wants to examine different production scenarios using discrete event simulation in order to gain better understanding of the different parameters that affect the performance of its factories. The company has previously used discrete simulation tools on other production processes with good results and is interested in increasingly using this technology company-wide.

In 2009 a discrete event simulation study was done on the same production process studied in this project as part of another master thesis. The model included the first two steps of the converting line and experiments were done to examine the requirements on the process for running printing and laminating as a one-to-one process. The model did not have any advanced visual presentation and did not include the physical layout of the factory as a factor. The 2009 study was done using the discrete event simulation tool SIMUL8, which is not the standard program used throughout the company. (Ferrada & Omeragic, 2010)

The packaging company has previously used the tool Flexsim to model other manufacturing processes. The Flexsim models have been used as a way of optimizing processes as well as a sales tool where the model can help the sales person answer the customers' questions regarding costs and efficiency by showing the process in the model.

This master thesis marks the beginning of an effort to make simulation a common tool at the company. Since the packaging company's earlier production simulation models have been done using Flexsim, and a business relationship has already been established with them, Flexsim is the tool used in this project.

1.2 Purpose

For the packaging company the purpose of this project is for it to be the first step in a long term simulation project. The model will be used to generate knowledge of which factors affect factory performance. Another purpose is to generate interest and promote the use of simulation as a tool for analysis at the company.

The more immediate purpose for the authors is to create a model of a converting line factory. The model will be used to investigate factory performance when altering the order structure as well as other what-if scenarios. The model should also be easily adaptable so that it can be used to model other factories as well without the need to rebuild it from scratch.

1.3 Scope of project

The scope of this project consists of four different parts.

- **Process study** - A process study is conducted during which knowledge on the system basics is gathered and used to create a conceptual model of the factory. In this step data such as order patterns, order sizes, setup times and processing speeds are gathered.
- **Model translation** - The gathered data and knowledge is put into a virtual model that fulfills a set list of specifications.
- **Experiments & analysis** - The completed model is used to conduct a series of experiments. Through analyzing the results from the experiments, conclusions regarding the system can be drawn.
- **Reporting and documentation** - A report containing the findings of the study as well as methodology and underlying theory is written. The model is also thoroughly documented to make further development of it by other personnel easier.

1.4 Scope of model

The scope of the model is to represent the process from the raw material storage through the three machine groups; printer, laminator and slitter, with buffers in between, and out to the finished goods storage. The model includes setup times in the machines based on the specific characteristics of different orders as well as different process speeds. The model also includes a sequencing scheme to mimic the production planning as well as the transports within the factory.

The model does not include the process steps pre-print, paint preparation and doctoring. The reason to exclude these parts is both a matter of time and the fact that they can be treated as individual systems that only support the main process. The operators are not modeled specifically but their effects are covered by the setup times. The warehouse containing the intermediary storage’s is included in the model but the inner workings and material handling of the warehouse is not. Waste is not included in the model nor is the inbound and outbound logistics of the material.

2 Methodology

The project will follow the well recognized process for conducting a simulation study presented by Banks (2000), which is shown in Figure 2-1. This process is also the basis for the disposition of this report.

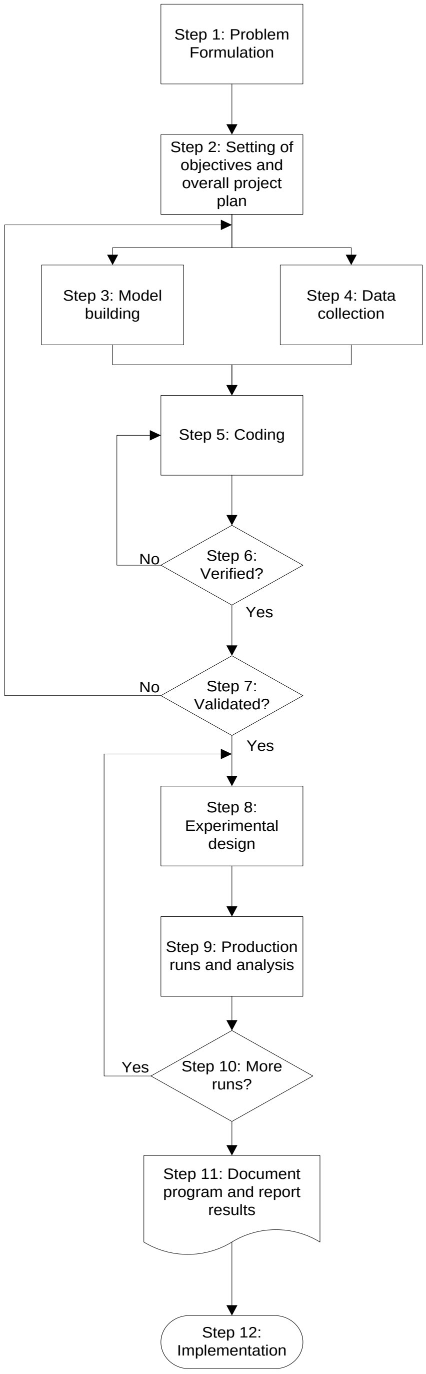


Figure 2-1 Banks (2000) process for conducting a simulation study

2.1 Problem Formulation

The initial step in a simulation study is to formulate the problem that the study is supposed to solve. If the problem is formulated by the client the analyst must ensure that the problem is clearly defined and understood by both parties. It is not uncommon that a problem exists but the nature of it is not known which means a reformulation may be needed when the process to be simulated has been studied in more detail.

2.2 Setting objectives and overall project plan

At this stage the problem has been formulated and a decision of whether or not simulation is the appropriate tool for the task needs to be made. More detailed objectives of what the study is supposed to produce are set and a project plan is created which details the resources and time needed to fulfill the objectives.

2.3 Model conceptualization

Since a real world system is too complicated to be completely replicated by a model, simplifications and abstractions need to be made. The nature of the problem decides how far the system can be simplified while retaining enough information and logic to be a representative model to solve the given problem. During this phase the system is studied to determine which parts are essential to describing and solving the problem. At the end of the phase a conceptual model has been created which will serve as a blueprint for what the simulation model should include.

In this project, model conceptualization was done by mapping the processes in the factory after spending a week observing them. System description and conceptualization is described in chapter 4 and 5 respectively.

2.4 Data collection

A simulation model requires a lot of data and the collection should start as early as possible. This is done in parallel to the conceptualization since the scope and detail of the model determined which data is needed. The data collection can continue well into the model translation phase as the model becomes more complicated and new data needs are discovered.

Much of the data used in this project comes from a previous study by Ferrada and Omeragic (2010). Additional data was obtained from the factory’s feedback system as well as direct observation and interviews. Data collection and analysis is described in chapter 6. Theories used to do this are explained in chapter 3.2.

2.5 Model translation

The model translation phase is also known as the coding phase since this is when the conceptual model is translated into a computer simulation model. There are many different programming languages and dedicated simulation software that are suitable for this and the choice depends on what level of flexibility and detail is needed to program the model.

The model was built using the software Flexsim, which is a dedicated simulation tool. The overall architecture of the model is described in chapter 7.1.

2.6 Verification

Verification means making sure the simulation model is an accurate representation of the conceptual model and that it behaves as intended. This requires a lot of debugging and is an iterative process conducted when programming the model.

Verification of the model is described in chapter 7.2. Theories regarding verification are explained in chapter 3.3.1.

2.7 Validation

Validation means making sure the model is an accurate representation of the real system and is often conducted in parallel with verification. The model is calibrated with historical data and the outputs of the model and the real system is compared. This is repeated until a satisfactory level has been achieved.

Validation of the model is described in chapter 7.3. Theories on validation are explained in chapter 3.3.2.

2.8 Experimental design

During this phase experiments that are to be run on the model are designed. The analyst must decide which experiments to conduct and decide parameters such as how many runs are needed and the length of those runs.

Experimental designs in this project are described in chapter 8.

2.9 Production runs and analysis

The predefined experiments are run and the results are analyzed during this phase. The performance measures of different experiments are compared to determine which solution is the best.

Output data analysis is discussed in chapter 9. Theories on this are explained in chapter 3.4. Results and conclusions are presented in chapter 10.

2.10 More runs

If the confidence intervals are too wide or a new configuration has suddenly become interesting more runs are needed.

2.11 Documentation and reporting

The documentation of a simulation project consists of two parts, documenting the model and documenting the project. The model needs to be documented to enable other users to examine and modify the finished model at a later date. This builds confidence for the validity of the model which is needed if it is supposed to be a basis for business decisions. The project process and findings should also be documented and finally a recommendation is made.

2.12 Implementation

During the implementation phase the results of the study are tested and implemented to the real system. Since this phase is beyond the scope of this project it will not be discussed further.

3 Frame of reference

3.1 Discrete event simulation

3.1.1 What is simulation?

A simulation model is, in this context, a computerized model created to mimic a real world process. Simulation is often used as a tool for trying out potential changes to a process before implementing them. It can also be a tool to be used when designing a new process or a way of estimating the capacity of a process that doesn't exist yet. There are several other types of computerized simulations that are not processes, for example simulation of abrasion resistance. This is however not what this project is about.

A simulation model is almost never a complete replication of a real world process. It often entails some kind of assumptions or simplifications without which modeling the process would become a lot more complex and time consuming. It is up to the developer to decide the trade-off of which assumptions and simplifications to make based on what output is needed from the model and how much this affects the accuracy of the model.

3.1.1.1 Technical explanation of discrete event simulation

In order to give a better understanding of discrete event simulation, this chapter describes how it actually works.

Discrete event simulation is a type of simulation where the model only deals with chronologically ordered events which are separated by lengths of time. During these times the model and the objects in the model take on different states. Processes, both physical and intangible, are modeled with a flow of discrete items. This can be customers, material, orders, data or money that passes through resources where activities are done on them. If another item is currently using a resource the resource is considered busy and unavailable, and the arriving item has to be placed in queue and wait for its turn. The basic events in a discrete event simulation model are:

- the arrival of an item to the system
- the departure of an item from the system
- the start of an activity
- the end of an activity (this may coincide with the start of the next activity, performed on the next item, and it can also coincide with the departure of the item from the system)

Some basic states in a discrete event simulation model are:

- number of items in the system
- number of items in a queue
- states of a resource (busy, idle)
- states of an item (waiting, processing)

The events are placed in the “event list” in ascending order by time of occurrence. When an event is read from the list, the procedure for that event is processed and an attribute representing the time, called the clock, is updated to the time of the next event (the clock can be said to represent “now” in time). As more events are added over time and events occur, the event list gets updated. In order for the simulation to work there must also be a defined inter-arrival time of items to the process. The inter-arrival time can either be a fixed number, a number taken from a list of numbers according to some function or a number drawn from a stochastic distribution. Each time an arrival occurs, a new arrival is added to the event list at the time defined as: clock time + inter-arrival time. Analogously the end of an activity is added to the event list when the start of the activity occurs. This time will then be equal to: clock time + activity time. If another item is placed in queue waiting for the activity, the end time for the activity on the first item will also be the starting time for the activity on the second item.

There are lots of ways that modeling can become more complex, for example adding the need for a certain combination of resources in order to perform an activity or adding breakdowns that disrupt activities and claims more resources. The logic described above is however essentially what discrete event simulation is.

3.1.2 Why use simulation?

Simulation is a tool that can solve large complex problems where other methods such as pure math fall short. Smaller problems like a single queue-server relationship can be solved mathematically using differential calculus, probability theory or other methods. The result of a mathematical model is often just a single or a few performance measures and does not give an overview of how the system produced these results. A mathematical model is also very rigid and any change requires the model to be solved again. Most real world systems are too complex to be simplified enough to be solved with mathematical models without losing important parts of the real system. Simulation enables the construction of models that can mimic the behavior of very complex systems. Simulation models also have the advantage of being adaptable once constructed. This allows the user to make changes in the probability distributions, input parameters or other model logic. This adaptability is the strength of simulation as it allows experimenting with different changes to the model. (Banks et al., 2001, pp.3-4)

3.1.2.1 When is simulation the appropriate tool?

Simulation is appropriate for systems that are large, complex and difficult to quickly get an overview of. It is appropriate for studying the system as it runs to gain knowledge of where the bottlenecks are and what causes them. By varying the input parameters and see the effects on the system a great deal can be learned and potential improvements and solutions can be identified. Simulation is furthermore a tool for experimenting with different solutions that would be too costly, time consuming or dangerous to test on the real world system. It can also be used to experiment with non-existing systems as a part of the development process. Verifying assumptions and presenting solutions, especially with a model that graphically shows what happens, is another use for a simulation model and gets more attention and face validity than a report based on calculations. (Banks et al., 2001, pp.4-5)

Simulation works best with systems that operate according to specific rules or can be simplified into logic flowcharts. It is easier to simulate a complex but well defined production process than for example a product development process. Human behavior is the most difficult to simulate as it is subject to individual preferences and is difficult to simplify.

3.1.2.2 When should simulation not be used?

There are many situations where simulation is not the appropriate tool. First of all it should not be used when the problem can be solved using common sense or analytical methods. This is the case for a simple process with 100 units arriving per hour and the units being served at a rate of 8 per hour. A simple calculation 100/8 = 12.5 reveals that 13 or more servers are needed to have a stable process. Simulation should not be used for systems where it is easier to conduct experiments on the real world system than developing a model. A simulation study is often costly and time consuming and should not be undertaken if the time frame is too short, if there is lack of funding or if the anticipated savings are less than the cost of conducting the study. Acquiring relevant and correct data is often a big part of a simulation study and if data is lacking and cannot be estimated a simulation study cannot be performed. A study should not be performed if the model cannot be properly verified and validated since the results produced by the subsequent experiments are not reliable. Finally it is important to manage the expectations for what simulation can accomplish, a model can never be a complete replication of reality and cannot solve every problem. (Banks et al., 2001, pp.5)

3.1.3 Components of a system

A simulation model is a simplification of reality and is comprised of the following components. (Banks et al., 2001)

3.1.3.1 Entity

An entity is an object of interest in the system. This can be customers, machines or packages depending on the system.

3.1.3.2 Attribute

Attributes are properties of entities which are used to describe information which is important to how the model runs.

3.1.3.3 Activity

An activity is a time period of specified length. It can be service times at a cashier or the time spent performing machine operations.

3.1.3.4 State

The state of the system is described by a collection of variables that describes the system at any point in time. Dedicated simulation software keeps track of a lot of information describing the system state automatically but it is sometimes necessary to add additional variables describing the properties of the system studied.

3.1.3.5 Endogenous event

An endogenous event is an event occurring within a system that may change the state of the system. This can be the completion of expediting an order.

3.1.3.6 Exogenous event

En exogenous event is an event occurring in the environment outside the system which may change the state of the system. This can be arrival of orders to the system.

3.1.4 Model types

3.1.4.1 Static/Dynamic

A static model describes the system at a certain point in time, often referred to as Monte Carlo simulation. A dynamic model describes a systems behavior over a predetermined time period.

3.1.4.2 Deterministic/Stochastic

A deterministic model contains no random variables. The output of a deterministic model is predictable and will produce the same output every time given the same input. A stochastic model contains random variables for arrival and service times, producing different outputs each run. The output of a stochastic model must be treated as a statistical estimate of the systems performance. By conducting many runs confidence intervals can be created and estimations of important parameters can be determined.

3.1.4.3 Discrete/Continuous

A discrete system is a system where the state variables only changes at a discrete set of points in time. For example the number of customers in a store which have to be a discrete number at all times and the number changes with the events of arrival or departure of customers. A continuous system is a system where the state variables change continuously over time. This could be a chemical treatment plant where compounds are mixed at a certain rate and the in and outflow is based on volume/time unit. A discrete system can be modeled using continuous methods as well as the other way around. The choice depends on what problem the model is designed to solve and what components it focuses on. (Banks et al., 2001)

3.2 Input data analysis

For a simulation model to be created a lot of data is required. Often that data is not available in a neatly categorized way and it needs to be analyzed and formatted before it can be used in the model. There are four steps that need to be taken in order to get acceptable input data. Chung (2004, pp.5.1-5.4) covers several techniques of input data analysis, some of which are described below.

3.2.1 Collect data

The first step is to collect the data. There are many sources from which data can be collected. If the system to be simulated exists it is possible to get actual performance data. The sources of this data can be historical data, manufacturer specifications and vendor claims, operator and manager estimates or data capture and direct observation.

3.2.1.1 Historical data

Most production processes collect feedback data automatically and these records can be very useful when building a model and it can be a time saver by not having to collect the data manually. This approach does however have some problems. The system may have changed since the data was collected and if the model is built using outdated data validation will be very difficult. Another problem that can arise is if the historical data lacks crucial data needed to build the model, something that can cause many problems if it is not detected in an early stage.

3.2.1.2 Manufacturer specifications and vendor claims

This is another form of data that is conveniently collected by someone else but requires some scrutinizing before being used in the model.

3.2.1.3 Operator and manager estimates

This data can be very valuable if it is not possible to collect actual data. Operators can be asked to estimate the minimum, mean and maximum time it takes to complete a task and that data can be used to create triangular distributions. Input from managers can also be useful to get a broader perspective.

3.2.1.4 Data capture and direct observation

If time and resources are available it is possible to collect the data manually. This can be done by an automatic data capturing device like a camera which can record the data of interest. If such a setup is not possible direct observation is the last resort. This can be very time consuming and exhausting but will i the end provide some data.

3.2.2 Identify a probability distribution

Once the data has been collected it needs to be analyzed by identifying which probability distribution describes the collected data. This is done by sorting the data and by creating a histogram of the values. By looking at the histogram plot it is often possible to identify which family of distributions that will fit the plot. If the number of data points is low, fewer than 30, the histogram approach becomes difficult and it is recommended to use a Quantile-Quantile plot instead. (Banks et al., 2001, pp.327-336)

3.2.3 Choose parameters

Once a family of distributions has been selected the parameters of the distribution must be identified. This can be done by a variety of mathematical functions depending on the given distribution, by trial and error or by using a software package capable of estimating the parameters. (Banks et al., 2001, pp.336)

3.2.4 Evaluate the chosen distribution - goodness of fit

After a distribution and its parameters have been chosen it needs to be tested to see how well it matches the original data. There is a number of goodness of fit tests that are used to judge whether or not the distribution can be accepted. (Chung, 2004, pp.5.19)

3.2.4.1 Chi-square

The Chi-square is the most commonly used goodness of fit test. The test is based on comparing the actual number of observations versus the expected number of observations that fall into a predetermined number of equally sized boxes. For the test to produce a reliable result, at least 30 data points are required. The test can be performed on different significance levels. (Chung, 2004, pp.5.21)

3.2.4.2 Kolmogorov Smirnov

The Kolmogorov Smirnov test is another goodness of fit test and is used as a complement to Chi-square. The test is not as accurate as the Chi-square but it requires less data points to work and can be used when there is not enough data to perform a Chi-square test. The KS test is also weaker when used on discrete distributions. The KS test compares the cumulative theoretical distribution with the cumulative observed distribution and if the maximum difference between the two exceeds a critical KS value the observed distribution cannot be from the theoretical distribution. (Chung, 2004, pp.5.24)

3.2.5 Computer support

Most simulation software includes third party statistical software that is able to match a probability distribution to a set of data points and perform goodness of fit tests. The simulation software Flexsim comes with the software Expert Fit which is able to fit data to over 40 different probability distributions and rank them depending on the goodness of fit result.

3.3 Verification and validation

Verification and validation is needed to ensure that the model produces reliable results. This is needed for the reliability of the results produced by the experiments performed later in the process.

3.3.1 Verification

Verification is the process where the developer makes sure the model behaves as it was intended and that it is an accurate representation of the conceptual model. This means checking that the data is input correctly and the logic is implemented as intended. To verify the model there is a number of methods.

1. *The model can be checked by someone else than the developer.* This can be done using formal audits before certain milestones are reached or be of a more informal character continuously throughout the process.
2. *Create flowcharts representing the logic of actions taken by the system when an event occurs.* The flowcharts can be seen as a kind of high level pseudo code and works as a blueprint when coding. It is good to continuously compare the code to the flowcharts to avoid mistakes.
3. *Examine the output of the model when varying the input parameters and see if it is reasonable.* Also check the statistics of different entities in the system to make sure their values are reasonable as well. Too high or too low utilization of a machine could indicate incorrect process time parameters. The average and total length of queues and items in the system should also be examined to detect logical errors and incorrect parameters. These kind of statistics are easy to come by in most dedicated simulation software since it is collected automatically. When using more general languages such as Java or C this requires more programming by the developer.
4. *Make the model as self-documenting as possible with comments in the code and informative variable names.* The model should also be thoroughly documented so that others can examine the model logic and verify it themselves.
5. *Examine the animations of the model for unintentional events and actions taken by the model.* This feature provided by newer dedicated simulation software is invaluable for detecting logical

errors and coding mistakes. Watching the model run might reveal items disappearing or being routed wrong in the model. It can also show queues becoming too large or empty or machines producing too fast or slow.

6. *Use the debugging tool that is present in most simulation software.* The debugger can be used to examine specific parts of the program, pausing the simulation when certain conditions are met. Information related to the event is then displayed on the console or is available in different menus. This enables the developer to check all parameters and responses during critical events to see if the model behaves as intended.
7. *Use the principles of structured programming.* Use a top down design and have a clear picture of the program before the coding phase begins. Follow the principle of modularity and break down the model into smaller subsystems which can be verified individually. (Banks, 2000, pp.27)
8. *Pair programming.* This is a programming technique where two programmers work at the same station writing the same program. One person writes the code while the other one checks it for errors. This technique has proven benefits of producing better quality code with fewer defects in a shorter period of time.

(Banks et al., 2001; Banks, 2000; Cockburn et al., 2000)

3.3.2 Validation

Validation is the process of answering the question "is the model a good representation of the real system?" It is often done simultaneously as the verification in an iterative approach calibrating the model with the system. It is impossible to reach a model that is 100% representative of the real system and thus a full validation can never be accomplished. The cost per iteration must be weighed against the expected gain from the improvement of the model.

1. *Face validity.* A model has face validity if it on the surface looks reasonable to persons knowledgeable of the real system. Potential users and people knowledgeable of the system should be involved early on to create a conceptual model that has the desired degree of realism built into it. Involving users also builds credibility with the users and managers. If possible the model should be validated by a number of experts who know different parts of the real system.
2. *Validate model assumptions.* In the process of building a model assumptions and simplifications are made to simplify a complex reality into a conceptual model. Assumptions are made both about the structure and on the data and it is important to show that these assumptions are sound and don't have a negative impact on the validity of the model. Structural assumptions are assumptions made about how the system works and are often simplifications and abstractions of reality. Structural assumptions should be observed and/or discussed with persons with knowledge about the system. Validating data assumptions means checking the reliability of collected data and making sure that the transformations of the data into distributions and

parameters have been done correctly. The distributions can be validated with various goodness-of-fit tests.

3. *Compare model input output transformations to corresponding transformations of the real system.* If the model is based on an existing system and performance data from that system is available to be collected the two outputs can be compared. It is advisable to have an additional historical data set other than the one used to develop and fine tune the model to be used as a final test of the model. An accurate replication of history is in this sense seen as a validation that the model will be able to predict the future. Most models are developed with the purpose of solving a specific problem and have a few performance measures that are important when solving that problem. This can be throughput or lead time and when validating, those measures are the ones that should be compared with the real system. If the model is used for another purpose than its initial and other performance measures are of interest there may be a need of validating the model again for those measures.
4. *Sensitivity analysis.* This test is done to see if the model responds in the expected way when varying the input parameters. If the arrival rate is increased or the process times are prolonged the queues should start building up and the other way around.
5. *Extreme condition tests.* This is done to check the model response for extreme input data. If the arrival rate is set to a very high number is the response reasonable?
6. *Consistency checks.* This is done when the model is used over longer periods of time and the system might have changed since it was last run. New machines might have been installed or other improvements could have been made. These changes should be incorporated to the model to keep it validated.
7. *Touring tests.* The touring test performed by having the model produce a report of the run and then letting someone knowledgeable about the system check if it looks like something that the real system would produce. It can even be done by placing the model report among some real ones and see if it is possible to determine which is which. If it is not possible to distinguish the one produced by the model the results are considered reliable.

(Banks, 2000; Banks, 2001)

3.4 Output data analysis

3.4.1 Welch Confidence Interval Approach

The Welch Confidence Interval Approach (Chung, 2004, pp.10.10-10.11) is a commonly used method to compare the output of simulation models. The method has the advantage that the data compared does not have to have similar variance for the test to work properly. The test assumes the worst case scenario of having different variances between the data compared.

The degrees of freedom, d.f., is calculated by:

$$d.f. = \frac{[\frac{s_1^2}{n_1} + \frac{s_2^2}{n_2}]^2}{\frac{\left[\frac{s_1^2}{n_1}\right]^2}{n_1 - 1} + \frac{\left[\frac{s_2^2}{n_2}\right]^2}{n_2 - 1}}$$

Where

$d.f.$ = degrees of freedom

s_1^2 = sample variance of the first alternative

s_2^2 = sample variance of the second alternative

n_1 = sample size of the first alternative

n_2 = sample size of the second alternative

To get the worst case scenario the d.f. should be rounded down to the closest integer. A lower d.f. leads to a bigger number for the t-distribution and wider confidence intervals. This creates the situation where the difference between the two datasets needs to be bigger to conclude that there is a statistically significant difference.

The confidence interval is calculated by:

$$\bar{x}_1 - \bar{x}_2 \pm t_{d.f,1-\alpha/2}\sqrt{\frac{s_1^2}{n_1} + \frac{s_2^2}{n_2}}$$

Where

$\bar{x}_1$ = the mean of the first alternative replication $\bar{x}_1$

$\bar{x}_1$ = the mean of the second alternative replication

t = the t value for the d.f. and 1-α/2

This produces a confidence interval of a max and a min value. If the interval covers 0 the difference between the datasets is not statistically significant. If the interval does not cover the 0 the difference between the alternatives is statistically significant.

3.4.2 ANOVA

ANOVA (Chung, 2004, pp.10.13-10.16) is an acronym for analysis of variance and is used to compare three or more datasets to see if any of them is statistically significantly different from the rest. The method is based on a ratio of the variance between different alternatives divided by the variance within the different alternatives.

The calculation steps to perform an ANOVA test are the following:

- Calculate the sum of squares total
- Calculate the sum of squares between
- Calculate the sum of squares within
- Calculate the mean squares between
- Calculate the mean squares within
- Calculate the f statistic
- Compare the f statistic to a critical f value

At the end the test will produce an f statistic and a critical f value. If the f statistic is greater than the critical f value the test indicated that at least one of the datasets is significantly different from the others. If the f statistic is lower than the critical value there is no statistically significant difference between any of the datasets.

The implementation of the ANOVA test included in EXCEL was used to conduct this test.

4 System description

4.1 Main Manufacturing process

A converting line factory has several supporting processes and one main manufacturing process. The main manufacturing process converts big rolls of paper into smaller rolls of packaging material, and it consists of three machine steps; printing, laminating and slitting. Figure 4-1 shows a basic visualization of that process.

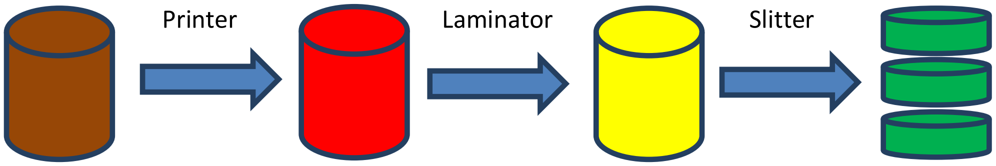


Figure 4-1 Steps in production

The factory in question currently has two printers, two laminators and three slitters. The machines have been designated as follows: old printer, new printer, old laminator, new laminator, left slitter, middle slitter and right slitter. Before and after each machine a buffer is available to increase availability of the paper rolls. And before each in-buffer and after each out-buffer, storage is available, located elsewhere. Transports between each step in the factory are done by AGVs (read more about this under AGVs). Figure 4-2 is a schematic picture of the flow of the main manufacturing process and its supporting processes.

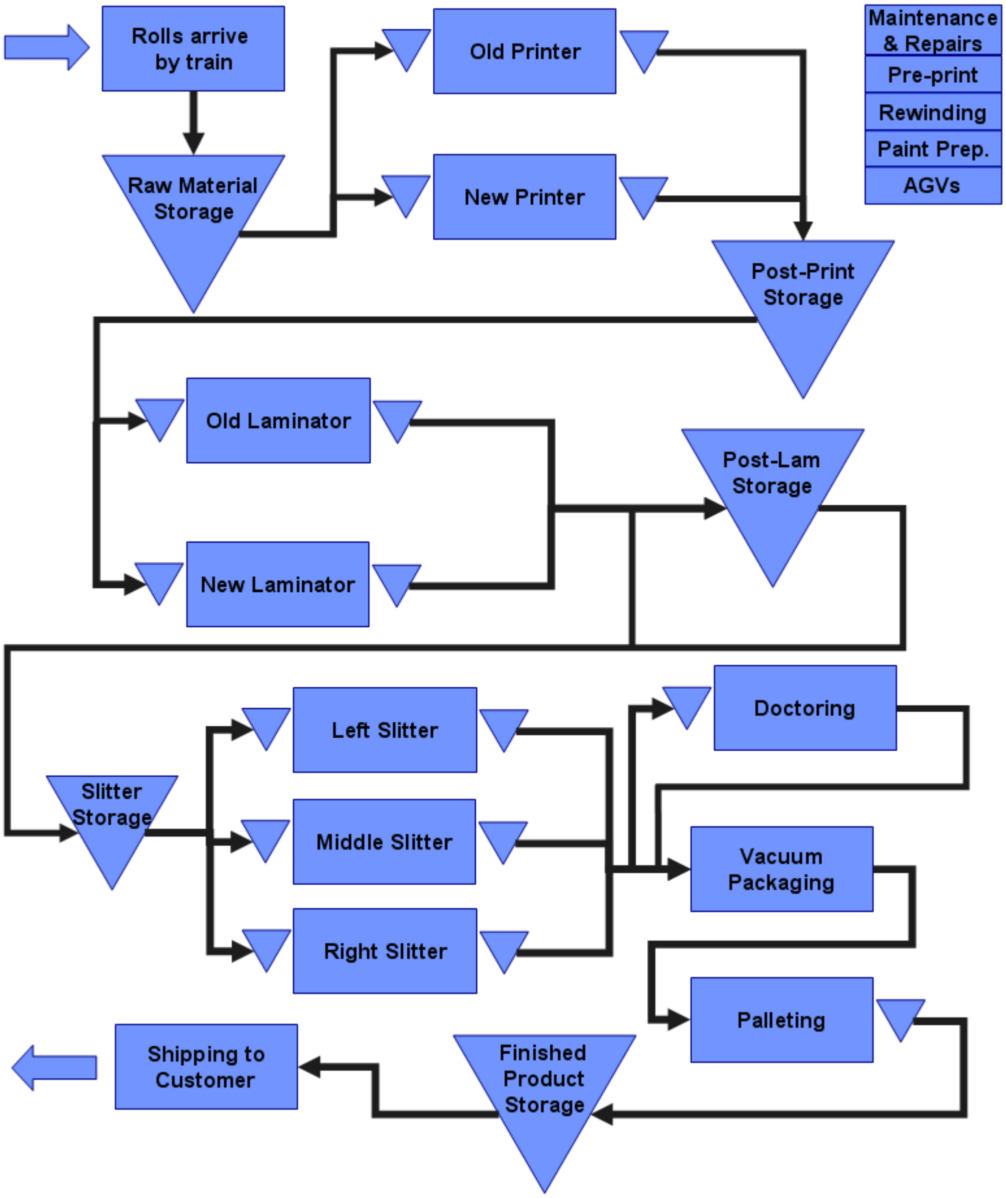


Figure 4-2 System overview

Just as the Figure 4-2 shows, the factory's main flow of materials begins with the raw materials storage. Rolls of paper are delivered by railroad and placed by lift truck according to its properties in different locations in the storage warehouse. Rolls remain in the raw materials storage until a production order is placed that shows that it's needed at the printer (read more about this under Planning and sequencing). Once a roll is about to be sent to the printer, a manned lift truck places it at a marked spot in a certain position so that one of the AGVs can pick it up and deliver it to the in-buffer at the printer. Once at the printer the paper roll is prepared to be spliced to the paper currently running in the printer (read more about this under Splicing). When the changeover to the new roll occurs several things can happen. If the incoming roll is part of the same order as the one before it, then no setup time will be required since the properties are the same in every aspect, including the print on the sleeves (read more about this under Support processes). If the incoming roll is from a new order, there will at least be a setup needed for changing the sleeves. Often the new order has a few properties that differ from the one before it (read more about this under Order attributes). The printer will then need to be adjusted accordingly. If a new print technique is used, the cylinder that places the paint on the sleeve, called an anilox, needs to be changed. This is a lot easier to do in the new printer, and because of this the old printer sticks to only one print technique, called Flexo Process, while the new one can handle both Flexo and Flexo Process. If the material coming in to the printer is for a different package type than the one running, the creasing tool that places marks for folding and openings needs to be changed as well. In order not to waste paper, setup rolls are often used (read more about this under Setup rolls). When the actual printing occurs, the paper is run through the printer and rolled up at the other end. Once the "out-roll" is full the paper coming out of the printer is automatically spliced to a new one. The printed roll is transported by conveyor to an AGV pick-up point. The operator calls for an AGV to pick it up and transport it to the intermediary storage. The intermediary storage for printed rolls will from now on be referred to as the post-print storage. If a printed roll has any deficiencies across all lanes, it will instead be sent to the rewinding (see Support processes) where the deficiencies will be remove before it's sent on to the post-print storage. All information about deficiencies (placement in the product, type, severity etc.) in the products is kept track of in a database. At the AGV's drop-off point a manned lift truck places the printed roll in storage. Figure 4-3 shows a flow chart describing the flow of paper rolls from the raw materials storage to the post-print storage.

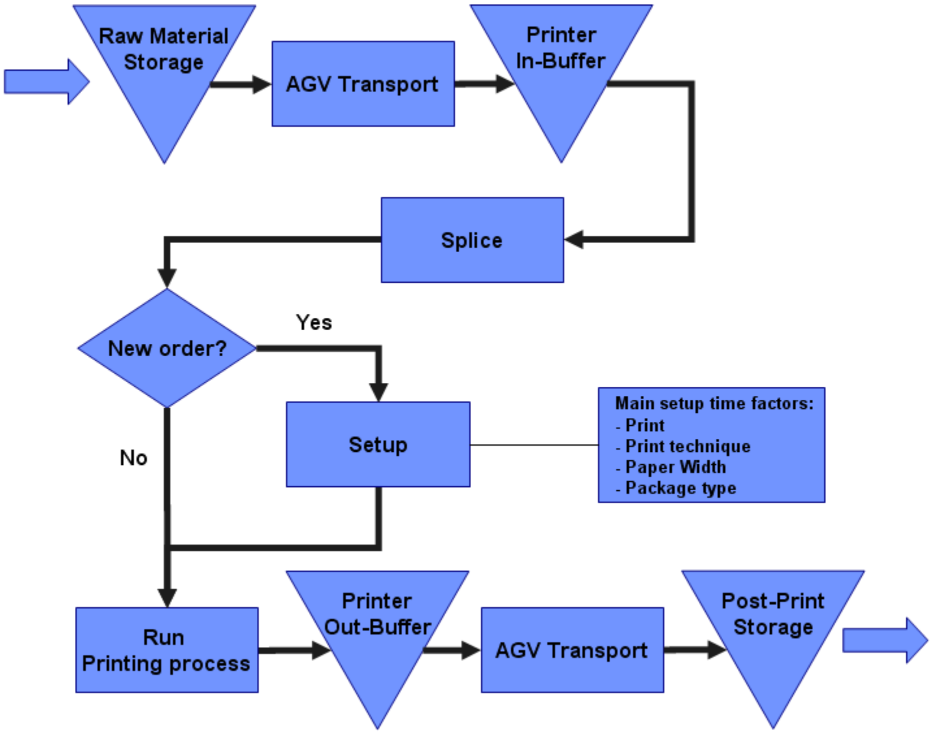


Figure 4-3 Printing flow chart

When an order is scheduled to run to be run in the laminator a manned lift truck places the roll at the pick-up point from where the AGV transports it to its destination. The new laminator handles all packages with exception for one certain type that is handled by the old laminator. This means that the old laminator is off line most of the time, and only started up to handle this particular type. The operator prepares the roll to be spliced in the same manner as at the printer. If the properties of one roll following another are the same no setup will be needed, even if the print is different. Major width changes and changes from a lower quality of plastic to a higher one however usually induces setup time, sometimes using setup rolls. A number of other properties forces the laminator to change its setting, but does not usually require the machine to stop. Rolls come out of the laminator the same way as at the printers. An AGV comes to pick it up and transport it to one of two locations. If the area by the slitters, from here on called the slitter storage, has free room, the roll will be transported there. Otherwise it will be transported to another intermediary storage in the same building as the post-print storage. This storage will from now on be called the post-lam storage. At the post-lam storage, handling is done the same way as at the post-print storage. Figure 4-4 shows a flow chart describing the flow of paper rolls from the post-print storage to the post-lam or slitter storage.

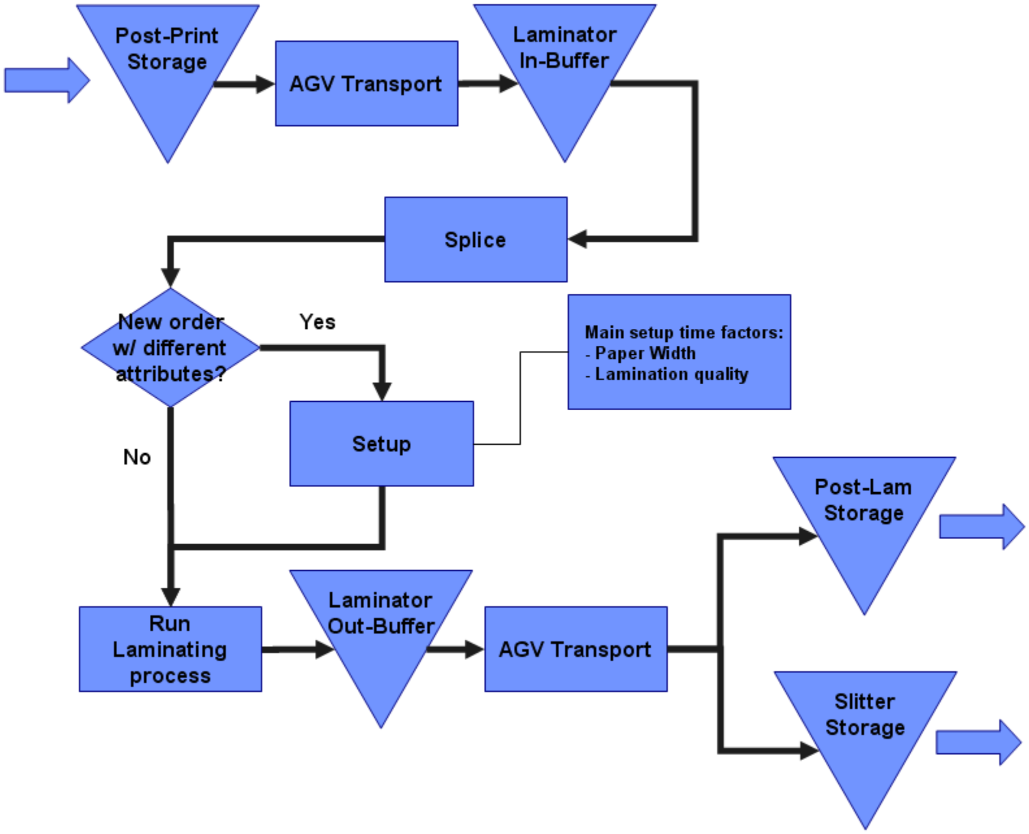


Figure 4-4 Lamination flow chart

Rolls that are placed in the post-lam storage are always sent to the slitter storage before entering any of the slitters' buffers. Rolls are then transported by hand truck to the in-buffers. The number of slitters running at the same time is dependent on how large volumes are run in the factory. There is however always a minimum of two slitters running. When this configuration is active the newest of the three machines will always be running, while the two others take turns, switching approximately every second week. Changing between rolls at the slitters takes considerably longer than at any of the other machines. Splicing requires the machine to stop both for the in-rolls and the out-rolls, and fixing the paper is done manually. The main parameters that induce setup time, except for the splicing, are the number of lanes of the order, and the width of these lanes. The actual slitting process can be quite a lot of manual work as well, especially if there are many defects in the paper. Defects that span across all lanes on a roll are cut away in the slitter. The finished one-lane rolls are automatically put on a conveyor and transported away. If there are any defects on these rolls the rolls are sent to doctoring to be trimmed. Otherwise the rolls go directly to a machine that places the rolls in a plastic vacuum package. The packaged rolls are then stacked on pallets by a robot arm. The pallets are then transported on by conveyor to an AGV pick-up point. The AGVs transports the pallets to the finished product storage. From here trucks come and get the rolls for shipping to the customers. Figure 4-5 shows a flow chart describing the flow of paper rolls from the post-lam and slitter storage until they are sent to the customer.

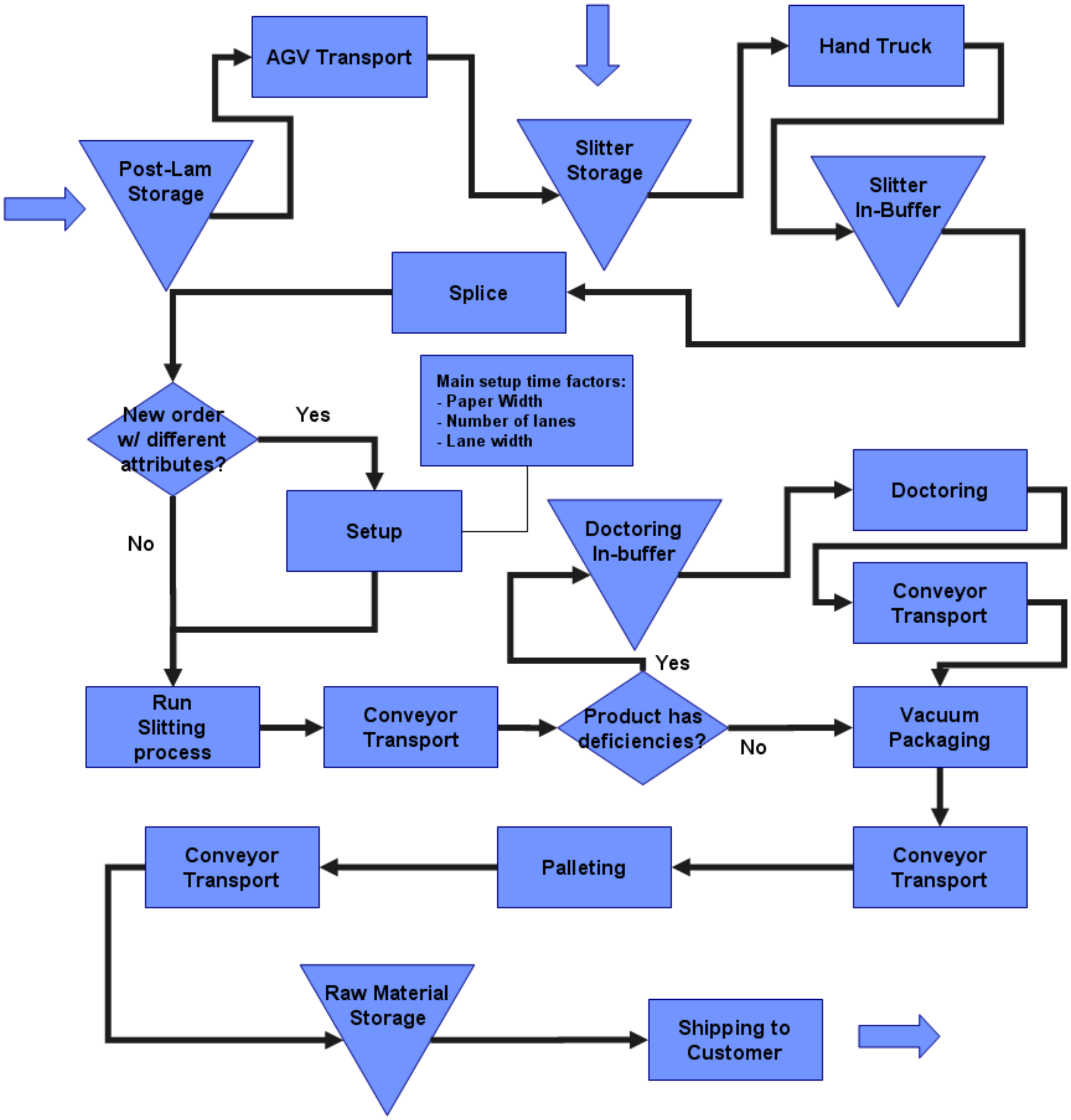


Figure 4-5 Slitting and finishing flow chart

4.2 Planning and sequencing

Planning of what orders are to be run at each machine is done at the planning department. The aim is to run orders in such a sequence as to minimize setup times overall and still keep delivery accuracy high. The laminator is the most expensive machine to have standing still in the factory. Therefore the planning for the printer is also done in accordance with what suits the laminator, and orders are run in the printer approximately two days before they’re planned to be run in the laminator. The sequencing for the laminator (hence the printer as well) is based mainly on paper widths. By mostly running orders that have similar widths next to each other the laminator can reduce setup times significantly. The setup times can also be reduced by sticking to the same quality of plastic used to laminate the insides of the package. Therefore the planning department have devised a schedule of what types of orders should be run after each other. The schedule goes full circle approximately once a week and goes from wide paper widths to narrow ones, and then back to wide effectively removing setup times due to width difference. For each width, types of plastic are also put close together in order to further reduce setups.

Planning for the slitters does not follow any sequencing rules other than to let the first order coming in be the first to run in the slitters.

4.3 Setup rolls

Sometimes the first few hundred meters of print or lamination after an order change will go to waste. In order to not waste valuable paper, a lesser quality of paper will be used. This paper comes on shorter rolls, called setup rolls. Paper that has been used for setup cannot be used for anything else and is considered to be waste.

4.4 Splicing

In order for the machines to have as high utilization as possible, and thereby creating maximum value, they should be standing still as little as possible. Between two rolls in the printer or the laminator, when all the order attributes relevant to the machine are the same, the machine doesn’t have to stop. Instead something called a splice is performed. Splicing is a technique that allows the first part of the paper in the machine to be standing still while the printing part is still running. This is done with the help of several cylinders that move apart, gathering up a length of paper before the splice occurs, then, when the part of the paper to be spliced stops, the paper that’s been gathered can be released by moving the cylinders back together. When the splice is done, the part of the paper where the splice was done starts moving again and the cylinders move back out to gather new paper in preparation for the next splice.

If the properties doesn’t match between the two rolls (i.e. a new order with different attributes) the machine will have to stop. The splice is still done in the same manner though, and it usually occurs before the machine is stopped.

4.5 Support processes

The support processes are those that only affect the flow of the main process indirectly.

Pre-print is one of the support processes. Here the plates used in the printer are prepared with the picture that’s to be printed. The plates are for one use only, so even if a picture have been printed before a new plate will be needed if it's to be run again. The plates are mounted onto what's called sleeves, which can be placed on the printing rollers in the machine. Sleeves need to be changed each time a new order is run in the printer.

Paint preparation is another support process for the printers. The printers use 4 standard colors and 3 non-standard colors. The non-standard colors need to be prepared according to specification. This process is out-sourced to another company, but the work is still done inside the factory. Some orders use non-standard colors, but not all; therefore paint only need to be replaced sometimes when new orders arrive to the printer. The changing of non-standard colors can usually be done while the printer is running and doesn't usually create any additional setup times.

Doctoring is another support process that has a quite big effect on the actual performance of the factory, since doctoring is sometimes needed to be able to deliver the finished products. In doctoring deficiencies are removed. This is done after slitting, and only to some of the products that come out. There are several stations where doctoring is done.

Rewinding is where rolls that have been printed can be sent if there are deficiencies across several lanes. This is also the place where the setup rolls are prepared.

Maintenance and repairs are done when needed urgently as well as according to schedules.

4.6 Planned downtime

The machines in the factory have planned stops each week to perform maintenance. Machines can also be taken off-line if they are momentarily starved (i.e. one machine outruns the one feeding it). The machines are staffed almost every hour of the day the entire week, with a few exceptions during the weekends, when some machines are off-line during the night.

4.7 AGVs

AGVs, Automatically Guided Vehicles, are the main transportation device used inside the factory. The AGVs have the ability to lift one roll or pallet at the time, and they are preferred over manned fork lifts due to the fact that they can handle the goods with extreme care without the driver getting impatient. The AGVs move quite slow compared to manned lift trucks and loading and unloading is done even slower. The slow movement speeds are set that way in order to not hurt the goods. The paper rolls are very heavy, and setting them down too quickly would damage the material, which would induce costs over time.

The AGVs are controlled by a computer system that tells them where to go, and their movements are guided by light sensors on the AGVs and mirrors placed on the walls in the factory. Operators can tell the computer system that they need rolls at a certain location or that a roll needs to be taken away to the storage; the rest is handled by the system.

The AGVs are run on batteries and need charging with certain intervals. The factory has a charging station in an outdoor garage, where the AGVs go when they need charging or when no work is needed of them.

4.8 Order attributes

Each order placed can be described with basically three different attributes. These are quality codes, size codes and order length. The quality code is a four digit number that refers to a set of attributes concerning type of paper, type of print, type of package, what plastic should be used for lamination, if there should be aluminum or not, and if so what quality that should have and so on. The size code is a three digit number that, similar to the quality code, refers to a set of attributes concerning dimensions. Examples of these are width of roll, width of lanes on rolls and number of lanes. Order length is given in meters, and it refers to the number of meters on the rolls going in to the factory, not the number of meters of finished products. The explanation for this is that the order length is an attribute for the production order and not the purchase order.

Quality and size code together contains most of the data needed to produce the order. When these two are put together they're called quality-size codes or just quality-size. A roll of a certain quality-size can have a specific length of material on each roll, therefore the quality-size and the order length together gives what number of rolls will be needed for the order.

Apart from these three orders attributes telling how the production is to be done the order also has an order number for identification and a due date for when it needs to be finished.

Sometimes the amount of material ordered is such that an uneven number of rolls are needed. These orders can let the last, the uneven, and part of the order tag along on another order's roll, just using up a lane or two. This is called co-printing, and production orders that are supposed to be run together with another order are marked with a 'C' in the beginning of the order number.

5 Conceptual model

The conceptual model is a simplified description of the real world system and it constitutes the blueprint for the simulation model. It includes the two printers, the two laminators and the three slitters. Of the slitters only two will be available at the start of the model, but the third one is enabled when storage levels in post-lam reach a certain level. Splicing rolls at each machine, except the slitters, is a separate activity that's done while the machine is running.

AGVs are part of the model, and they follow the same physical paths as they do in the real factory. The AGVs don't have to charge their batteries, but they do however return to the parking area after each job have been completed, unless they are closer than any other AGV to a new job that has come up.

The planning and sequencing of production orders at the different machines should be done in the same manner as the real factory. By delaying orders in post-print the lead times of the model will resemble those of the real factory where orders are run in the printer approximately two days before they're run in the laminator.

Setup times will be modeled from historical data and will occur when attributes differ between orders in such a way that a setup is needed. Order attributes included in the model will be the quality-size, package type, lamination quality, paper width and number of lanes. These are the ones that are needed to keep track of the main setup requirements.

Planned downtime, such as maintenance and off shift hours, will be included. Breakdowns will be included as well, using historical data to find distributions for time before failure and time to repair.

In order to get a model of suitable complexity parts of the factory are cut out.

The train arrival with incoming paper rolls is considered to be outside the system of this model. Delivery accuracy for this activity of course has an effect on the factory, but assuming its high enough, modeling this part of the factory will not add value to the model.

The manned lift trucks inside the different storages are not modeled. Rolls are assumed to always be in place when AGVs arrives to pick them up.

AGV charging is assumed to not affect performance at all, since there is a surplus of AGVs allowing them to take turns charging keeping availability constantly high. AGV charging is therefore not modeled.

Operators and their tasks are not modeled in detail since that would require more time for the project, and not necessarily increase the models accuracy. The operators are however modeled in the way that the time their work takes during setups is accounted for.

Pre-print is a process that, though vital to the factory functioning correctly, can't speed up the production by further optimizations. Therefore the pre-print is excluded from the conceptual model and considered to have to have 100% delivery accuracy.

Paint preparation is another part excluded from the model. The argument for this is the same as for pre-print. Paint preparation is a separate system that if it fails affects the performance of the factory, but can't help the performance past a certain level. Paint preparation is excluded for this reason, and is assumed to never slow down the printer.

Doctoring has more of a direct effect on the factory's performance. Parts of some orders need to move through this step before the orders can be delivered, which means it sometimes has a direct effect on lead time for orders. However adding doctoring means adding deficiencies and waste to the model, which would make it a lot more complex. Given the time and resources the scope has to be limited. Doctoring, material waste and deficiencies are not part of the conceptual model.

The handling, storage and transportation of setup-rolls can be considered to be a different system. These activities have today no apparent slowing effect on the printers or laminators and are therefore not included in the model. The setup times during which the setup-rolls are used are still part of it though.

Not including waste, deficiencies or setup rolls means that rewinding is superfluous. This is because it only deals with removing deficiencies and preparing setup rolls. These activities do not directly affect the primary flow of products in the factory. Therefore it is excluded.

The production steps vacuum packaging and palleting that occur after the slitters is not included either. The production times for these steps are negligible in comparison to the slitters, so the production time for the slitters and production time for the slitters plus the following steps is approximately the same.

Co-printing has also been excluded from the conceptual model since it severely increases complexity in the model, but doesn't necessarily increase accuracy of the model.

Figure 5-1 shows an overview of the conceptual model.

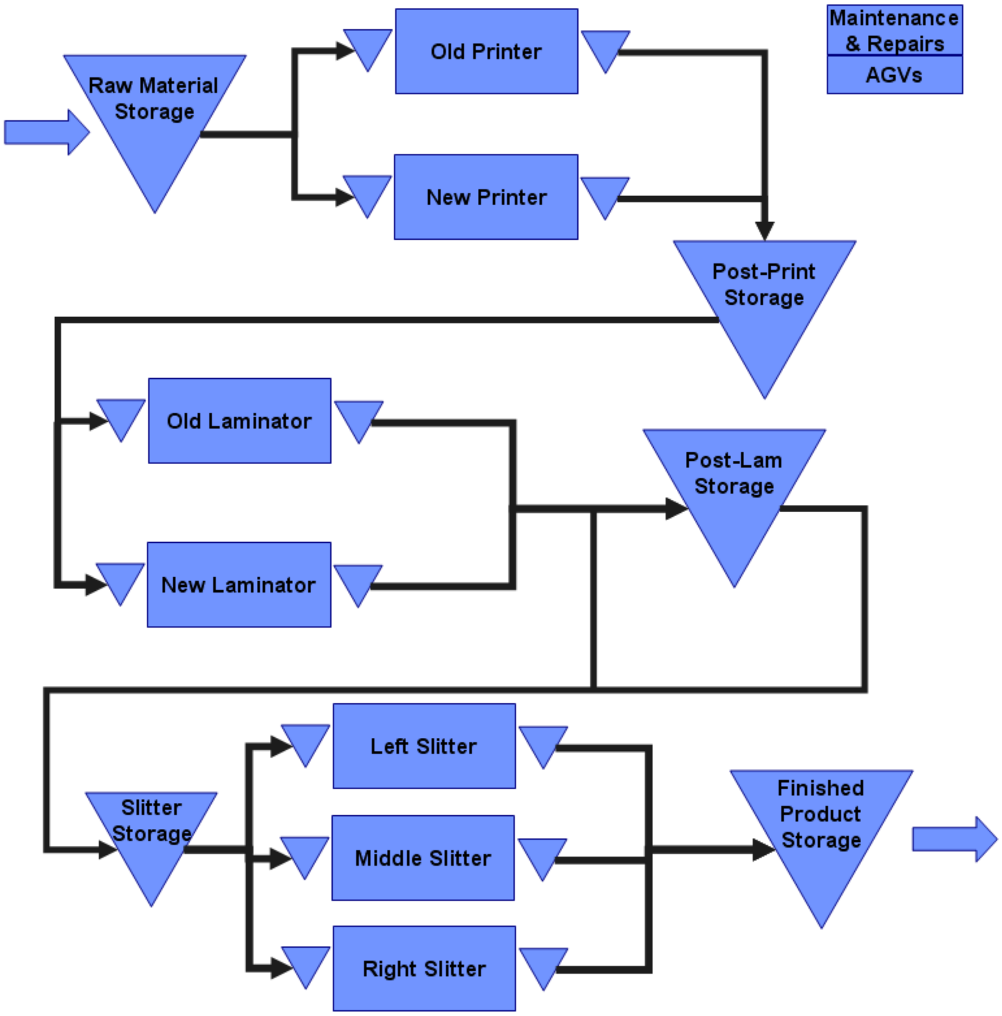


Figure 5-1 Conceptual model overview

6 Empirical data

6.1 Requirements

To adequately model the system the following data is required:

- *Order data.* The order data is needed to be able to generate orders with the correct attributes. Important attributes for the order data are: quality and size codes, order ID, the start date of the order and the length in meters. The quality and size codes can then be translated into more attributes that are needed to determine which setups need to be performed.
- *Production logic.* This data is needed to model the factory flow and how the machines operate when they are fed with orders with different attributes. What is needed is an understanding of what the attributes of the orders mean and a way of translating them into different setups. For example, in the printer a change in the order-number will require another print and trigger the setup “12-01 Sleeve change”. All of the relations between the attributes and setups need to be understood and categorized. Another logic that is needed is what determines the sequencing of orders.
- *Setup times and runtimes.* The duration of each type of setup is required as well the time it takes for a length of paper with certain attributes to be processed by a machine.
- *Scheduling and unplanned stops.* The time when each machine is operational, under maintenance of scheduled offline is required. Unplanned stops are also an important factor of the production and the mean time between failure (MTBF) and the mean time to repair (MTTR) is needed for the more common disturbances.
- *Layout and AGV routes.* To be able to have a nice graphical representation of the system as well as modeling the transport times within the factory an accurate blueprint of the layout is needed. The AGV travel routes as well as their speed are also required to model travel times.

6.2 Method of collection

The data was collected using a number of different methods described below.

6.2.1 Direct observation and unstructured interviews

The authors spent five days in the factory to observe and get to know the system. During this time the entire process flow was observed and unstructured interviews with the machine operators were conducted. These interviews were done to clarify and explain what the authors had observed as well as to answer technical questions about setups and provide time estimates for setup and run times. These interviews also provided the time schedules for each machine and when they were scheduled for maintenance. The week provided most of the logic needed to implement the model as well as a deeper understanding of the system. The size of the buffers and intermediary storage's were also observed.

6.2.2 Structured interview

A structured interview with a process manager at the factory (personal communication, 2010-08-02) gave a deeper understanding of the sequencing in the factory as well as a schedule that shows the sequencing of the printer in detail.

6.2.3 Old master thesis report

The master's thesis by Ferrada and Omeragic (2010) contains information about setup times and unplanned stops. The data in the report is based on historical data from the factory's logging system which logs all activities in the factory. The report also contains probability distributions for setup times and breakdowns. Since their model from 2009 did not include the slitters no information about them is included in the report.

6.2.4 Historical data

The factory planning department provided historical order data. The data based on two months of production in the summer of 2010, showed production events of at what date an order had been started in each of the machines. It also provided the order-number, quality-size code and the length of each order. Further historical data was also provided on which average speed each quality-size code had been run during the first half of 2010. This data also contained a large list of quality-size codes needed to translate the codes on the orders to more detailed attributes.

6.2.5 Data capture

Since it was not possible to get data on the AGVs from any other source data capture was the last resort. This was done by clocking the time it took for the AGV to travel a certain distance and the time it took for it to load a roll. Setup times for the slitters were also unavailable and had to be clocked manually. The setups in the slitters are relatively infrequent, and therefore only one observation was made during the time spent observing the process. The AGVs are considered to perform theirs tasks consistently, and therefore only one observation was made for loading and travel speed respectively.

6.3 Unavailable data

There is lack of data concerning the slitters. This had to be alleviated by manually clocking setup times and use operator estimates. There is also no data on breakdowns for the slitters.

There was no data on how the buffer levels change over time which would have been good to use as a validation parameter. There was also no data on the exact contents of the buffers which could have been used to setup the system as it was at the start time of the simulation.

The time resolution in the historical order data is days, which would have been better to have in hours from a validation perspective.

6.4 Input data analysis

Raw data can rarely be used directly, and needs to be analyzed, sorted and formatted before use in a model.

6.4.1 Historical order data lead times

The historical order data was provided as three excel sheets, one for each machine group, which contained data on which orders had entered each machine and at what date. In the order data there was a number of co-printing orders which had to be taken out as they are physically part of another order and should not be treated as a separate entity. There was also a need to restructure the data in order to follow a given order through the three machine groups. From the newly structured data it was possible to calculate the lead times between each machine for each order. The results can be seen in Figure 6-1, 6-2 and 6-3.

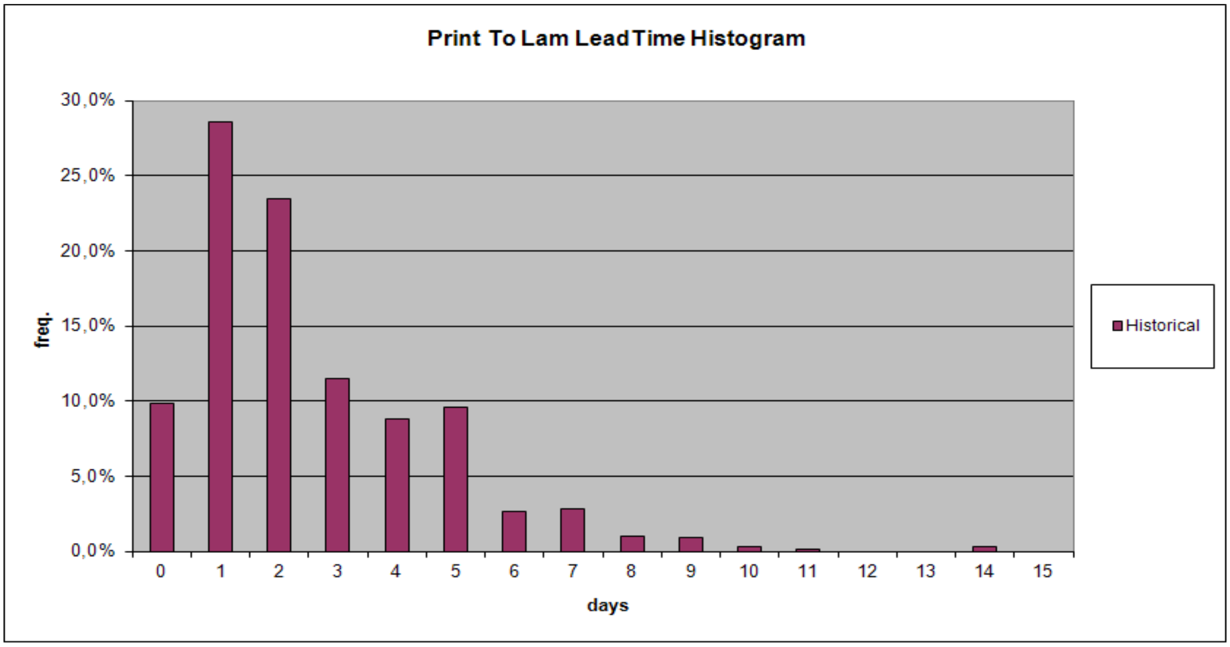


Figure 6-1 Historical printer to laminator lead time histogram

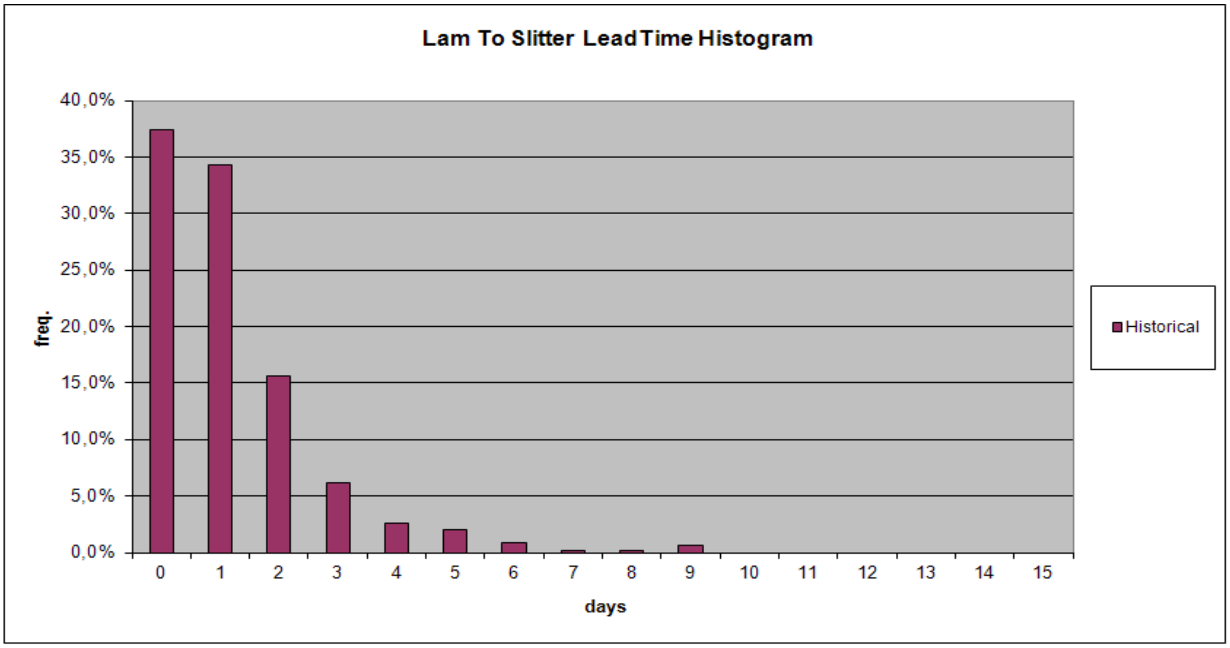


Figure 6-2 Historical laminator to slitter lead time histogram

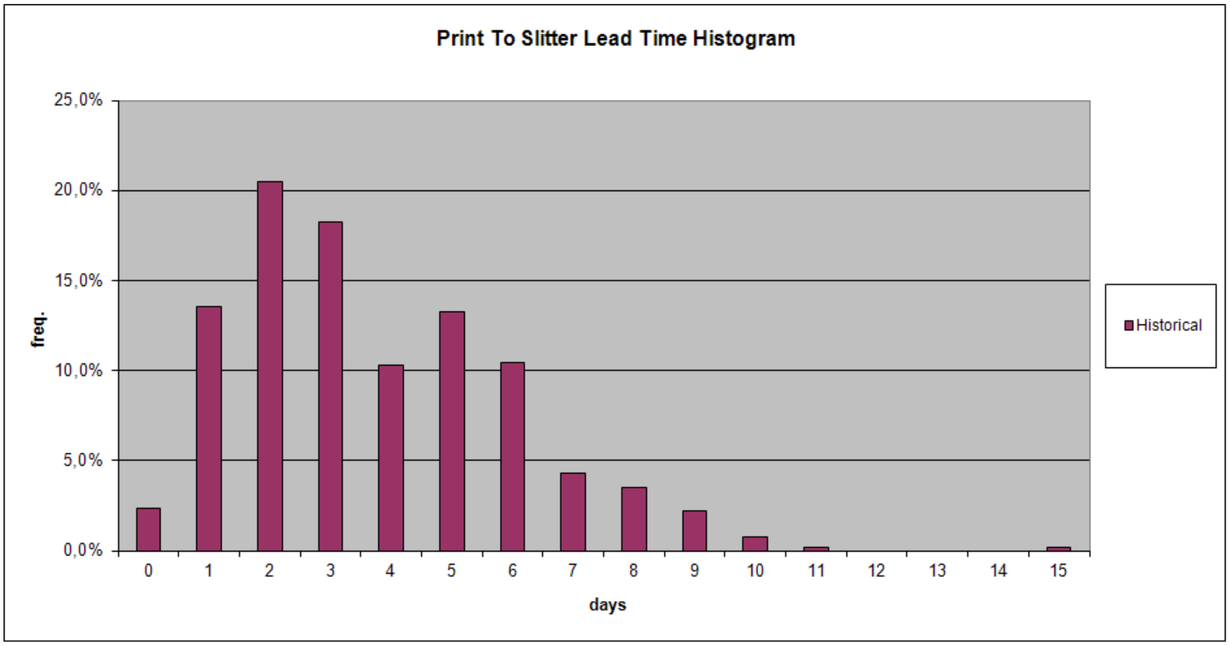


Figure 6-3 Historical printer to slitter lead time histogram

The lead time data serves as a means of calibration and to validate the model.

6.4.2 Throughput

The data doesn’t describe the throughput for the machines or the system explicitly but an approximation can be made by looking at the volume that goes into each machine. Table 6-1 shows the sum of the lengths of the orders that were started in each machine during the period.

Table 6-1 Production started in each machine and machine group

Group	Machine	Million meters started	Million meters started	Million meters / month
Printers	Old	9.31	33.13	16.57
	New	23.82		
Laminator	Old	0.41	32.76	16.38
	New	32.35		
Slitters	Left	11.65	34.97	17.49
	Middle	18.67		
	Right	4.65		

What is interesting about this data is that it shows more volume going into the slitters, the final step, than was started in the printer during this period. This can only mean that the buffers during this period were quite large and that part of the work during the two months was to finish up older orders. This poses a problem since in order to get the model to produce the same results proper buffer levels must be set at the start of the run. This would require having data on what was in the storage's at the time and since that is not available a warm-up period will have to do.

6.4.3 Distributions of quality-size in the order data

During the period of interest orders with about 100 different quality-size codes were produced. Some quality-size codes were more common than others and there was also a big variance in the order lengths. This data was analyzed in order to be able to create input date that was not historical data but had the same properties and probability shape. A table with the frequency of each quality-size code can be seen in Appendix 1. The distributions for length of orders from each quality-size code have been approximated with triangular distributions using the minimum, maximum and average length for the orders from each code. The length distributions can also be seen in Appendix 1.

6.4.4 Verification and validation of input data

The probability distributions for setup times and breakdowns are taken from the old report (Ferrada and Omeragic, 2010). The data for setups and breakdowns comes from the company's own reporting system and should be considered reliable. The probability distributions based on that data has been checked using the goodness of fit tests Chi-square, Kolmogorov Smirnov, and Anderson Darling. The data has also been checked for auto correlation.

The machine speeds and order data are also from the company's own reporting system and should be considered reliable.

The observed and data captured values are too few to be able to perform goodness of fit tests. And have to be assumed to be constant.

6.5 Problems encountered

Due to the size of the company and the fact that the information needed was owned by departments outside of the one who commissioned this study, it was difficult to get access to people and data. Much of the data is also sensitive as it describes the performance of the factory and few people have access to the feedback database. The ideal would have been to get full access to this database in order to get updated data on setups and breakdowns. Since this was not possible the old data will have to suffice.

7 Modeling

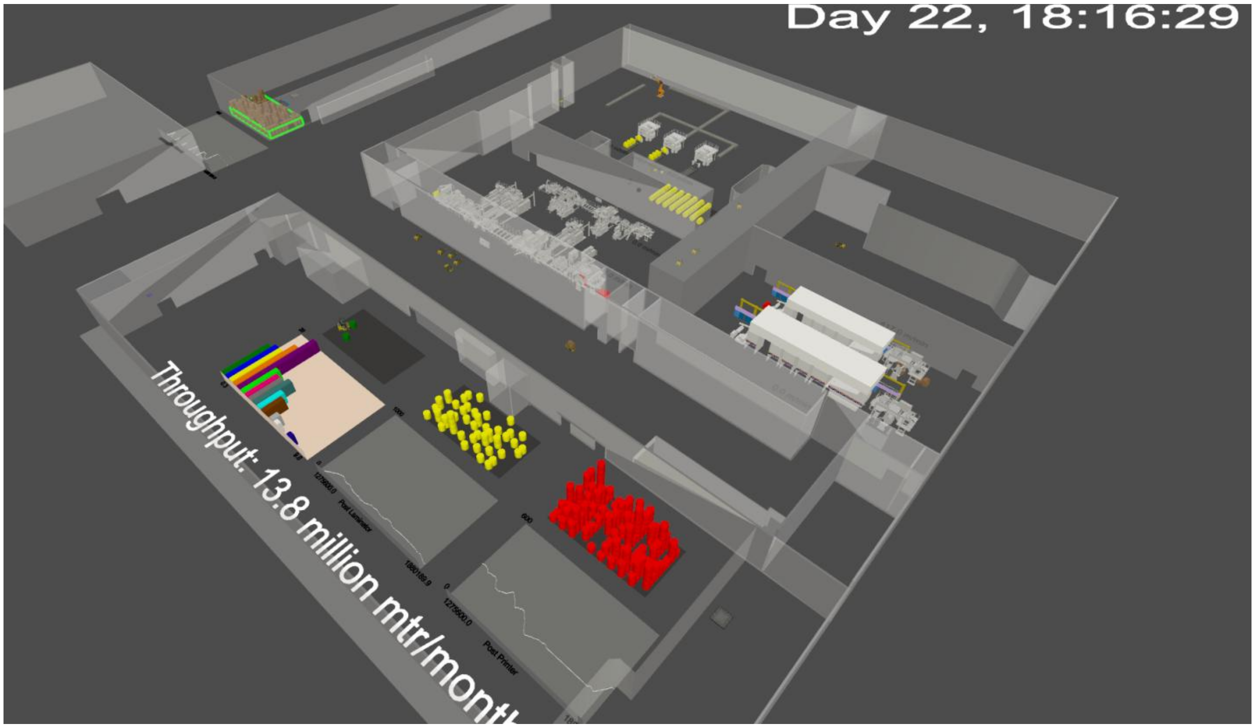


Figure 7-1 Screenshot from model

7.1 Model translation

The actual model, shown in Figure 7-1, is done using the software Flexsim. Flexsim is based in Utah, USA, and their products are used in different businesses around the world.

Flexsim’s user environment is in 3D and all models are basically constructed by placing 3D objects in an empty space and connecting them together in different manners.

The model of the system is a discrete event simulation model. It’s dynamic, meaning it is run over time, not just one calculation. And it’s stochastic in that it uses stochastic distributions to simulate the times different activities take.

The entities of this model are the paper rolls, the machines, the buffers, the storages, the AGVs, the entry points and the exit points. All these correspond to different object types used in Flexsim.

The paper rolls are modeled as so called flow items. The flow items are what flow through a model in Flexsim, and usually the flow items are the products that are being produced. The flow items are created at the entry point, or source, every 24 hours. In this model the source first creates one flow item that represents the production order. What production order is created is decided by an empirical table of orders that were run in the printers in the real factory. By creating the production orders that were run in the printers the model gets the same input as the real factory with exactly the same variations in order frequency. The production order contains the attributes quality-size, order number and order length in the form of so called labels. The production orders move into another object called a separator. At the separator the production order is separated into the correct number of rolls, also flow items, and all the rolls are given their appropriate attributes regarding plastic quality, package type, print type, paper width, number of lanes, how much paper is on each roll and how fast it can run in each machine.

When the paper roll generating procedure is done the flow items are placed in the raw materials storage. In this model all storages are represented by rack objects. From here flow items are pulled to the printer in-buffer in the same manner as in the real factory. To make this happen in Flexsim a somewhat unconventional method had to be used. Each time a flow item enters the storage a custom function, in Flexsim called a user command, is run that checks if each buffer has allocated the maximum number of flow items it can hold, or if there will be room for one more, in which case it will automatically be allocated to that buffer. Allocation is used because once a flow item has been pulled it doesn’t count as occupying space in the buffer until it has been transported there and unloaded into the buffer. If there is no room for one more the flow item will wait in the storage until it’s allocated. The other way of allocating flow items to in-buffers is when a flow item leaves the in-buffer. When this happens another user command is run that reads a row in a vector, which is placed in a so called global table. The vector contains the sequence used at the planning department, with the attributes paper width and package type. The first flow item found in the storage that has the same properties as on that row gets allocated to go the in-buffer in question. When searching for appropriate flow items the oldest flow item will be checked first. If all the flow items that have the attributes from the first row read are already allocated, or if none where there to begin with, the storage is checked for flow items that have properties that correspond to the next row of the vector. This goes on to the last row of the vector after which the first row is read again. The row number is always stored in a label on the buffer, so that the next time the user command is run it starts where it last stopped. For the flow items to actually be pulled the logic in the pull requirement code says that only flow items that have been allocated to the in-buffer in question can be pulled.

After it’s been allocated and pulled an AGV picks up the flow item at the raw materials storage and transports it to the printer in-buffer. AGVs are represented by Flexsim’s task executer object. The AGV closest to the pick-up point will be the one to go for it, unless it’s busy with another task. An AGV that doesn’t have a task returns to the AGV parking area.

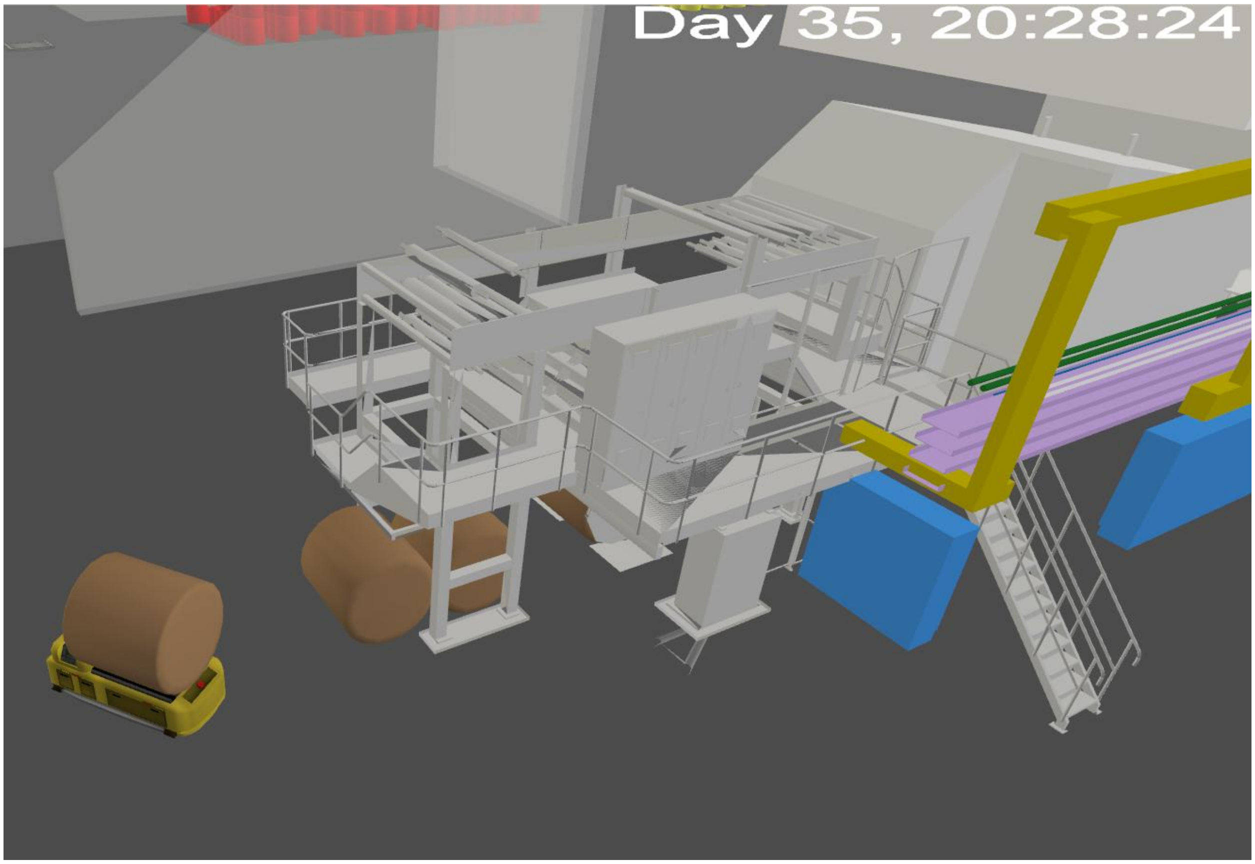


Figure 7-2 An AGV is delivering a paper roll at the printer

The printers' in-buffers act as a FIFO-queue and are represented by a queue object in Flexsim. The same goes for every other buffer in the model.

Before a paper roll enters the printer it has to first be prepared for splice and then spliced on to the paper running in the machine. The splice preparation is modeled with a separate processor object. Once a flow item in the printer is done, the flow item in the splicing object can move into the printer. The splicing object is now free and lets in another flow item.

Printer, laminators and slitters are all modeled using the processor object with a capacity of one and a setup time and a processing time that are set according to different criteria. When a flow item enters the printer, if the flow item before it had a different value on the label for order number, a setup will be triggered. If this is the only printer related attribute that differs a sleeve change will occur, or rather a time delay from a distribution that mimics the time for the activity sleeve change will happen. If other attributes differs other setups will be required.

Once the setup stage is done the printing process runs. The time this will take is calculated from the length of paper on the roll and the average speed this quality-size can be printed in, both of which are labels on the flow item.

Printed flow items are put in a queue object representing the out-buffer before they are transported off to post-print. Flow items are delayed in the post-print storage in order to mimic the way production is scheduled in the factory, where rolls are printed approximately two days before they are laminated. On entry to the post-print storage a time delayed message is sent to a dummy object, which when it's delivered changes a label on the flow item that allows it to be pulled to the laminators in-buffer. Items reaching post-print check a global table to see if any other rolls from the same order have already set a delay time. If not, i.e. if the delay time in the table is set to zero, then a number is drawn from an exponential distribution and put in the table. The same number is also used for the time delay on the delayed message. When another flow item from the same orders enters post-print it will use the same delay time as the one before it, thereby grouping flow items from the same orders so that they are released simultaneously. If the storage is empty when a flow-item enters, an exception is made which allows it to be pulled immediately.

The flow items are allocated and pulled to the laminators using the same kind of logic as when pulling from the raw materials storage to the printers. However, the pull logics are disconnected from each other so they don't have to pull from the same rows.

Both laminators have a separate object that models the splicing before flow items enter the actual machine. This is done in the same manner as for the printers.

Just like for the printers, flow items that enter the laminator are examined, and if their labels for paper width or plastic quality are different than what the machine is currently set to, a setup will occur. After setup the actual process occurs, and the time this takes is calculated from the length of paper on the rolls and the speed at which the quality size is run. From here flow items are put in an out-buffer.

From the laminators' out-buffers AGVs transport the flow items primarily to the slitter storage, or if that is full, to the post-lam storage. From the post-lam storage flow items are transported by AGVs to the slitter storage as soon as it's possible.

Flow items in the slitter storage are pulled to one of the slitter in-buffers in a FIFO manner. However if an order is currently being run in one of the slitters the other slitters won't pull any flow items from that order, and the slitter running the order will continue pulling flow items from that order until there are no more of them left. Flow items are transported by an operator object, which represents an operator using a hand lift to fill the buffers for the slitters.

Flow items move from the in-buffers to the slitter without moving through a separate splicing object. The real machine always stops to splice a new roll. If a flow item has labels for width, number of lanes or width of lanes that differ from the machines current setting, an additional setup is triggered. After the setup the flow item is processed. The time for this is calculated by the length of paper on the roll and the average speed, including the stops to correct deficiencies, at which this material usually is run.

Flow items exit the slitter objects and are put on a conveyor. The conveyors from the three slitters merge to one conveyor that leads the flow items to a robot object. This is more of a visual feature to show that the rolls are handled by a robot than an actual model and the robot is set to be so fast that it never decreases the throughput of the factory. The robot puts the flow items down on another conveyor which leads to an out-buffer for all the slitters.

From the slitters' out-buffer AGVs transport the flow items to the finished product storage. The finished product storage is the end of the system according to the conceptual model. Any measurements on the model are made up to the entry of flow items into the finished product storage. In the model however a manned fork lift transports the flow items to a loading bay to visually show that the rolls are shipped after they've been put in the finished product storage. The loading bay is a so called sink. This is where the flow items leave the model.

All of the machines have planned stops due to maintenance and off-shift hours. This is modeled using the time table function that exists in each processor object.

Except for the slitters all the machines in the model can break down. Breakdowns are modeled using the MTBF/MTTR (Mean Time Between Failure/Mean Time To Repair) functions in the processor objects.

7.1.1 Visuals

One of the big advantages of modeling in Flexsim is that the visual representation of the model is created simultaneously with the actual model and doesn't require that much extra work. For this model a CAD layout of the factory was used when placing the objects. This way travel distances and so on automatically became part of the model.

To make the model more visually appealing for demonstrations at the company some extra work was put into making the visual representation even more accurate. 3D CAD models of the printers and laminators were provided by the packaging company and replaced the standard 3D models for processors used in Flexsim. No model was available for any of the slitting machines; therefore a 3D model was created using Google Sketchup and then put in the model. The walls for each of the buildings were also made using Google Sketchup and then put into the model in a visual tool object. Examples of the visuals in the model are shown in Figure 7-1 and 7-2.

The visual tool objects were also used to display a series of KPIs (Key Performance Indicators) as was a recorder object. The levels in each of the storages are displayed as line graphs and are updated in real time when running the model. A histogram showing the lead time from entering the raw material storage to entering the finished product storage is also displayed and updated in real time.

7.1.2 Model expandability

Since the model is built in Flexsim, expanding it does not mean it has to be rebuilt. The model is based mainly on processor objects that have been customized to behave as the intended machines; printers, laminators and slitters. These basic building blocks can be used to model a different factory by simply copying them into another model. In order for the blocks to work all the quality-sizes in the orders need to be available in the quality-size-table in the model. The blocks support all ranges of different attributes (number of lanes, paper width etc.) since the code only is based on detecting changes in them, and sometimes the size of the changes. If another make or model of a machine is to be modeled, setup times and running speeds needs to be updated. Other types of setups may need to be added, but this is easily done by following the example of the existing setup types in the code.

In order to try out other production order sequences the arrival table of the source object needs to be replaced. This can be considered a simple task as long as the correct table format is used.

All the code in the model has been commented and should be fairly easy to understand for someone who's used to working with Flexsim.

In addition to this it's possible to create a custom library of objects containing the customized processor objects. This would make the machine models available for all models without the need to copy them from the original file.

7.2 Verification of the model

The model has been verified using the methods described in chapter 3. Flexsim is designed so that it imposes structured programming since the code for different events is tied to the specific objects in the model. The model was created using pair programming which reduces the risk of mistakes and produces better code. Before the coding phase started, flowcharts of the process had been created and they were subsequently implemented into code. Each line of code in the model is commented and the variables have appropriately chosen names. Some of the model has been developed with the aid of a consultant from Flexsim and he has also checked the rest of the code. During testing, the animations and the input output relations have been monitored to detect errors and the debugging tool has been used for critical parts of the code to step through each line of the code.

7.3 Validation of the model

7.3.1 Face validity

The model has been shown in depth to a number of people knowledgeable about the system. The model has also been presented to the entire Development and Engineering department during a company event.

7.3.2 Validate model assumptions

At the end of the week in the factory a preliminary conceptual model was created and shown to a person knowledgeable about the factory. When new assumptions have been made the appropriate people have been consulted when found and available.

7.3.3 Compare model input and output transformations

The only comparable input output transformation, given the data, is comparing the lead time distributions of the historical data to the one produced. The validation data was produced by inputting the historical order data into the model and comparing it to the historical outcome. Since the system is non-terminating, the model requires a warm up period. The warm up period in these runs was set to 2 months, equally long as the time period simulated. This was done so the order generation list would be back at the beginning when the actual run started.

In Figure 7-3, 7-4, and 7-5 the historical lead time distribution is shown along with the upper and lower confidence intervals for the model.

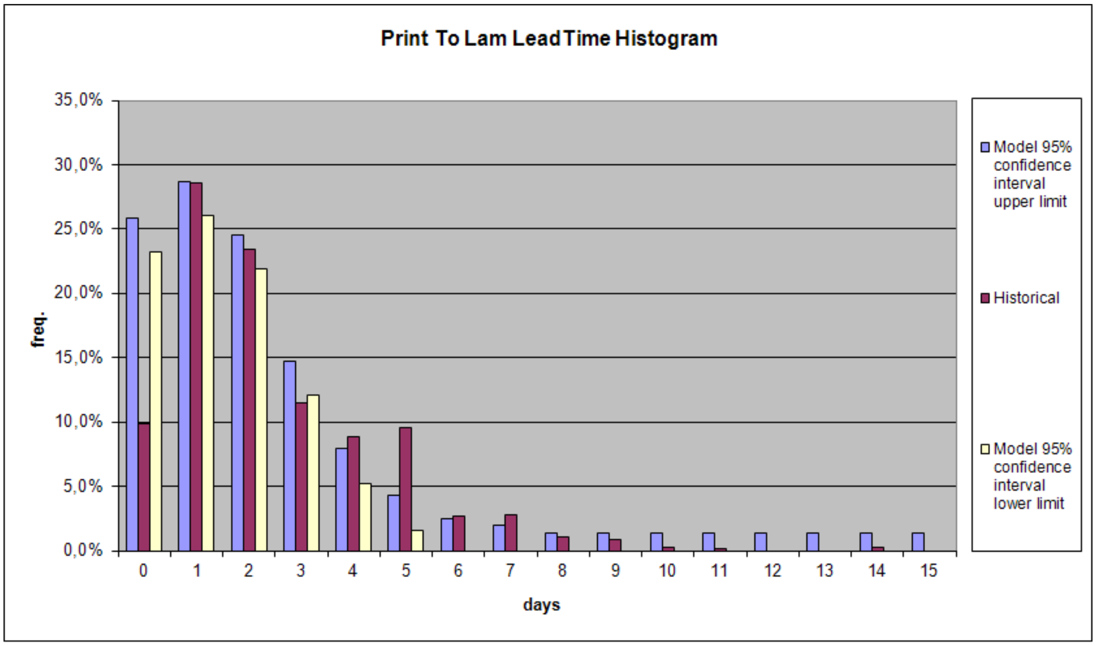


Figure 7-3 Model vs. history - printer to laminator lead time histogram

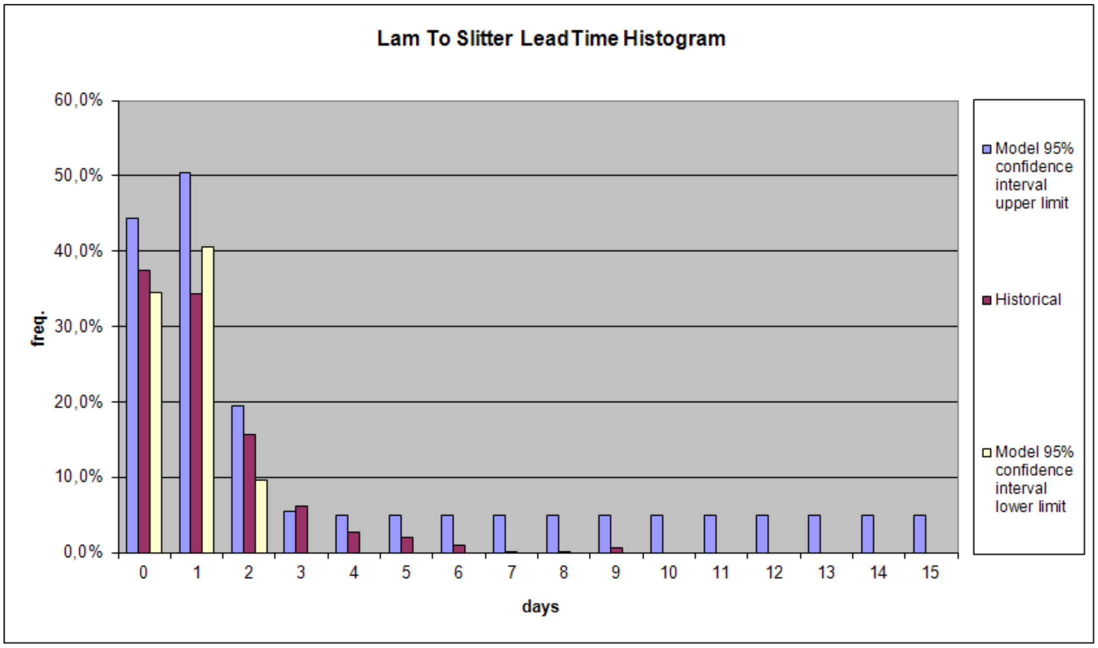


Figure 7-4 Model vs. history - laminator to slitter lead time histogram

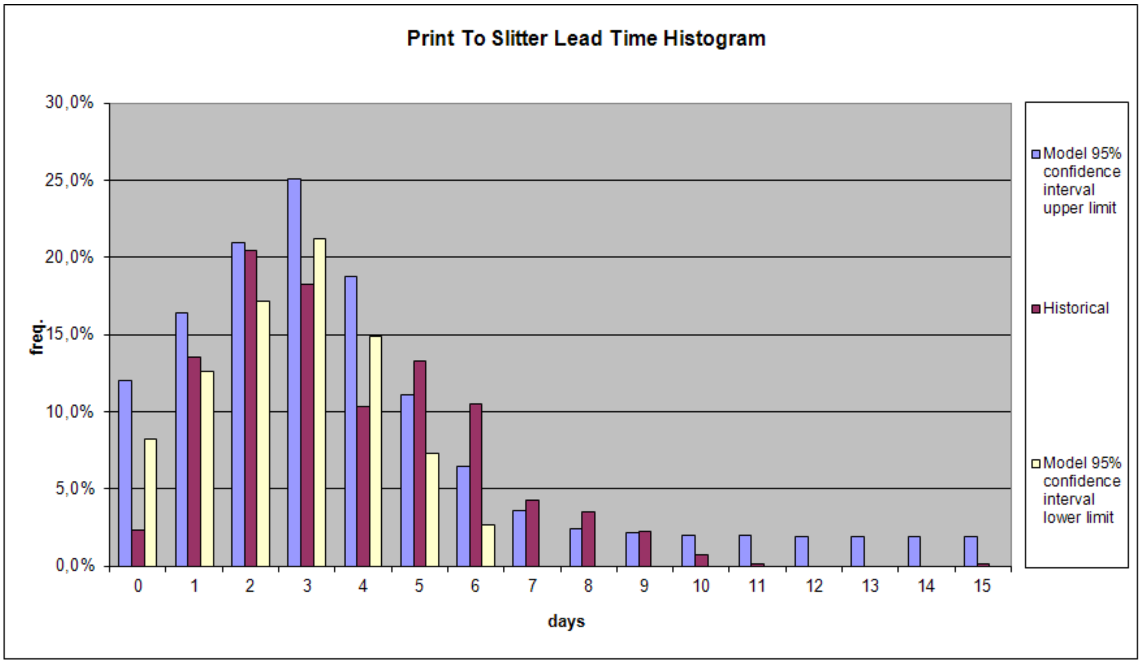


Figure 7-5 Model vs. history - printer to slitter lead time histogram

As can be seen in the charts in Figure 7-3, 7-4 and 7-5, the distributions are similar but far from identical. This was to be expected given the circumstances. The matching is better for the shorter lead times but as the tail grows the discrepancy increases. The tail in the historical data can be explained by the manual planning in the factory where some orders may be delayed to run together with another one at a later date. Logic for these kinds of delays is not implemented in the model resulting in a shorter tail for that data.

To see how far apart the model results are from the historical data a statistical distribution can be fitted to each data set. Using the statistics software MINITAB the two data sets can be approximated by a Poisson distribution. The results of this can be seen in Figure 7-6 and 7-7.

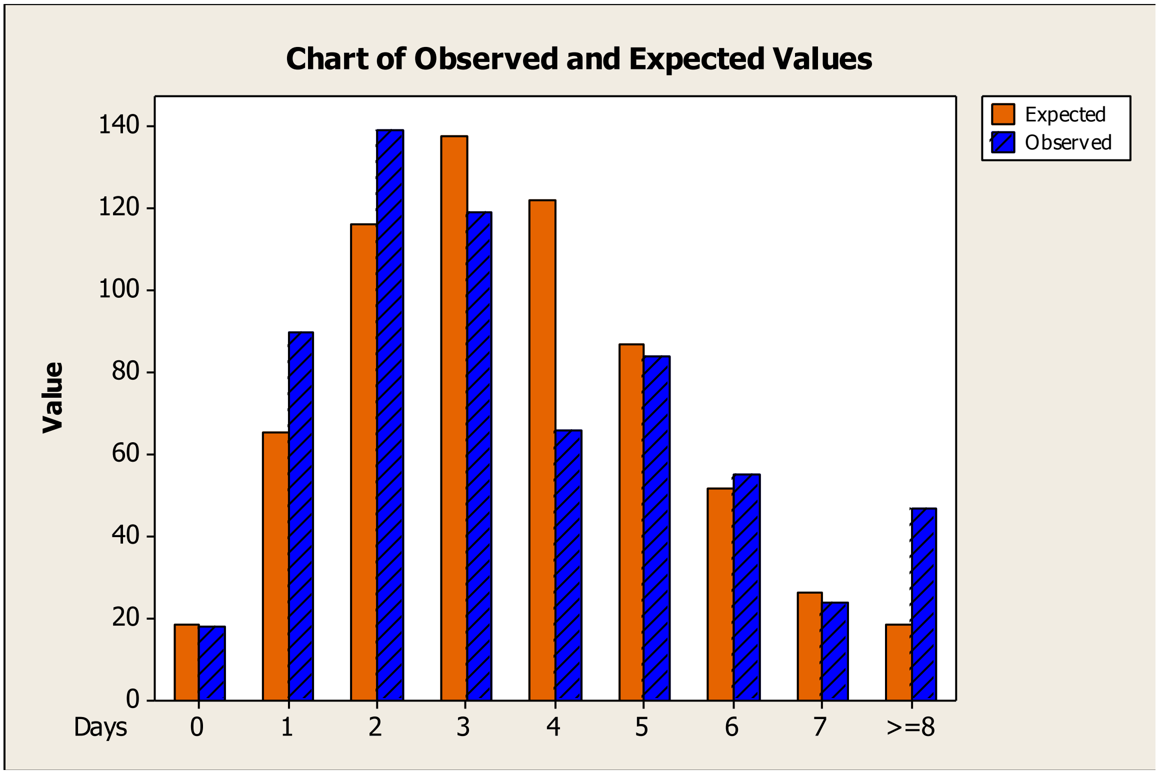


Figure 7-6 Chart of observed and expected values for historical data

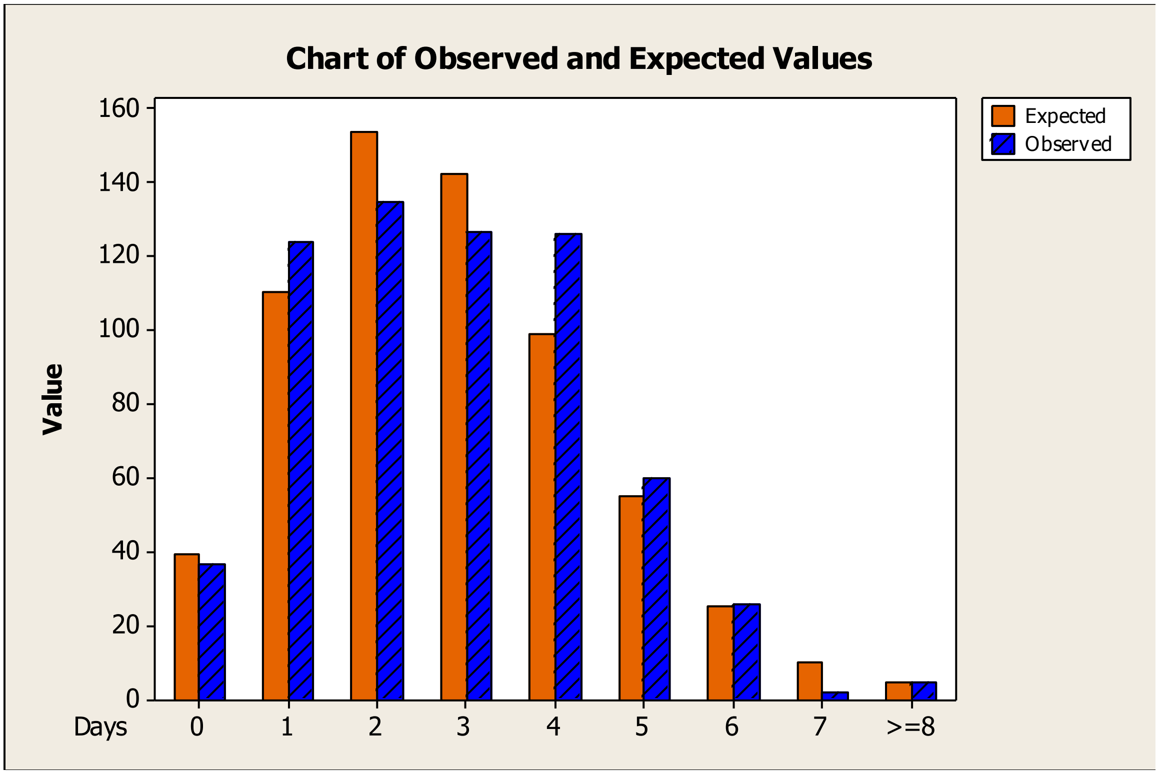


Figure 7-7 Chart of observed and expected values for model data

The historical lead time data can be described by a Poisson distribution with the mean 3.56 days and the model is described by a Poisson distribution with the mean 2.79. The higher mean for the historical data is due to the long tail. If only the data for the shorter lead times are compared the two distributions become much more similar. If only the data up to 4 days are compared the mean values become 2.7 days for the historical data and 2.6 days for the model.

7.3.3.1 Throughput comparison

Although not as relevant it is still interesting to compare the throughput in the model with the meters started in the slitters in the real factory. Table 7-1 shows a comparison between these two values and how many meters that were started in the printers during the same period.

Table 7-1 Throughput comparison

Real factory – started in slitters (Million meters)	17.49
Model – throughput (Million meters)	15.74
Real factory – started in printers (Million meters)	16.57

The difference is quite large and this can be attributed to the fact that the data for the model is taken from a time period when relatively few orders were placed. The in data used to generate the orders in the model is based on the orders going into the printers. Only 16.57 million meters were started in the printers during the period. However the real factory started production orders for 17.49 million meters in the slitters in the same time frame. The explanation for this is that the factory had large amounts of paper rolls in the storages in the beginning of this period.

Furthermore the two numbers are not completely comparable one to one, since one refers to the meters started in the slitters and the other is the total number of meters completed at the slitters. The real factory's number being higher than the model is expected.

7.3.4 Sensitivity analysis and extreme condition tests

The model has been run with different inputs and the results have been examined. The model responded as expected when increasing or decreasing the arrival rate and when varying the process times.

7.3.5 Conclusion

Validation is not a matter of yes or no but a degree of how closely the model represents the actual system. With the current data the model comes close but is not accurate enough to be used as a planning tool to track on an order basis. It is accurate enough to experiment with the system wide performance and the overall effects those changes incur.

7.4 Problems encountered

The main problem when building the model has been the lack of data. The lack of data leads to assumptions that may not be entirely accurate and are detrimental to the models validity. The mismatch of data from different time periods made it impossible to completely replicate the historical results. The lack of throughput numbers and buffer levels complicated the validation process which had to be done using the lead time distribution as comparable output. Another problem was modeling the sequencing since it encompasses many human decisions and was very difficult to simplify into logic and code.

8 Experimental design

8.1 Type of simulation

The system is of a non-terminating nature, but the data used for this model is collected over a period of two months. Therefore only two months will be used for result collection from the model in any of the cases. In addition to the result collection period a warm-up time will be used to put the model in a state that resembles typical production. I.e. a state where there is flow items in each of the storages and in the machines.

8.2 Scenarios

A total of five scenarios were run on this model of which one is the baseline scenario.

8.2.1 Baseline

The baseline scenario is the as-is scenario of the factory. It is basically the same settings as was used in the validation with the only difference being how the production orders are generated. For the validation the production orders were read of a list, making the orders run in the model exactly the same as the ones that were run in the factory. In the baseline scenario, what production orders are created is decided by choosing from an empirical table of different order types. The probability of choosing a certain type is based solely on how frequent that type was during the data collection period. Once the type of order has been chosen the length of the order is decided by a stochastic distribution based on historical data on how long orders of the chosen type usually are. Production orders are created until the daily requirement of total length of ordered material have been reached.

The baseline is the base model that all the other scenarios are run in. The baseline scenario mimics the model based on historical data, but the inflow of production orders is more evenly distributed than for the model based on historical orders since the baseline is controlled by a daily average. The result of this is that the throughput of the baseline scenario slightly higher than the model based on historical data. This should be kept in mind and analysis of the other scenarios should be done in comparison to the baseline and not the model based on historical data. The benefit of using the baseline is that the structure of the orders and the mean lengths ordered can be altered to see how it affects the factory's performance.

8.2.2 Scenario 1 - Full FIFO

Running production orders straight through factories without any specific sequencing has been a topic for discussion. Running orders first in first out, could be a way to reduce lead times for orders, but can have a negative effect on the throughput. Therefore the change from the baseline to this scenario is to simply shut of all sequencing logic and let the first order coming in to the any of the storages be the first one to be run.

8.2.3 Scenario 2 - Full FIFO without buffers

An even more extreme version of the full FIFO concept is to remove the buffers in between the different machines. This should lower lead times and inventory holding costs. However, the instability of the process increases because a breakdown in one machine will force the entire factory to stop. Just like the other full FIFO scenario throughput can be expected to be affected negatively.

8.2.4 Scenario 3 - No transport times

The factory modeled is the result of incremental changes over a long period of time. The buildings housing the factory and its storages were built with other purposes in mind. Therefore the layout and placements of the different parts of the factory is not in the most logical order, and transport distances are a lot longer than they would be in a factory where the flow was more in line.

To investigate how big an effect the physical layout has on the performance of the factory, a scenario where transports takes no time at all was run. This means that the flow items simply “teleport” to their destination.

8.2.5 Scenario 4 - Varying complexity in production orders

The factory is running a complex mix of orders. While most factories have a product mix of 3-5 different quality-sizes the modeled factory has more than 100. Running different quality-sizes means that the number of setups goes up. An interesting question to ask is how much this affects the performance of the factory.

To be able to see how much the product mix affects performance a series of tests were run varying the lengths of the orders from 10% up to 200%. The amount ordered per day still stays the same, but each order is either shorter or longer.

Two extremes were added to provide values on how “good” and “bad” performance could possibly be. The “bad” extreme is when each order is exactly one roll. This means that between each roll of paper a setup occurs, slowing down the process. The “good” extreme is simply removing all setup times in each step of the factory. The total length of production orders was adjusted to make sure that the factory never was starved.

8.3 Number of runs

Each scenario was run 10 times and scenario 4 was run 10 times for each complexity setting. 10 runs per scenario might seem like a low number, but taking into consideration that out of the 5 scenarios one contained 22 different settings, 10 runs seems more reasonable. Running a scenario 10 times takes approximately 30 minutes.

8.4 Performance measures

The performance measures in the scenarios are the lead times between the start of each machine step and the throughput per month.

9 Output data analysis

9.1 Baseline

Running the baseline scenario yielded the throughput shown in Table 9-1.

Table 9-1 Throughput result for baseline scenario

	Average	95% Conf. Max	95% Conf. Min
Baseline	17.32	17.48	17.18

The throughput in the baseline model is higher than the validated model, and corresponds better to the historical data. The higher throughput can be explained by the way the logic of the model works when creating production orders. The model is required to create production orders each day that together reaches a certain minimum total length, which is set to the average of the historical data. This requirement results in the model creating orders whose combined total length always is longer than the actual average of the historical data.

Comparing the lead times to the validated model (based on actual production orders) shows great similarity. The lead time results for the baseline in comparison with the validated model (based on historical orders); can be seen in Figure 9-1, 9-2, and 9-3.

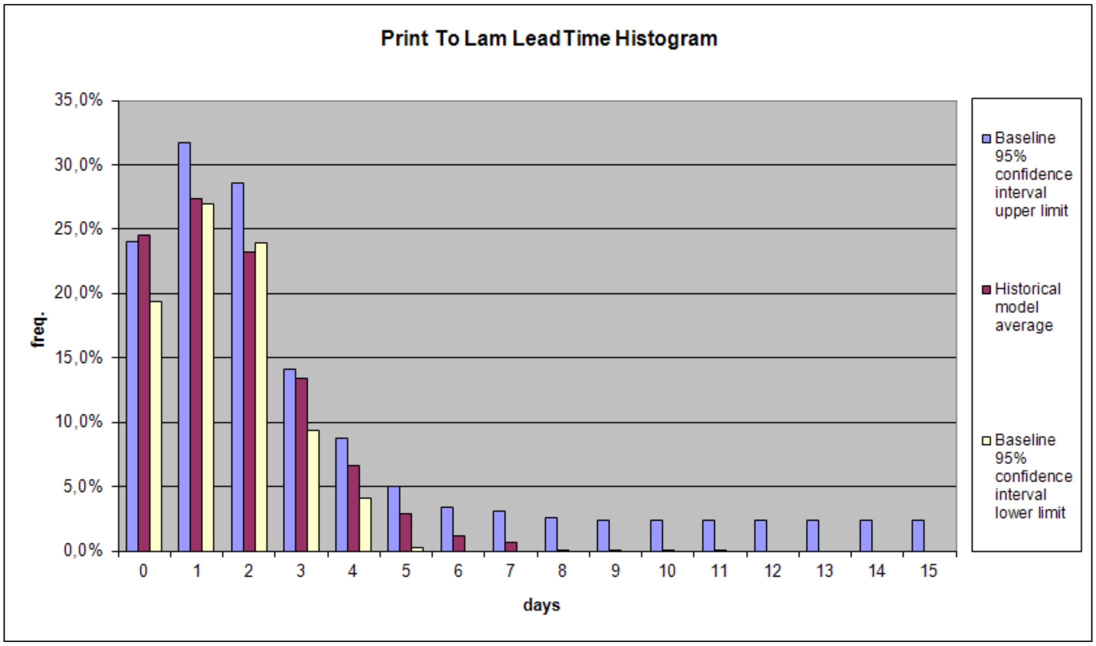


Figure 9-1 Baseline vs. model with historical orders - printer to laminator lead time histogram

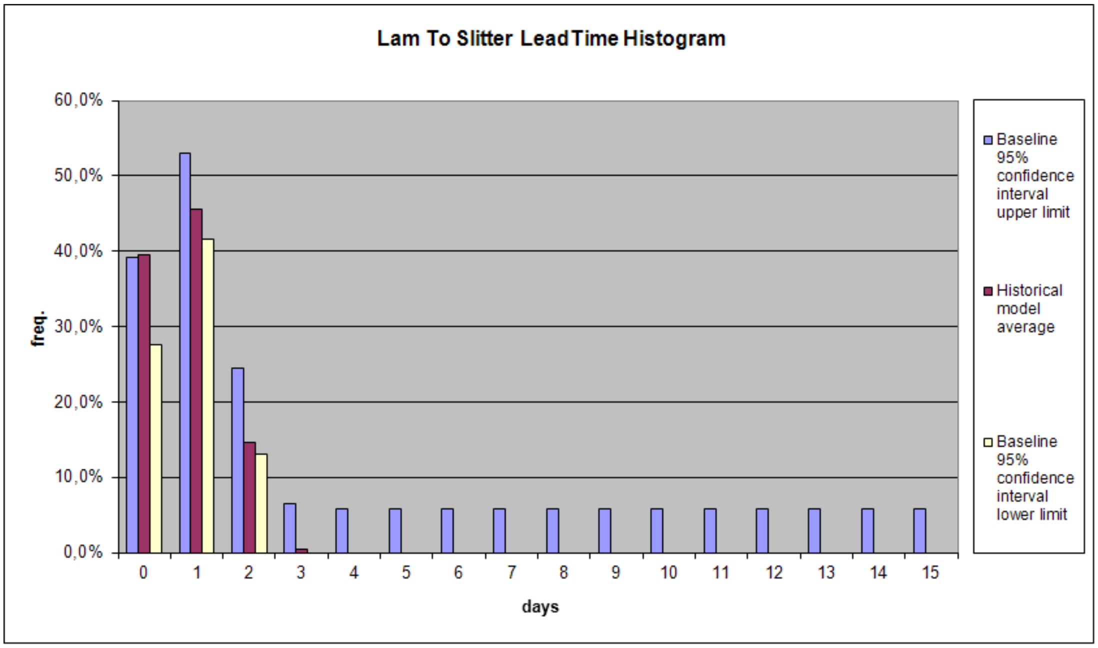


Figure 9-2 Baseline vs. model with historical orders - laminator to slitter lead time histogram

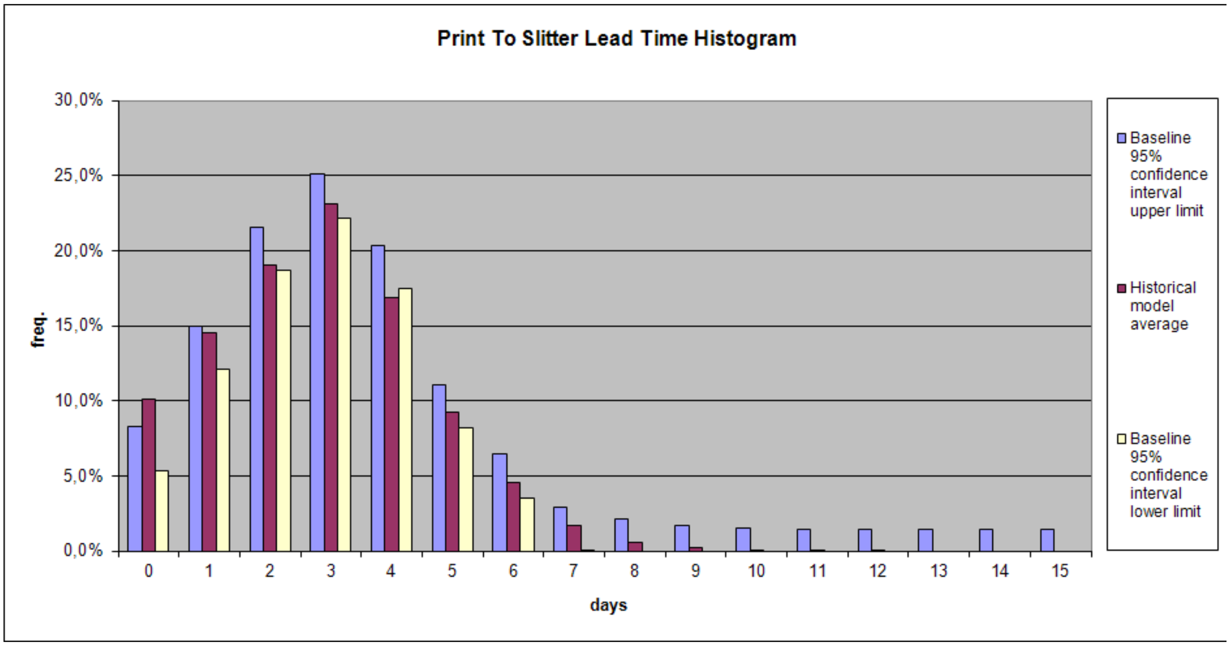


Figure 9-3 Baseline vs. model with historical orders - printer to slitter lead time histogram

This is the scenario to which the other scenarios will be compared.

9.2 Scenario 1 Full FIFO

The throughput result for scenario 1 is shown in Table 9-2.

Table 9-2 Throughput result for scenario 1 and baseline

	Average	95% Conf. Max	95% Conf. Min
Scenario 1	16.61	16.98	16.25
Baseline	17.32	17.48	17.18

In scenario 1 the sequencing logic is removed which results in a lower throughput. The confidence intervals do not overlap which indicates that the difference is statistically significant and this is supported by using the Welch Confidence Interval Approach (WCIA).

In Figure 9-4, 9-5, and 9-6 the lead times between the different machine steps are compared

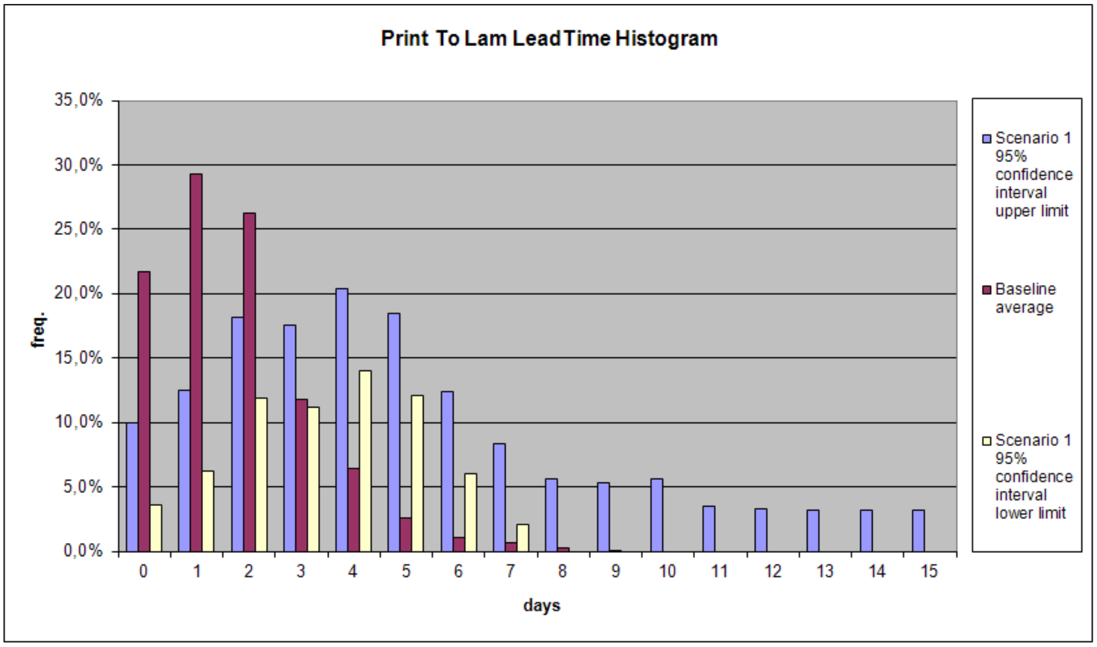


Figure 9-4 Scenario 1 vs. baseline - printer to laminator lead time histogram

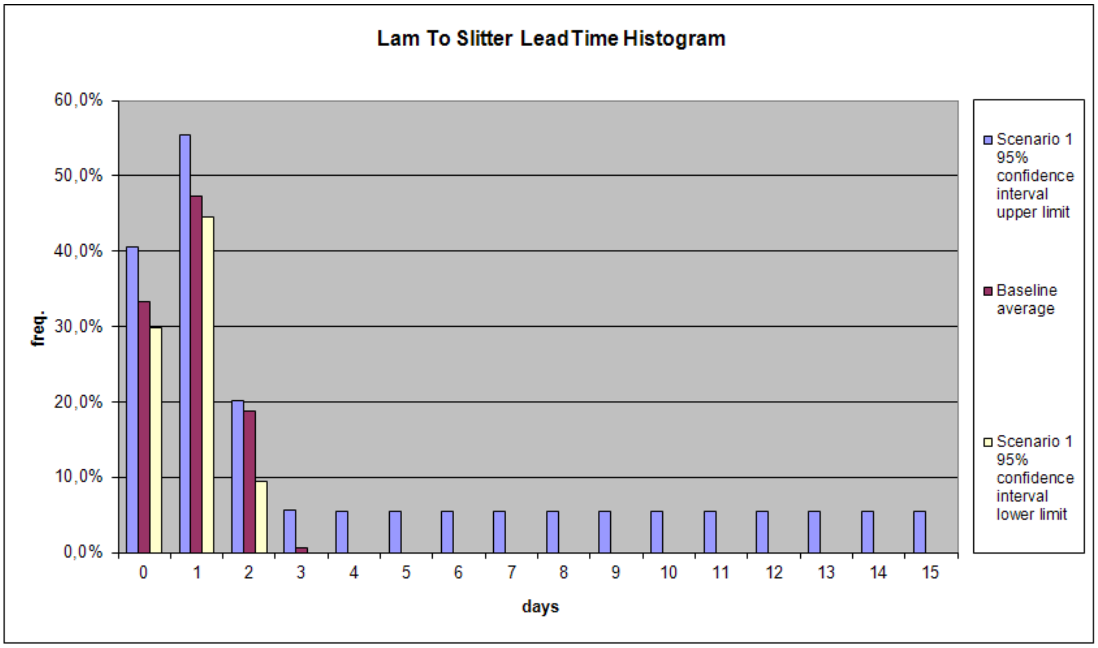


Figure 9-5 Scenario 1 vs. baseline - laminator to slitter lead time histogram

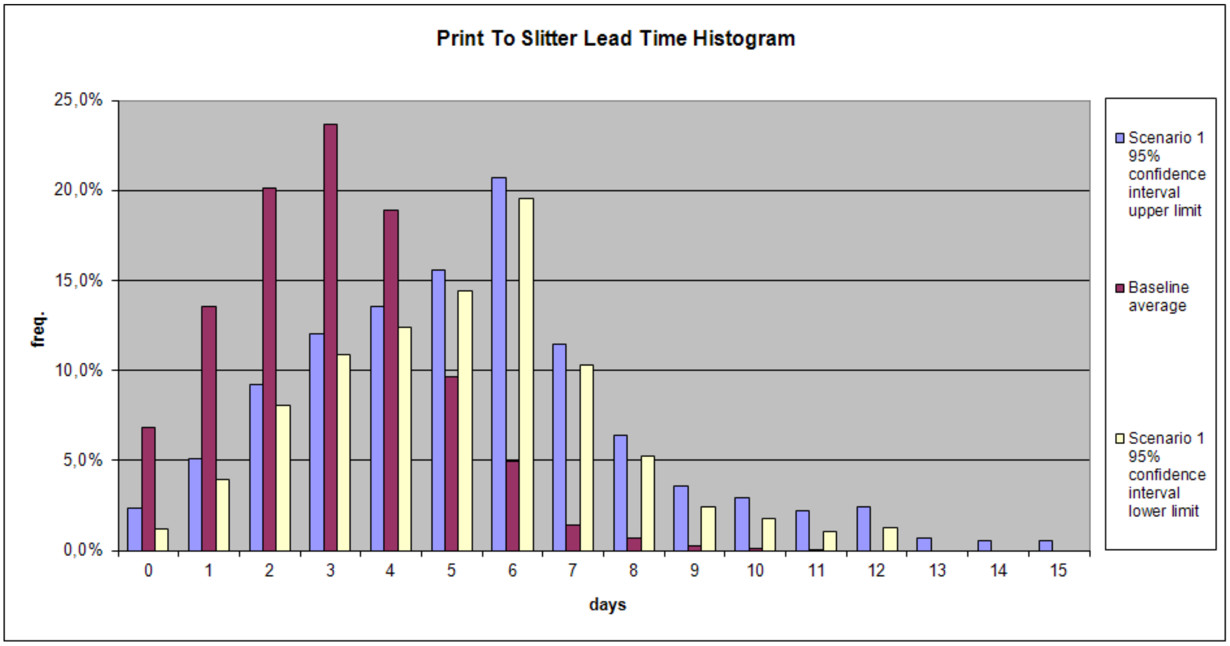


Figure 9-6 Scenario 1 vs. baseline - printer to slitter lead time histogram

It is clear that the lead times are considerably longer for scenario 1 in the first step of the process, printer to laminator. The second step, laminator to slitter, is fairly similar to the baseline, but when put together, printer to slitter, the increase in lead time in the first step is shown to be significant for the overall process.

9.3 Scenario 2 Full FIFO without buffers

The throughput result for scenario 2 is shown in Table 9-3.

Table 9-3 Throughput result for scenario 2 and baseline

	Average	95% Conf. Max	95% Conf. Min
Scenario 2	8.73	8.82	8.64
Baseline	17.32	17.48	17.18

In scenario 2 the sequencing logic is removed and the buffers between the machine groups are removed. This results in a steep decrease of the throughput but a shortening of the lead times to less than one day for all orders. This is because the printers become the bottleneck in this scenario and create a starvation for the other machines which produce everything exiting the printer in less than a day. The Welch Confidence Interval Approach shows that the difference is statistically significant.

Histograms for the lead times in scenario 2 are available in Appendix 2..

9.4 Scenario 3 No transport times

The throughput result for scenario 3 is shown in Table 9-4.

Table 9-4 Throughput result for scenario 3 and baseline

	Average	95% Conf. Max	95% Conf. Min
Scenario 3	17.16	17.27	17.04
Baseline	17.32	17.48	17.18

In scenario 3 all transport times in the factory are set to zero. The average throughput is slightly lower than the baseline but the confidence intervals overlap and WCIA includes the zero so the difference cannot be said to be statistically significant.

In Figure 9-7, 9-8, and 9-8 the lead times from scenario 3 are shown. As can be seen in the histograms, the lead times are almost identical.

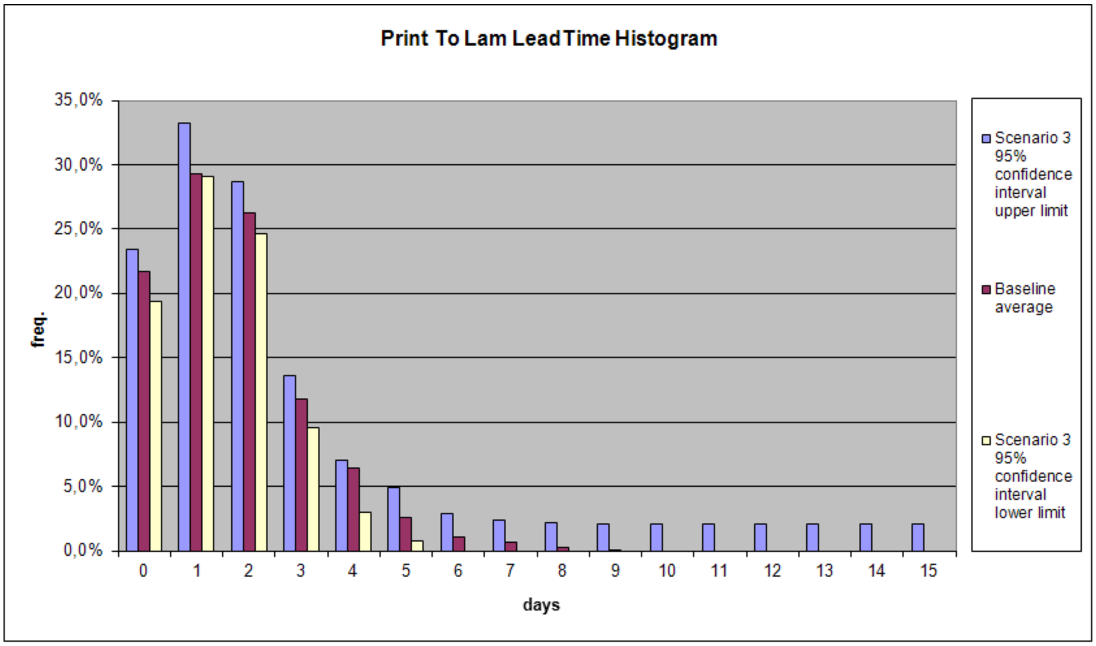


Figure 9-7 Scenario 3 vs. baseline - printer to laminator lead time histogram

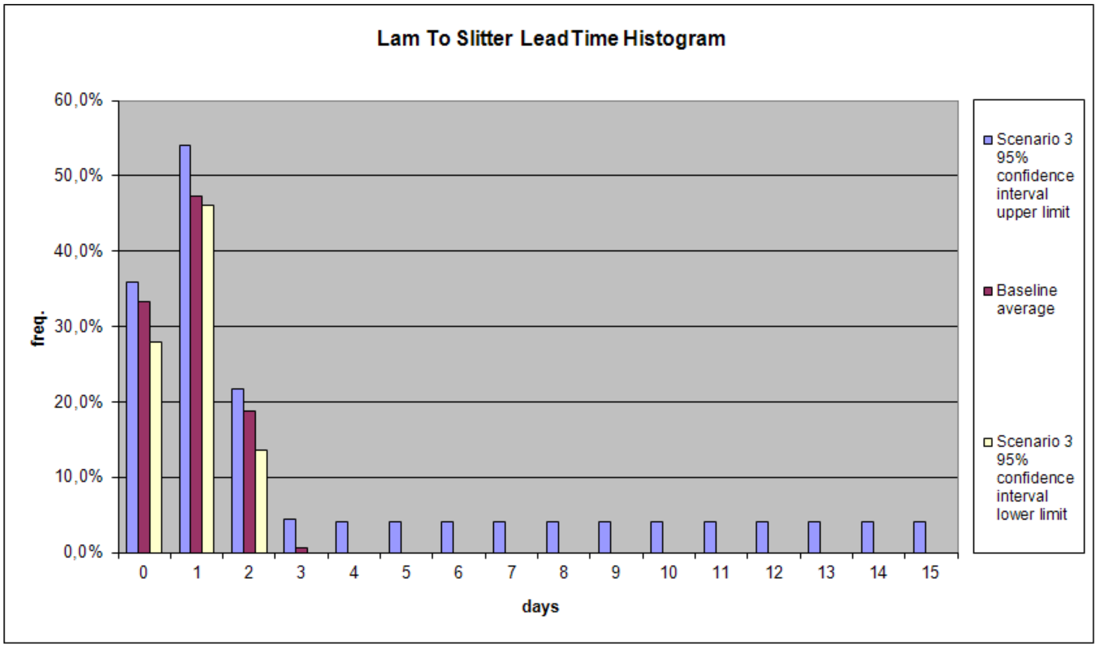


Figure 9-8 Scenario 3 vs. baseline - Laminator to slitter lead time histogram

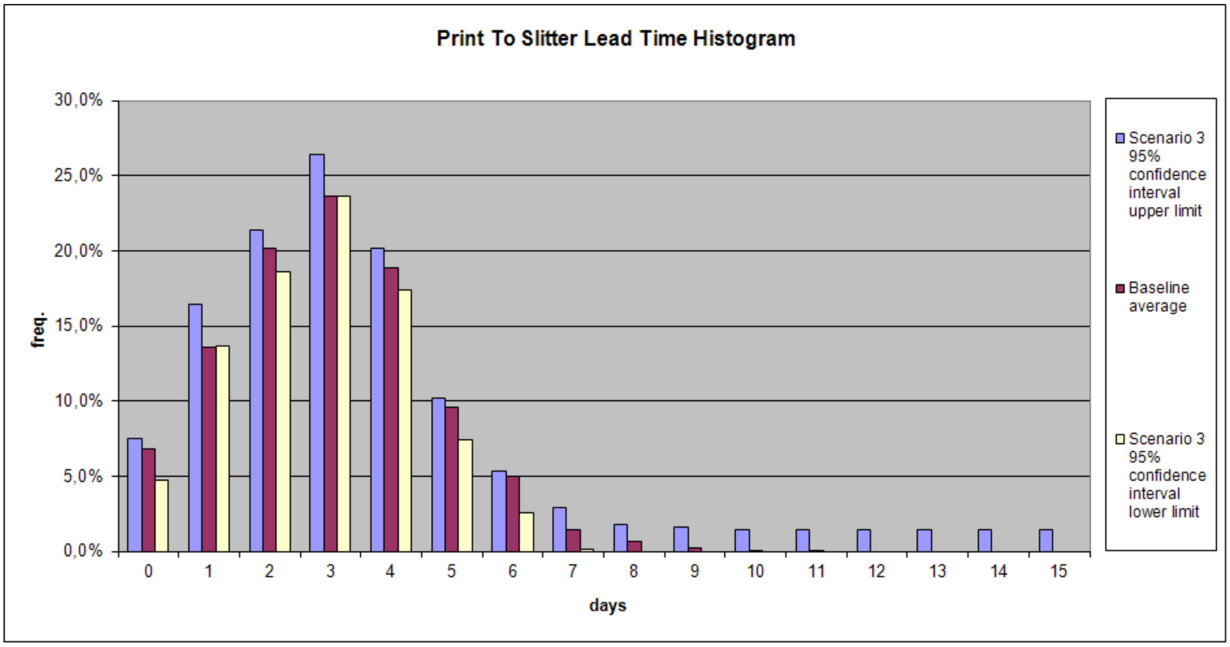


Figure 9-9 Scenario 3 vs. baseline - printer to slitter lead time histogram

9.5 Scenario 4 Varying complexity in production orders

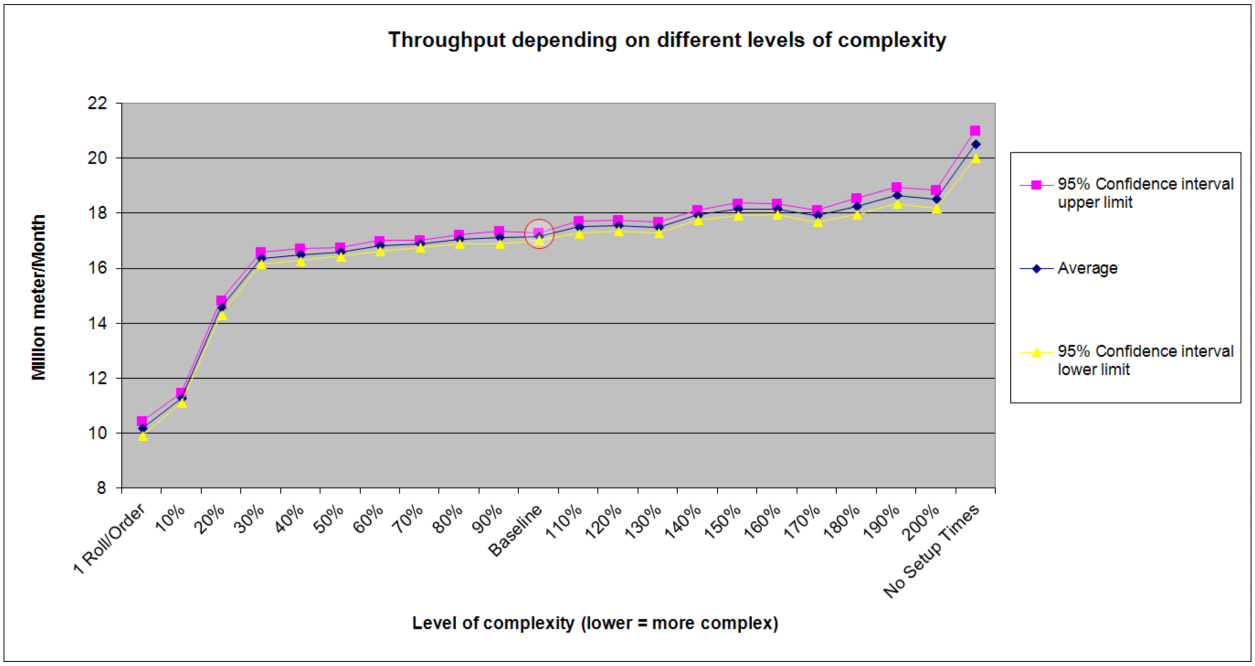


Figure 9-10 Scenario 4 - throughput depending on different levels of complexity

Figure 9-10 shows that there is a slowly rising trend in throughput when increasing the length of the orders. Looking at the cases from 30% to 200% there is a slowly growing mean for the throughput when the order size is increased. The differences between these cases are not always statistically significant. Between close cases like from 40% to 50% the confidence intervals overlap, something that can also be seen in the graph. This overlap is present for all closest neighbor comparisons from 30% to 200% except for 120% to 130%. When looking at cases that are more far apart the difference is statistically significant as the confidence interval do not overlap. This is also supported by an ANOVA test that showed that at least one of the cases is statistically different from the rest.

When the order size falls below 30% the throughput decreases drastically and the 10% to 20% scenarios seem to follow a different curve than the rest of the cases. The differences are statistically significant.

The results from the two extreme scenarios, “1 roll/order” and “no setup times”, are statistically significant from the other cases and serve as a frame of reference to show the possible minimum and maximum values when altering the order lengths.

10 Results and conclusions

10.1 Scenario 1

Not having sequencing in a factory of this type seems to only bring negative effects on performance. The production capacity will be reduced a bit, but most of all the production lead time will increase quite a lot. The fact that lead time becomes higher than when sequencing orders means that this change results in the opposite of what was wanted from it.

10.2 Scenario 2

Not having sequencing and removing the buffers reduces the throughput severely. The absence of buffers makes the system very vulnerable to any disturbance since a stop in any machine quickly affects the whole system. The system as it stands today cannot function efficiently without disconnecting the machines with buffers.

10.3 Scenario 3

The factory’s flow is far from stream lined, but the output from scenario 3 shows that the travelling times for the AGVs have very little effect on the performance of the factory. The resulting throughput from the experiment was slightly lower than for the baseline scenario, which seems illogical. This is simply because of random variations, and the conclusion is that the resulting differences are not statistically significant. There are of course other factors regarding goods transportation that could be looked into such as system reliability, cost of maintaining transportation system and damages on goods as it’s transported.

10.4 Scenario 4

The results of scenario 4 show that the throughput of the system is very stable when it comes to changing the order structure. The orders can be shortened without too much losses of throughput, the real performance dip doesn’t occur until the orders have been shortened to less than 30%. Doubling the order lengths gives a slight but still significant increase of the throughput by 1.4 million meters. Shortening the order length to 30% decreases the throughput by 0.8 million meters which might be an acceptable solution if the factory needs to run shorter orders.

11 Concluding remarks

The experiments produced a number of interesting results which can be of use to the packaging company. The output shows the effect of changes and which factors have the greatest impact on the performance. This knowledge can be useful for future design decisions as well as for the work with continuous improvements in the factory.

The model is easily expandable and work has already begun to include doctoring and other functionality. In addition to new functionality, measures such as securing better data for slitter setups can increase the validity of the model.

All in all, the project is deemed successful and its purpose has been fulfilled.

12 References

Banks, J. Carsson, J. Nelsson, B. and Nicol, D. (2001). *Discrete-event system simulation*. 3rd ed. Upper Saddle River: Prentice Hall

Banks, J. (2000). *Getting started with automod*. [e-book] Bountiful: Autosimulations, Inc. Available at: http://www7.informatik.uni-erlangen.de/~heindl/teaching/ws04/sm1/f/exnotes/AutoMod11/gswam.pdf [Accessed 2010-11-04]

Chung, A. (2004). *Simulation Modeling Handbook: A Practical Handbook*, Boca Raton: CRC press

Cockburn, Alistair; Williams, Laurie (2000). *The Costs and Benefits of Pair Programming,* [PDF]. Proceedings of the First International Conference on Extreme Programming and Flexible Processes in Software Engineering (XP2000). Available at: http://collaboration.csc.ncsu.edu/laurie/Papers/XPSardinia.PDF. [Accessed 2010-11-04]

Ferrada & Omeragic, (2010). *Optimizing a production line of packaging material by the use of Discrete-event simulation*. Theseis, MSc. Faculty of Engineering LTH at Lund University.

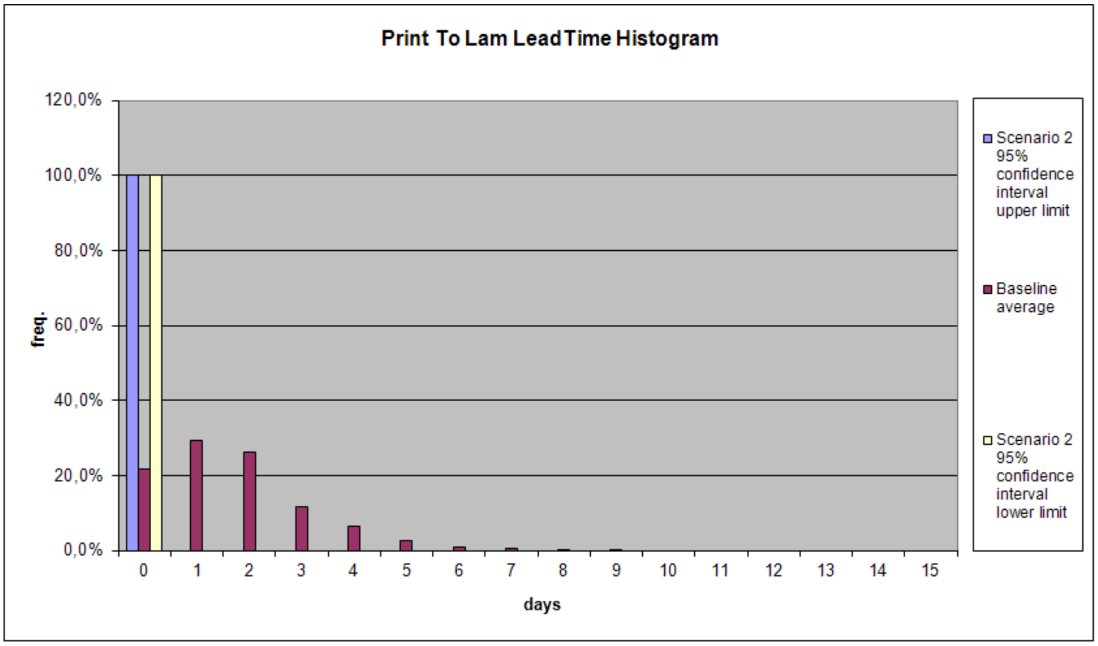
Appendix 1 – Frequency and length distributions for orders of different quality-sizes

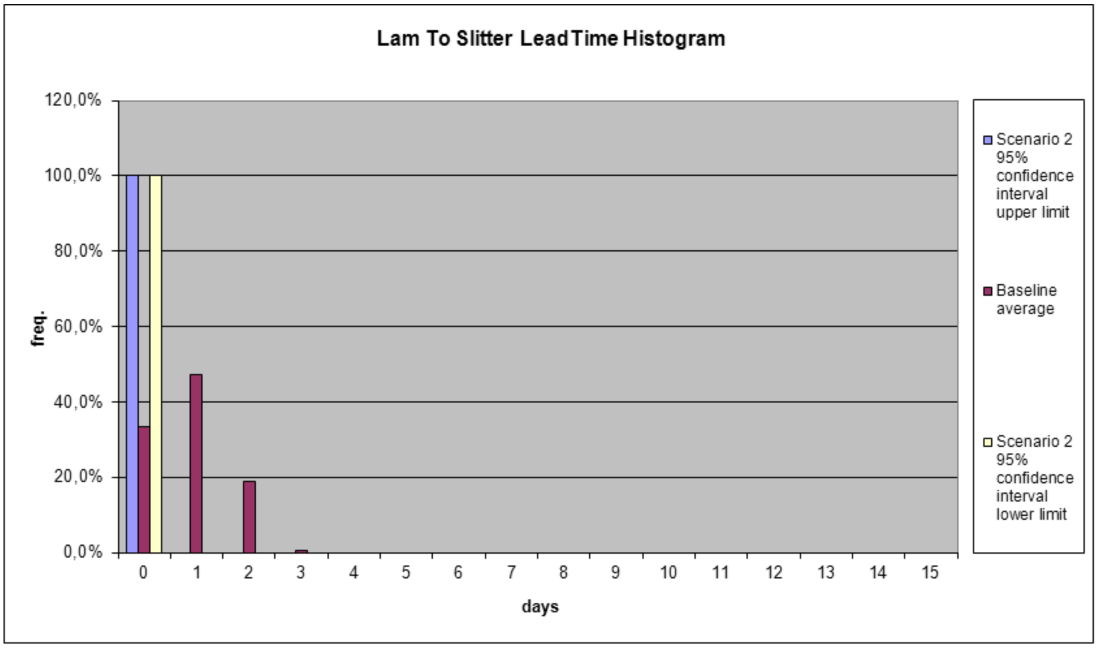
Quality-size	Frequency	Length distribution
4541-566	1	triangular(47753,47755,47754)
4541-809	1	triangular(28171,28173,28172)
4541-810	4	triangular(13443,21309,12764,5)
4542-463	2	triangular(8837,95989,52413)
4542-466	1	triangular(6360,6362,6361)
4542-566	20	triangular(7032,102312,17624)
4542-567	7	triangular(12171,242823,21599)
4542-609	16	triangular(12495,134023,21002,5)
4542-661	3	triangular(11518,112096,42467)
4542-700	23	triangular(13579,135707,29032)
4542-702	22	triangular(9317,167697,20065)
4542-760	1	triangular(131125,131127,131126)
4542-809	13	triangular(41072,131813,51797)
4542-810	50	triangular(23444,144513,27113)
4543-566	12	triangular(1725,134502,60004,5)
4546-760	3	triangular(21239,41661,21239)
4546-809	53	triangular(17255,231913,40992)
4549-236	2	triangular(13852,16329,15090,5)
4549-566	9	triangular(42082,91162,22107)
4549-567	4	triangular(13622,13909,11230,5)
4549-609	3	triangular(6877,301609,41044)
4549-700	4	triangular(73227,86772,47896,5)
4549-702	10	triangular(18846,79579,19834)
4549-760	3	triangular(42414,58227,42414)
4549-809	4	triangular(1324,20421,13758)
4560-567	1	triangular(52136,52138,52137)
4565-809	1	triangular(13356,13358,13357)
6065-460	3	triangular(6947,78517,58771)
6065-560	2	triangular(21376,42844,32110)
6065-580	1	triangular(18134,18136,18135)
6065-701	4	triangular(20995,49869,32831,5)
6065-836	2	triangular(11316,78854,45085)
6066-810	19	triangular(19959,149523,36219)
6131-560	2	triangular(14024,124729,69376,5)
6131-580	12	triangular(28625,90209,27933)
6131-701	1	triangular(68542,68544,68543)

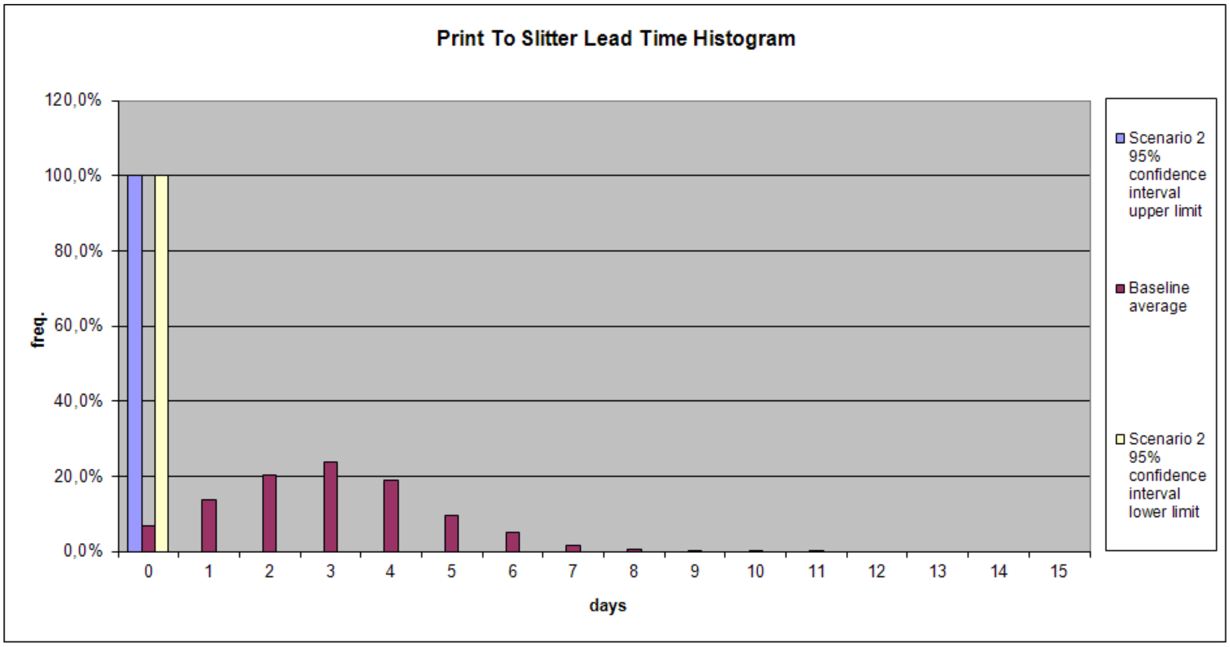
6486-560	1	triangular(14185,14187,14186)
6539-460	1	triangular(21266,21268,21267)
6539-465	3	triangular(13728,222049,148112)
6539-560	10	triangular(17143,145715,17260)
6539-630	1	triangular(14351,14353,14352)
6539-813	3	triangular(11726,43429,42737)
6552-813	1	triangular(14066,14068,14067)
6935-460	1	triangular(20134,20136,20135)
6935-630	9	triangular(15863,67966,13897)
6971-460	2	triangular(14366,39287,14366)
6971-565	2	triangular(52766,87318,70042)
6971-705	1	triangular(1326,1328,1327)
6973-465	6	triangular(29176,100731,58747)
6973-560	1	triangular(117807,117809,117808)
6973-810	1	triangular(75067,75069,75068)
6973-813	5	triangular(97057,100575,42093)
7024-813	2	triangular(4588,46996,25792)
7037-465	1	triangular(21465,21467,21466)
7042-810	42	triangular(27915,213592,33281,5)
7042-812	2	triangular(14515,144814,79664,5)
7045-460	3	triangular(42163,42313,42163)
7045-465	23	triangular(13628,124847,40139)
7045-705	3	triangular(14719,67774,14719)
7045-813	12	triangular(68907,252145,44528)
7069-565	3	triangular(2593,130819,2593)
7090-701	4	triangular(104885,178272,123656)
7104-701	1	triangular(12631,12633,12632)
7185-705	8	triangular(1665,78339,31300)
7191-465	6	triangular(132704,144798,39292,5)
7197-460	1	triangular(68474,68476,68475)
7197-705	14	triangular(21433,87245,14018)
7198-460	8	triangular(104807,105772,60428)
7369-460	6	triangular(51422,133641,51211,5)
7369-465	4	triangular(51389,62249,36944)
7369-560	15	triangular(48932,144304,28262)
7369-565	15	triangular(84698,91613,14716)
7369-705	11	triangular(28035,130937,72528)
7369-810	4	triangular(21234,28646,21135)
7369-813	18	triangular(69978,131206,43328)
7369-835	14	triangular(2066,112915,15190,5)
7406-465	4	triangular(13862,77986,15569)
7406-565	2	triangular(20658,77388,49023)
7406-813	1	triangular(15138,15140,15139)

7414-460	5	triangular(33199,41027,21446)
7414-465	1	triangular(1381,1383,1382)
7414-560	13	triangular(43666,163335,40334)
7414-565	3	triangular(12673,42484,12673)
7414-630	6	triangular(29369,79774,13991)
7414-705	19	triangular(15955,90691,19292)
7414-810	3	triangular(49096,84468,49096)
7414-813	15	triangular(70204,138404,34722)
7421-810	1	triangular(58151,58153,58152)
7429-630	3	triangular(7106,70579,7106)
7438-701	14	triangular(64665,188816,34393,5)
7445-460	1	triangular(131962,131964,131963)
7445-560	1	triangular(13736,13738,13737)
7445-565	1	triangular(99472,99474,99473)
7445-705	15	triangular(42333,117815,28533)
7445-810	3	triangular(42967,73024,56359)
7445-813	4	triangular(22459,61616,17747,5)
7469-813	5	triangular(13758,67387,21777)
7524-701	12	triangular(11523,149787,27083,5)
9171-701	3	triangular(13539,21137,13539)
9999-999	12	triangular(42674,157059,21670)

Appendix 2 – Lead time histograms for scenario 2






Print To Slitter Lead Time Histogram
freq.
120,0%
100,0%
80,0%
60,0%
40,0%
20,0%
0,0%
0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
days
Scenario 2 95% confidence interval upper limit
Baseline average
Scenario 2 95% confidence interval lower limit