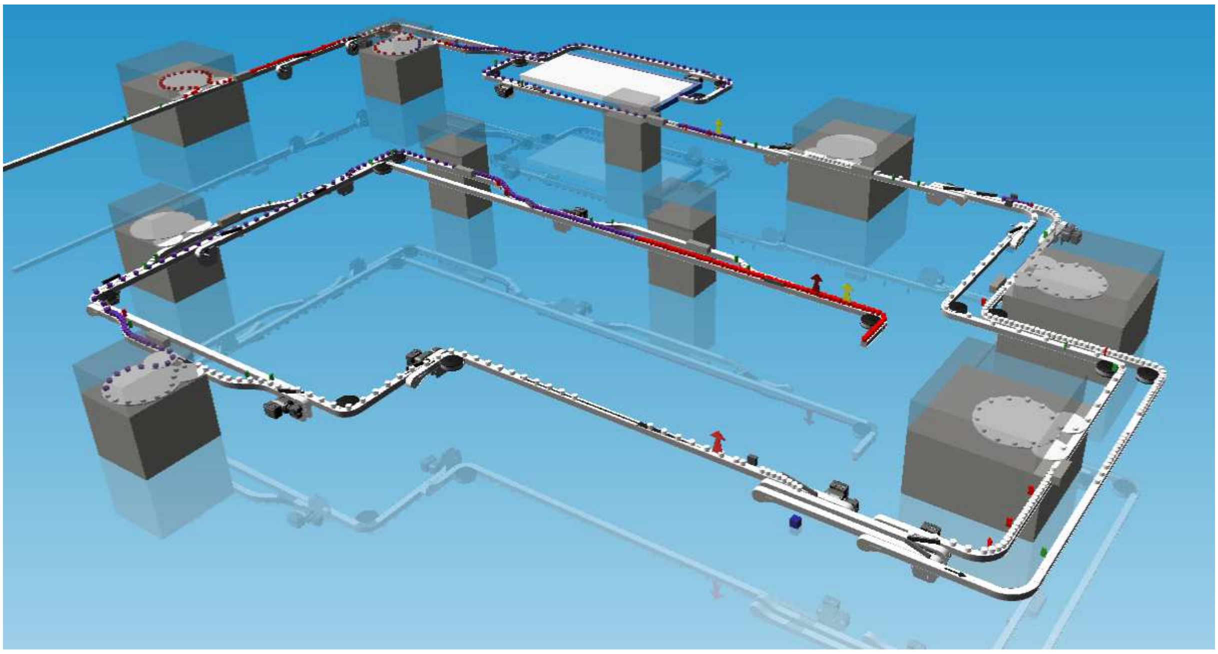

Using Modular Discrete Event Simulation for Modelling Fast Moving Consumer Goods Lines

Master's Thesis in the Master's Programme Production Engineering

ROBERT COLLIN-KARLSSON & FREDRIK RAHM

Department of Product and Production Development
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2015

Using Modular Discrete Event Simulation for Modelling Fast Moving Consumer Goods Lines

Master’s *Thesis in the Master’s Programme Production Engineering*

ROBERT COLLIN-KARLSSON & FREDRIK RAHM

Department of Product and Production development
CHALMERS UNIVERSITY OF TECHNOLOGY
Göteborg, Sweden 2015

Using Modular Discrete Event Simulation for Modelling Fast Moving Consumer Goods Lines

Master's Thesis in the Master's Programme Production Engineering
ROBERT COLLIN-KARLSSON & FREDRIK RAHM

Examensarbete / Institutionen för produkt- och produktionsutveckling,
Chalmers tekniska högskola

Department of Product and Production development
Chalmers University of Technology
SE-412 96 Göteborg
Sweden
Telephone: + 46 (0)31-772 1000

Cover:
Screenshot of the simulation model.

Reproservice, Chalmers
Göteborg, Sweden 2015

Using Modular Discrete Event Simulation for Modelling Fast Moving Consumer Goods Lines

Master's Thesis in the Master's Programme Production Engineering
ROBERT COLLIN-KARLSSON & FREDRIK RAHM
Department of Product and Production development
Chalmers University of Technology

ABSTRACT

Today high efficiency, low environmental footprint and good personal working conditions of production lines are important objectives for factories in the world. This master thesis work aims to create a simulation model of a food processing line based on qualitative and quantitative data. Using modular discrete event simulation as main approach, a model able to quantify effects on productivity and noise level from production line alterations is produced. To aid the modular approach, a framework of how to represent common machines in a fast moving consumer goods line is formed. The simulation model is used to analyze suggestions of alterations to the flow of products. By reducing the supply of products coming in to the system, and reducing the variance in the supply to the bottleneck machines, the collisions could be reduced by 65% and the productivity could be raised by 0.6%.

KEYWORDS:

Discrete event simulation, DES, Modular Discrete event simulation, Fast Moving Consumer Goods, FMCG

Contents

Preface

There are a number of persons that by their support and guidance have been fundamental in the work leading to this master thesis. Thanks all of you! A special thanks is directed to our supervisor at FlexLink, Ken Uozumi for arranging this project and discussions driving us forward.

Secondly we would like to thank Ingela and the rest of the employees at the food producing company for always welcoming us to the factory and for good cooperation.

Thirdly our gratitude goes to our supervisor at Chalmers, Malin Karlsson and examiner Björn Johansson for helpful guidance and good ideas.

Finally we would like to wish a good luck to the food producing company in their work with improving their production line and to FlexLink on their future simulation journey.

Göteborg, June 2014

Robert Collin-Karlsson

Fredrik Rahm

1 Introduction

Today high efficiency, low environmental footprint and good personal working conditions of production lines are important objectives for factories in the world. The designs of those lines are mainly based on engineers' experience together with rough cut calculations and in many cases without complete production line perspective. An optimal design of those production lines is a key factor to achieve the aforementioned objectives.

A way to ensure the performance of a conceptual line and to reduce the risk of unexpected layout mistakes is to bring the use of simulation tools into the design process. Simulation has been found to be both useful and powerful as a tool for designing and analyzing production systems. (Law 2007) In industry simulations are used as one-offs for example to design a new factory, as well as for improvements where models can be reused. Few companies use it as a mandatory tool. (Skoogh, 2011)

1.1 Background

FlexLink is the world leader of high end production logistics solutions for the Healthcare, Consumer Goods (FMCG), Electronics and Automotive industries. Headquartered in Gothenburg, Sweden, FlexLink has more than 800 employees and operates in 26 countries across the globe. The company is part of Coesia, an industrial group including companies in advanced automated machinery, industrial process solutions and precision gears sectors of industry.

FlexLink are seeking a way to integrate the use of simulation tools in their engineers' daily work of designing production lines. In this, the R&D department of FlexLink is running several projects for the enhancement of new simulation tools for production line efficiency optimization in cooperation with Chalmers University of Technology, together with in-house efforts to aid the use of simulation tools.

Research has shown that a modular approach to Discrete Event Simulation (DES) can divide the needs for expert knowledge in simulation and in manufacturing systems. This allows for a simulation expert to produce validated modules that can be used by a manufacturing systems expert to model a production line. These modular simulation blocks can be re-used, which can also decrease the lead times for model creation (Johansson, 2006). Examples of simulation blocks are various machines and conveyors.

With this approach in mind FlexLink have chosen a software and started to develop an add-on to fit their needs. As for now this add-on is in an early stage of development and the use and knowledge of the software as well as for simulation projects in general is limited among the engineers at the company. As a step to spread the knowledge about simulation projects in the organization and at the same time test and demonstrate the new software this master thesis is conducted.

The project is conducted together with a food producing company and is applied on their production system that produces food filled in glass jars.

1.2 Purpose

This master thesis will focus on the modelling of a production line at a food producing company using a discrete event simulation software, in order to improve the line efficiency and assess a problem with colliding products causing a high noise level and increasing the risk of food hazard due to broken glass jars. While performing this simulation, modular blocks will be created that can support application engineers around the globe in their layout design work and communication with colleagues of other professions. Furthermore, a framework for representing a number of different types of machines will be created and analyzed in order to create flexible and reusable blocks.

The modelling of the production line will include analysis and modelling in the software 3DCreate, together with FlexLink Design Tool, which is the add-on to 3DCreate developed by FlexLink. As the add-on is under development, some testing of the applicability of the software to similar engineering problems is included in the work.

The simulation model will cover the entire production line, from the supply of raw material to the palletizing of finished goods. This operation will need collection of data from the factory.

1.3 Goal

- Create reusable blocks that can be used for communication between different professions.
- Build a simulation model of the studied production line, based on collected data that can predict productivity and number of collisions.
- Do experiments to the model to try find a solution that minimizes the collisions at the line while maintaining at least the same productivity

1.4 Delimitations

This project will not create blocks for representation of robots or conveyors, as such are already present in libraries available for the project.

2 Theoretical background

In this chapter all relevant theory for the project is presented regarding manufacturing simulation, the studied production system, and industry specific frames.

2.1 Manufacturing simulation

Discrete event simulation (DES) is a simulation procedure where the system state variables change value instantaneously at certain points in time. The points in time when the state variables change value are decided by when events occur in the system. In this way DES has the ability to capture dynamic behaviors which makes it suitable to use for simulation of the dynamics of a production system. (Law 2007, Fishman)

A couple of different classic DES methodologies have been developed. In common for the different classical methodologies is that a simulation project carried out according to these methodologies requires the work of a simulation expert and it is likely that each new model has to be created from scratch (Johansson, 2006).

2.1.1 Modular discrete event simulation

Modular DES is a methodology aimed to reduce the amount of time needed for a simulation project and also to enable for non-simulation experts to use the tool of simulation. This is reached by modifying the classical DES methodology to cater for reusability of modules from previous simulation projects.

One of the most well-known classical DES methodologies is Banks' methodology which can be seen in Figure 1.

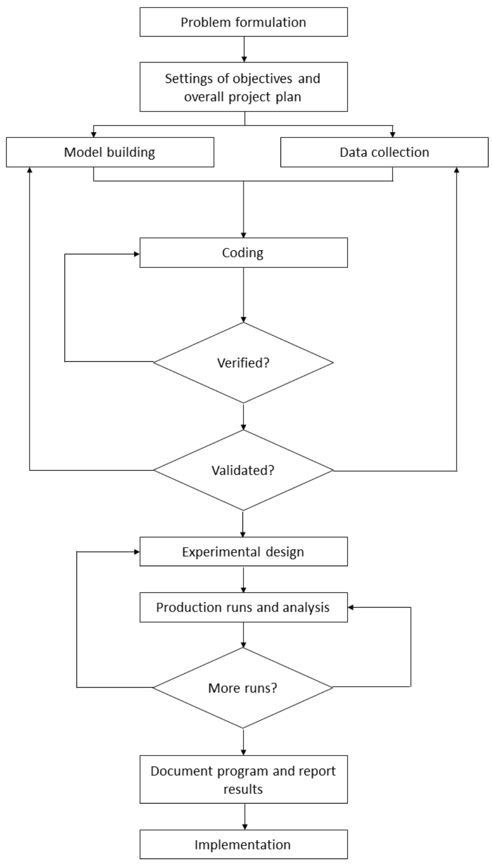

Figure 1. Steps in a simulation study proposed by Banks based on Johansson (2006).

In the modular DES methodology proposed by Johansson (2006), the required coding of logics and functionalities in the model has been taken out of the actual building loop of the simulation model and placed in a separate loop for building modules. The proposed methodology for module building is presented in Figure 2. Figure 3 show the steps in model building.

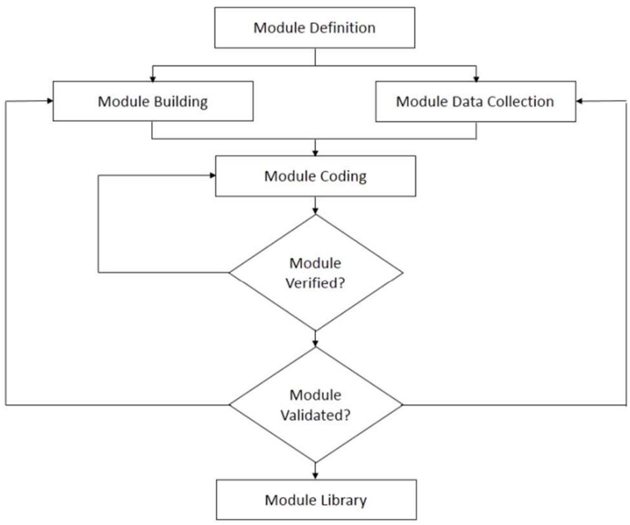

Figure 3. Method for module building based on Johansson (2006).

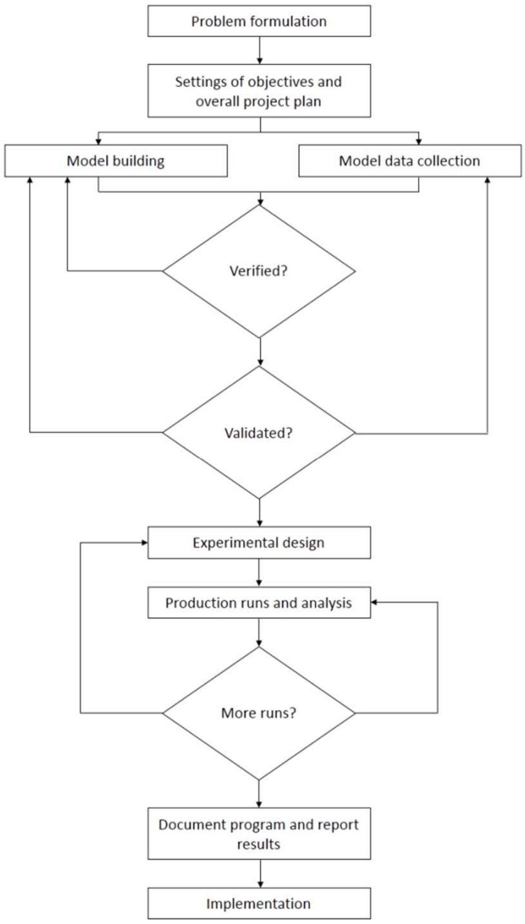

Figure 2. Method for model building based on Johansson (2006).

The idea is that the module building is taken care of by simulation experts that builds general modules which easily can be modified through parameters to fit the properties of different production systems. These modules are then placed in a digital library where others can access them to build simulation models. In this way the time to build a model is reduced as the logics are already implemented in the modules. This makes it possible for someone to build a model even though not being familiar with the internal parts or the code of the modules. The modular DES methodology is mainly aimed for companies building similar manufacturing systems over and over again, containing different combinations of modular components. Examples of such companies are line builders, machines builders and system integrators (Johansson, 2006).

2.1.2 3DCreate and FlexLink Design Tool

3Dcreate is a simulation software developed by Visual Components. The software is offering highly detailed 3D simulations as well as a drag and drop interface to place components in the simulation environment to build a simulation model. The latter is a prerequisite for modular DES (Johansson, 2006). 3DCreate is part of a series of versions based on the same simulation engine where the different versions gives the user more or less functionality as editing capabilities and analysis tools to work with. 3DCreate enables the user not only to drag and drop components from a library to build a simulation model but also to create new components to expand the library.

When creating new components in 3DCreate the geometry and functionality of the component has to be defined. The geometry can be defined either by importing a CAD model to use in 3DCreate or by creating the geometry directly in 3DCreate using simple design tools. The functionality is defined through adding behaviors and parameters to the component through the graphical user interface. Behaviors give the component the simulation functionalities needed to represent the production system.

The most relevant behaviors in this project are paths which move products inside a component, interfaces to connect components with each other, sensors to detect products and other components, signals to communicate internally and with other components, and python scripts to control the logics of the component. The python script can also be used to manipulate parameters of the other behaviors. Properties is a function which is used for giving the user the possibility to assign new values for the components' parameters.

FlexLink Design Tool (FLDT) is an add-on for 3DCreate which is developed internally by FlexLink. The tool provides a fast and easy way of building and configuring conveyor layouts using FlexLink's own product range and a user-friendly user interface. The FLDT add-on together with the capability of importing layout drawings facilitates the foundation for the simulation model, the conveyors that transport the products.

2.1.3 Input data for manufacturing simulation

Manufacturing simulation is about building a computerized model of a production system with the purpose to imitate the operations of a real world facility. Input data for simulation models can be quantitative or qualitative, where the quantitative data is collectable in form of numbers and qualitative data can be collected in form of for example logical relations or rules. In order to categorize what input data collection activities that are needed, Robinson and Bhatia (1995) presents a classification of raw data in three steps, see Table 1.

Table 1: Classification of data (Robinson and Batia, 1995)

Category	Data availability
Category A	Available
Category B	Not available but collectable
Category C	Not available and not collectable

2.1.3.1 Quantitative data

In order for the model to be an accurate imitation of the real world system it is dependent of quantitative input data that makes good representations of all parameters necessary to include in the model. To generate good representations for those parameters the quantitative input data should consist of as much raw data as possible, especially for those parameters that tends to vary. A rule of thumb is to collect at least 230 samples for those variable parameters. The reason why this great amount of samples is preferred is that they are used to create statistical distributions over the variations. Typical parameters necessary to include in a manufacturing system simulation model are; cycle times, time to failure, time to repair and conveyor speeds. (Law 2007, Skoogh and Johansson 2008)

In order to transform quantitative data to information needed for input to a DES model, five important methods for adding value to the raw data are presented in Table 2.

Table 2: Important methods to add value to raw data (Skoogh et al. 2012)

Method	Purpose
Contextualization	Knowledge about what purpose data was gathered for
Categorization	Knowledge about units of analysis or key components of the data
Correction	Errors are removed from the data
Calculation	Mathematical calculations or statistical analysis of the data
Condensation	The data have been summarized in a more concise term

2.2 The production system

The production system studied in this project is a production line for pickled herring in Sweden. The line is fully automated and consists of several machines connected by a conveyor system. At the beginning of the line, empty glass jars are fed into the system by a robot cell. From there the jars are traveling throughout the conveyor system to get to different machines. The machines are mainly connected in a serial flow. The only place where the flow is divided into parallel flows is where herring is filled into the jars by two similar machines. Even though working in parallel, the machines form the working station with the least capacity in the line. Apart from the herring the jars are also filled with spices, vegetables and sauce and then fitted with labels and lids before the products, consisting of a glass jar filled with the actual product, are washed, scanned, bundled and packed on euro pallets. The production line is capable of producing between 160 and 180 products every minute depending on the product size. Two sizes are produced, big and small. The used routing is decided by the recipe. One recipe is produced at a time, but a great amount of different recipes can be produced. When changing recipe, the line needs to be run empty and then stopped to allow for setup. An overview of the process can be seen in Figure 4.

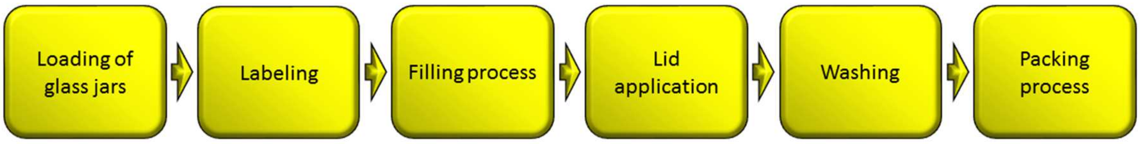

Figure 4. Principal flow of the production system.

The production system is set up so that conveyors and other machines run at a higher speed than the herring fillers are capable of in order to keep the herring fillers busy. Buffering occur on conveyors, as products can close in on each other. This is the definition of accumulation. During normal production this accumulation is present in big queues in front of the herring fillers which gives rise to much noise as the glass jars are constantly colliding against each other. It is also a problem with jars breaking due to the collisions, both in the production line and after the production is finished. A broken jar in the production line causes a big disturbance to the production process, as the line needs to be shut down and thoroughly cleaned from fragments of glass which could otherwise compromise the product and cause food hazard. Broken jars post production causes claims from customers and also, due to the stacking of products on pallets a broken jar compromises the entire palletized batch. There are also two accumulation tables on the line, one downstream from the herring fillers in the filling process, and one between the washing and packing processes.

2.3 Industry-specific frames

To be able to use the simulation model for communication between different professions present in the industry, industry-specific methods for representing machine control logics and measuring productivity will be implemented.

2.3.1 Pack ML

Packaging machine language (PackML) is a standardized machine language for automated machines developed by the OMAC Packaging Workgroup. The standard aims to provide a common "look and feel" together with a common way to structure the control logics of machines across the plant floor in packing industry. Included in this standard is a state model which defines 17 states that together represent the operational sequencing of the automated machinery (OMAC, 2014). Figure 5 illustrates the operation sequencing.

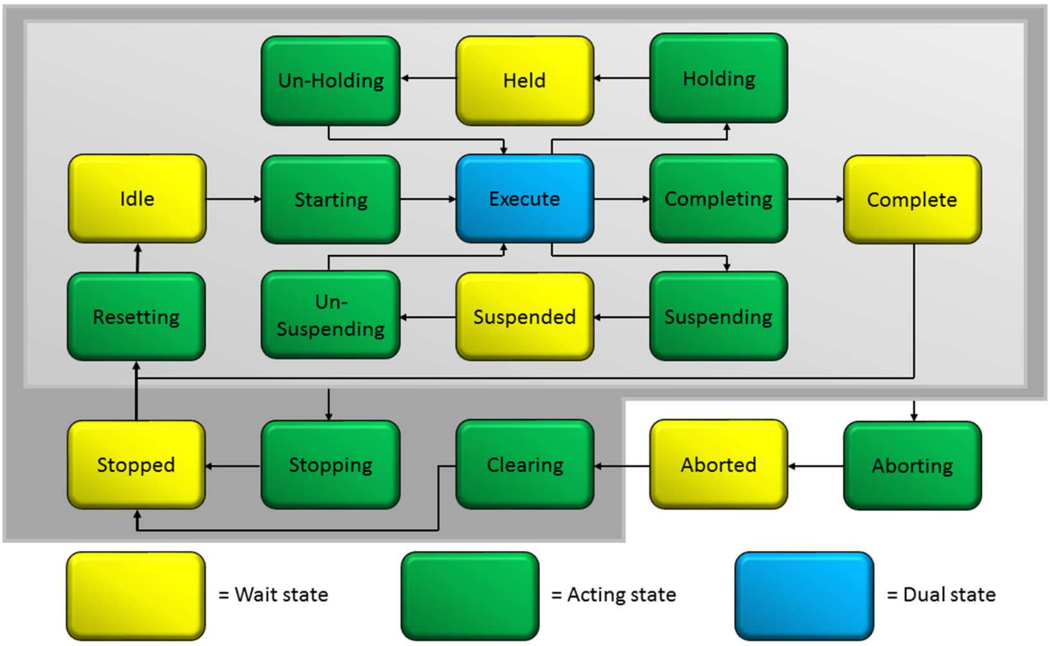

Figure 5. The Pack ML state model based on Automation World (2014). The color coding show the type of activity performed in the state.

When a machine is producing it is situated in the state Execute. If it is starved or blocked it enters the state Suspended, and if it is paused it goes into the Held state. A fault in the machine sends it into the state Aborted. For all other time when the machine is up but not producing it is situated in either of the states Stopped, Idle or Complete. Not all machines requires all 17 states to define their operational sequencing. PackML also include a standardized way of communication between machines in a packing line (Automation World, 2014).

2.3.2 OEE

Overall Equipment Effectiveness (OEE) is described by Badiger *et al.* (2008) as a general measurement aimed to improve plant performance in any kind of manufacturing organization. The measurement focuses on three factors; Availability, Performance efficiency, and Quality rate, which all are crucial to maintain in order to maintain effective operations. Tied to these three factors are six major losses, namely:

1. Equipment failure/breakdown losses.
2. Setup and adjustment losses.
3. Idling and minor stop losses.
4. Reduced speed losses.
5. Reduced yield losses.
6. Quality defects and rework.

The first two losses are known as downtime losses and are used to calculate the availability factor. Loss three and four are called speed losses and determines the performance efficiency. The last two losses, called defect losses, are used to calculate the quality rate. The OEE measurement does not give a specific reason why a machine is not running as efficiently as possible but it helps to detect the possible areas for improvement. The traditional way of calculating OEE is shown in Figure 6.

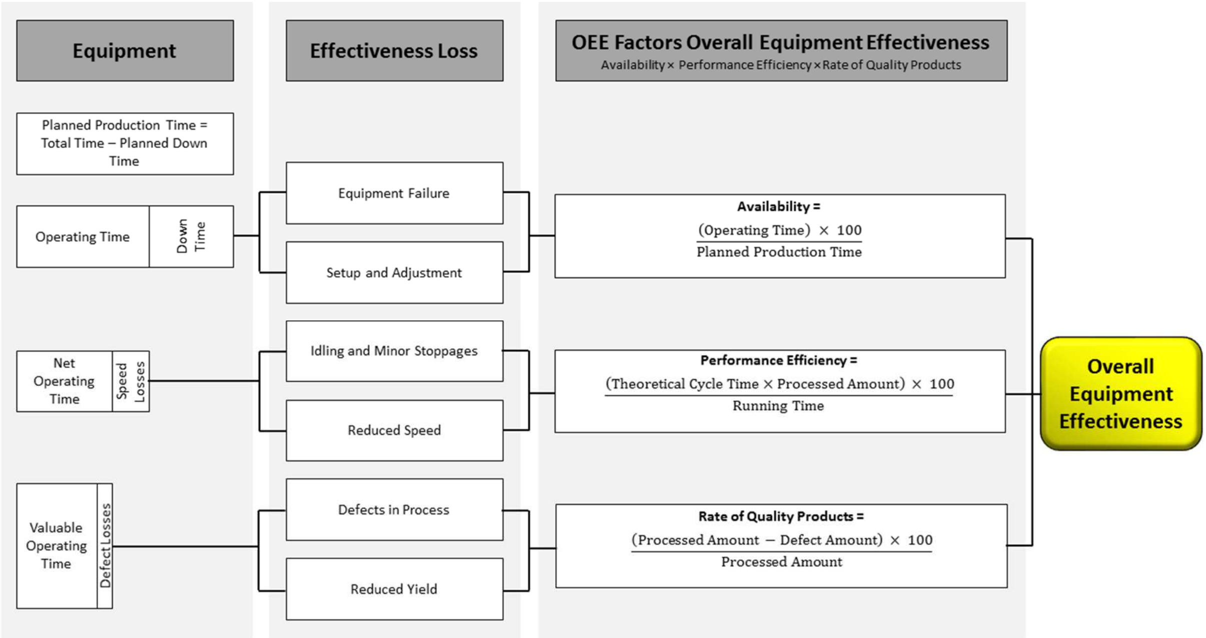

Figure 6. Traditional OEE calculation.

2.3.2.1 Modified OEE

Badiger *et al.* (2008) have proposed a modified way of calculating OEE by adding a fourth factor, Usability, to the OEE measurement. The function of this factor is to divide the unplanned downtime into equipment-related downtime and process related downtime. In this way the final OEE value will still be the same but it will provide a clearer view of what is causing the losses. Figure 7 shows the modified way of calculating OEE.

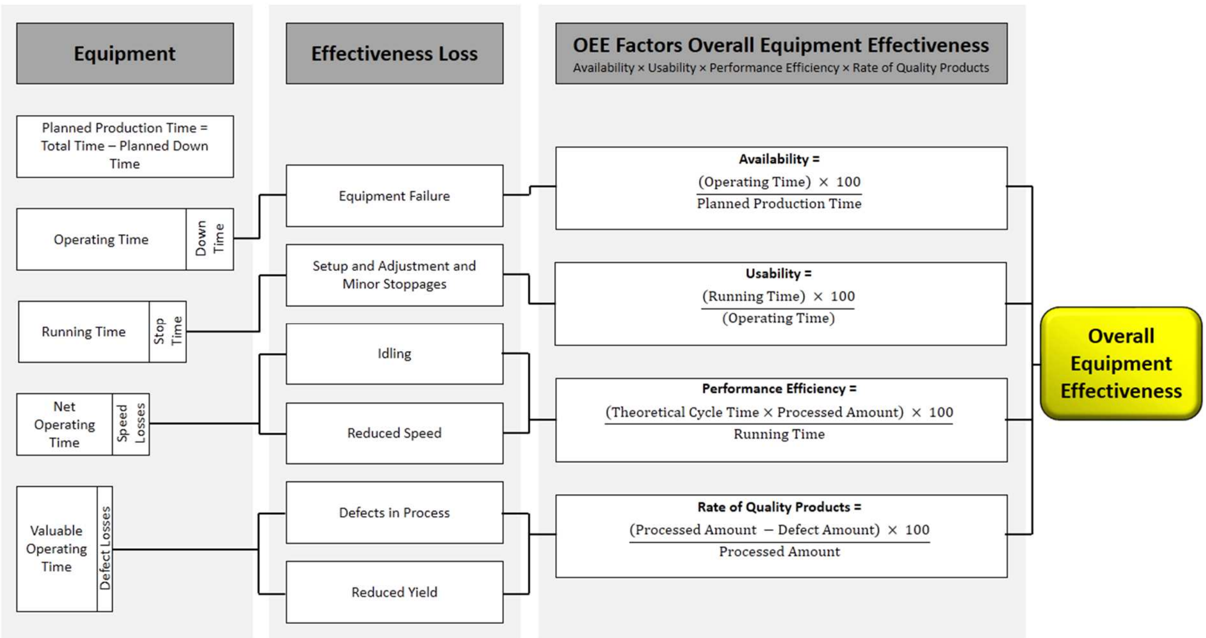

Figure 7. Modified OEE calculation based on Badiger et al. (2008).

3 Methodology

This chapter presents the methodology for the project. The project has been carried out in several steps where the subsequent activities have been dependent of the previous activities to be finished before it could be started. Parallel to this, the data collection was performed. An overview of the projects proceedings can be seen in Figure 8.

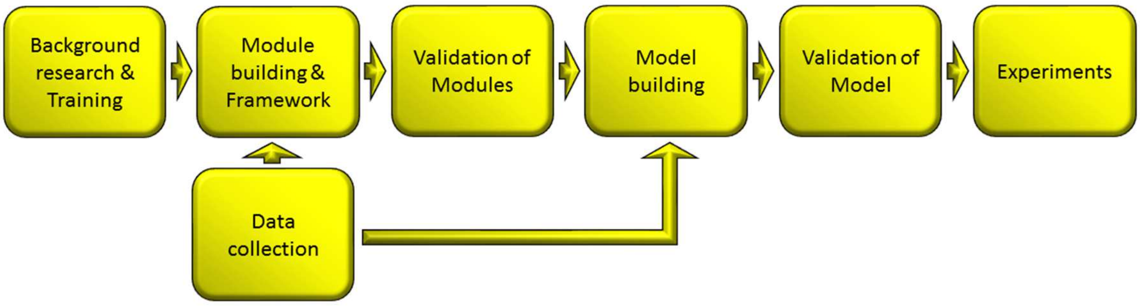

Figure 8. Project overview.

3.1 Research and training

The project has been based on known simulation methodology, data collection strategies and ways to do performance measurements. In order to achieve the necessary knowledge, a literature study was carried out as a first step of the project. As mentioned in the theory chapter, manufacturing simulation requires good knowledge about all parts of the manufacturing system. Knowledge in machine standards and production line configurations were provided by experts at FlexLink and basic knowledge about the specific line was acquired by thorough on site studies of the production line and discussions with the staff at the food producing company. The software 3DCreate with the FLDT add-on had been assigned for the project and necessary training in the software was taken through tutorials.

3.2 Data collection

Alongside the module and model building a data collection was carried out to retrieve necessary input data for the modules and the model. The data collection was carried out both manually and automatically and both quantitative and qualitative data was pursued. As a first step, the overall behavior of the line was studied to find out if some of the machines were similar enough to be represented by the same module. The base of these modules are defined as frameworks and consists of identifiable and measurable features. With the framework definitions as foundation, a classification of data availability was performed. Category A data was available in some form, and different methods presented in 2.1.3.1 had to be used. For category B data, raw data had to be collected in order to support the DES model, and aforementioned methods had to be applied.

3.2.1 Quantitative data

Quantitative data needed to be collected were divided in two distinctions; high frequent events and low frequent events. Examples of interesting events can be seen in Table 3.

Table 3: Categorization of events by frequency

High frequent events	Low frequent events
Cycle times	Failures
Deadplate disturbances	Repair times
Bounce disturbances	
Acceleration curve	
Deceleration curve	

As the two types of events differ highly in their nature, two separate approaches were chosen to collect the data.

3.2.1.1 High frequent events

The method used to collect high frequent data is based on manual identification and recording of events. As the frequency of events is high, around three events per second, video recording of the process was used to be able to study the process slower than real time. Events were identified during the video study and time stamps relative to the start of the video recording were noted in a spreadsheet. The time between events were then calculated to transform the data to the information needed and categorized using the correct SI unit. Each data collection point was contextualized through using a separate tab in the spreadsheet identifying the data point.

3.2.1.2 Low frequent events

As the low frequent events happen seldom, a manual data collection was not practical as it would require too much time. An automatic data collection system was set up using the Programmable Logic Controller (PLC) controlling the line to identify when a machine broke down and started up. To find stops that were independent from the production line dynamics, modifications to the existing PLC logics were implemented to identify a stop as when the machine is not running and at the same time is not blocked or starved. The logics implemented in the PLC is visualized in Figure 9.

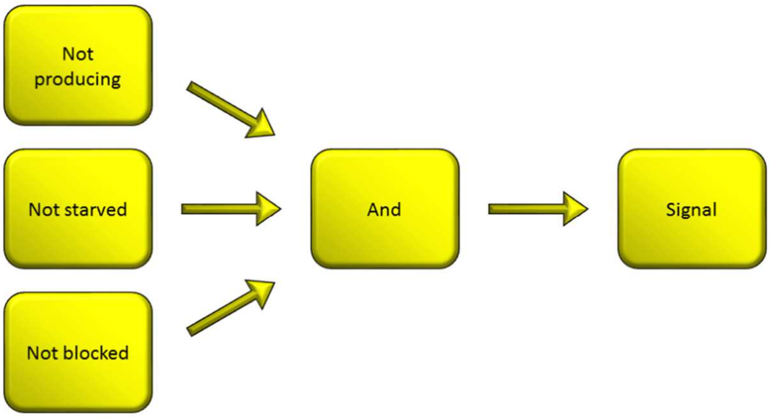

Figure 9. Logics for stop identification implemented in PLC.

An existing device for automatic stop collection and categorization, connected to the PLC was used for recording and for contexualization of the data. The setup of the automatic data collection regarding implementation in the PLC, connecting the PLC to the device for automatic data collection and export of data from the device for automatic data collection to a spreadsheet was done by experts on the system.

Time stamps of starting time and ending time for each stop was recorded in one spreadsheet for each data collection point. After events recorded in non-scheduled time and events affected by changeover activities was corrected, calculations to find the duration of the stops and the time between stops was performed. The information was then transferred to the corresponding tab in the same spreadsheet used for high frequent data.

3.2.1.3 Validation of automatic data collection

The automatic data collection system was validated by forcing stops for each of the data collection points on the production line in order to control if the system recorded an event on the time of the known stop.

3.2.2 Qualitative data

Each modelled machine was studied using both factory visits and digital camera recorded films to identify the overall behavior of the machine, as well as the control logics. Experts on the actual production system as well as experts in production systems in general was interviewed in order to focus the data collection on the most important factors to study.

3.3 Framework

Key properties and adjustabilities commonly needed to generate general reusable modules for all different types of machines present were identified through on-site studies of the production system. The criteria for selecting these key factors were that they should be observable and measurable. This was to make it possible to collect the necessary data to adapt the modules to the site specific conditions, which was the aim with the module. The framework is based entirely on observations made on the machines at the food producing company and the experience of the project team in 3DCreate gained through this project. The observed adjustabilities used as a base for the simulation modules can be seen in Table 4.

For simple machines, typically machines that do not move the products from the conveyor and do not have an in feeder, the properties identified as important to include are dimensions, time between failures, time to repair, cycle time and outlet behavior. Dimensions are important to include so that the module can be modified to replicate the size of different machines. Time between failures and time to repair as well as cycle time needs to be adjustable so the module can be given the unique parameters of a certain machine. Even though the cycle time can be represented without any variance, the distance between products on the conveyor can have variance due to bounces and other disturbances, thus outlet behavior exists and has to be included in the module. Machines at the food producing company that can be considered as simple are the lid applicator and the dishwasher.

For more complicated machines there are more properties that may have to be included in the module. Except for the properties necessary for simple machines also inlet logics, and ramps for acceleration and deceleration may have to be implemented. Start/stop logics needs to be implemented and has to be adjustable so the module can be set up according to the operation of the machine to mimic. If the infeed to the machine is controlled by an infeed screw this behavior has to be modeled and made adjustable. If the machine has significant ramps for acceleration and deceleration these should also be implemented as adjustable properties in the module. Machines at the food producing company that holds any of the above behaviors are the labelers, the scanner and the filling machines. The filling machines holds all of the above behaviors.

For mergers and diverters the behaviors found necessary to include in the module are dimensions, inlet logics, switch logics and outlet behavior. The dimensions, inlet logics and outlet behavior should be implemented in the same way as for the machines. The switch logics needs to include adjustable parameters for stopping of flow, switching time and rules for making diversions.

In order for all modules to work well with FLDT they should have a functionality to snap directly to an existing conveyor. Any applications which are used together with sensors should be modeled in a way that gives the user the opportunity to connect the sensors and specify their functionality. For machines a functionality to momentarily take down a machine to study certain behaviors is convenient. Also, to aid the analysis of the flow, each machine is equipped with an OEE calculator based on the Pack ML states.

Table 4. Key factors to form the framework for simulation modules

Framework key factors
Dimensions
Failures and repair times
Cycle time
Start/stop logics
Infeed logics
Outfeed behavior
Speed dynamics
Switch logics

3.4 Module building

In advance of the actual module building, during the on-site studies at the food producing company, it was identified which machines that needed to be represented through a new module and which machines that could be represented using the modules already available in the standard library in 3DCreate. To be able to evaluate collisions between products at the line, the flow of products in the final model needs to be very accurate compared to the real system. This puts high demands on the level of detail of the modules and the way they are representing the flow of products. It was found that for the robot cell in the beginning of the line and for the packing process, which both are not interesting from a collision perspective, modules from the standard library were enough. For all other machines new modules had to be created.

Building of the modules were carried out according to the modular discrete event simulation methodology proposed by Johansson (2006). To aid usability when using the modules to build a model the machines with similarities in behavior and design were modeled as one adjustable module. Also for the modules to interact well with FLDT as well as to imitate the real world installations at the food producing company they were given a functionality to snap on to an existing conveyor system.

3.4.1 Validation of modules

The validation process was divided so that validation was carried out separately for the modules and the model according to the methodology proposed by Johansson (2006).

The general functionality of the models were first validated together with a production systems expert at FlexLink, comparing the behavior of the modules with his experience of similar machines. Especially the flow of products through the machines was studied and that the modules were prepared to replicate any disturbance found in each application. It was then validated that the adjustability of the modules were sufficient to capture the site specific prerequisite at the food producing company by comparing their behavior using the specific parameters filled in, with the functionality of the real machines in the production line.

3.5 Model building

With modules ready work continued with building the model. 2D drawings of the production line were imported into FLDT and the conveyor layout was built on top of those drawings using the built in FLDT functionalities. Figure 10 shows an example of conveyors built on top of a 2D drawing. This was identified as the fastest and easiest way of building the conveyor layout as conveyors could be built with a few clicks instead of pulling modules from a digital library using drag and drop functionality. This method also ensured that the layout was built according to the right dimensions, as functions as total length of a conveyor and relative positioning are included in FLDT.

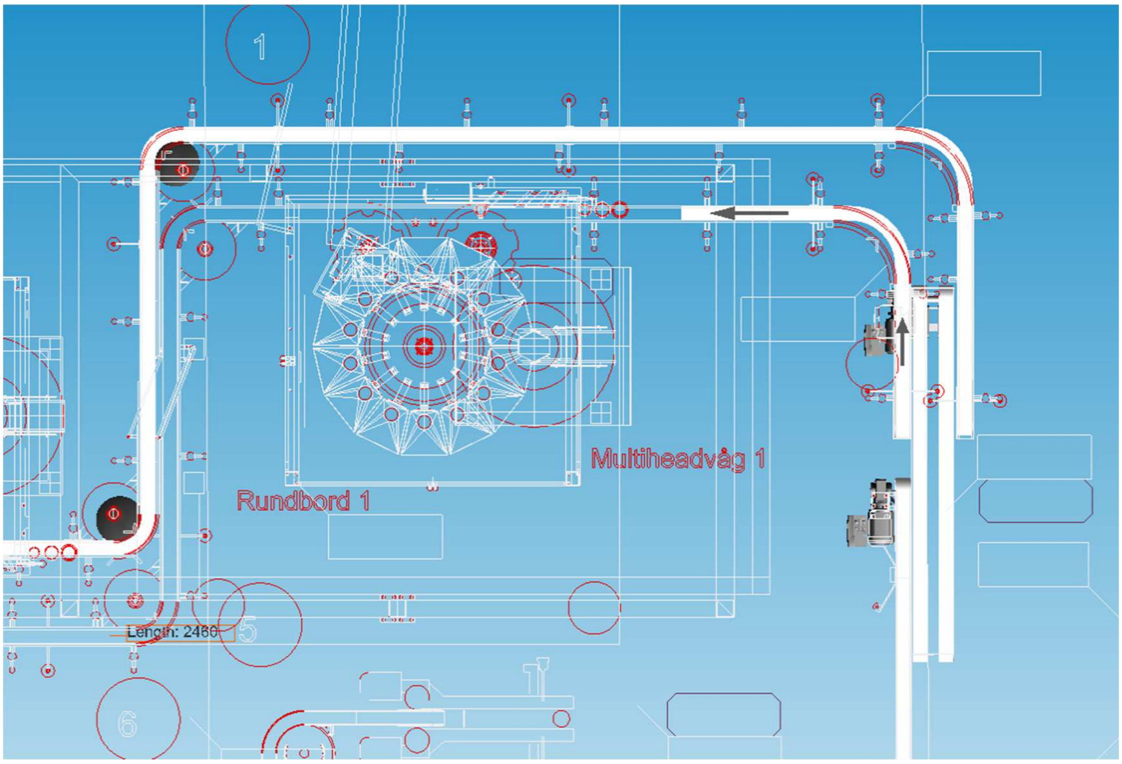

Figure 9. Example of a conveyor being built on top of a 2D layout. The thin lines are parts of the 2D layout, and the thick, white lines are the conveyors.

With the conveyors in place the machine modules were snapped on to the layout at correct places and adjusted to mimic the site specific machine it was representing. Then the sensor modules were snapped on to the conveyors and connected to the right machines. A picture of a machine together with connected sensors can be seen in Figure 11. The buffer tables placed in the conveyor system are represented by a standard module from the 3DCreate module library.

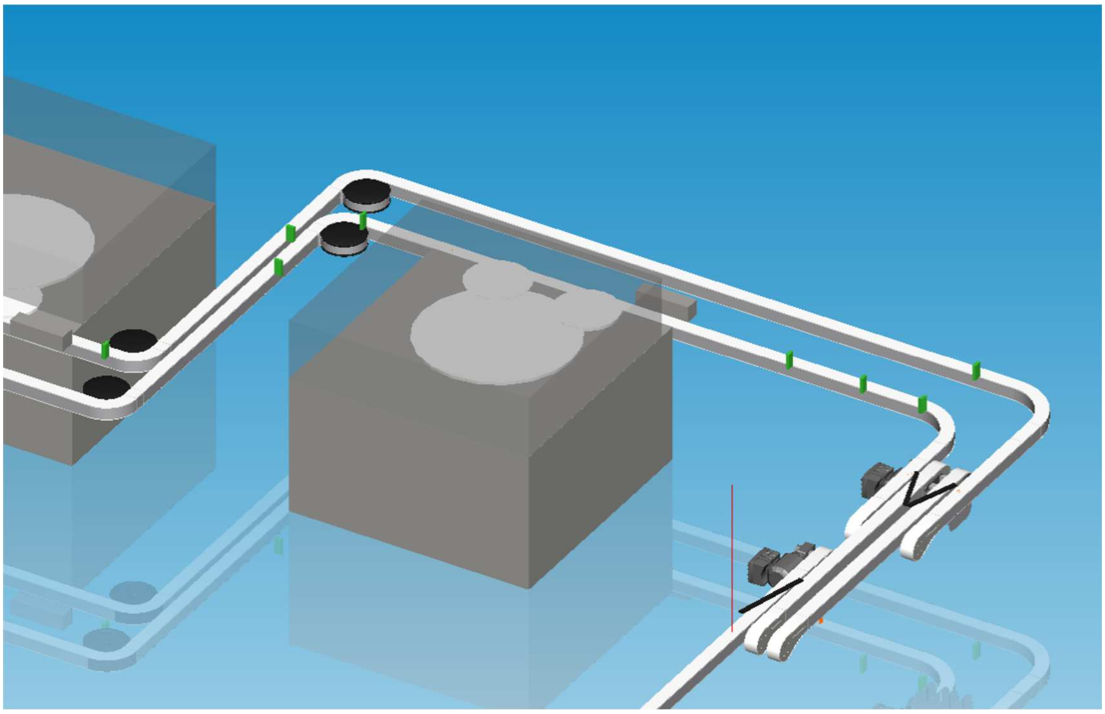

Figure 10. Machines snapped on conveyors with connected sensors.

The loading cell at the beginning of the line consists of a robot loading empty glass jars from a euro pallet on to a wide conveyor that marks the starting point of the conveyor system. The model of the robot cell can be seen in Figure 12. This part of the model was built using the existing modules embedded in the 3DCreate module library as a base, but with some alterations. To overcome a limitation in 3DCreate, which does not allow moving several products in parallel at one conveyor the products fed into the model were adjusted to look like an entire row of glass jars. To solve the representation of the functionality where the glass jars are reordered to a single line and fed into the narrower conveyor a new component had to be created. The component replaces the products looking like rows of glass jars by single glass jars that continue to move throughout the layout. In order to control the working cycle of the robot as well as to make the robot lift the correct amount of products each time slight modifications had to be made to the inner functionality of the module controlling the robot. Due to the highly customized functionality of this component it was not modeled as a general changeable module.

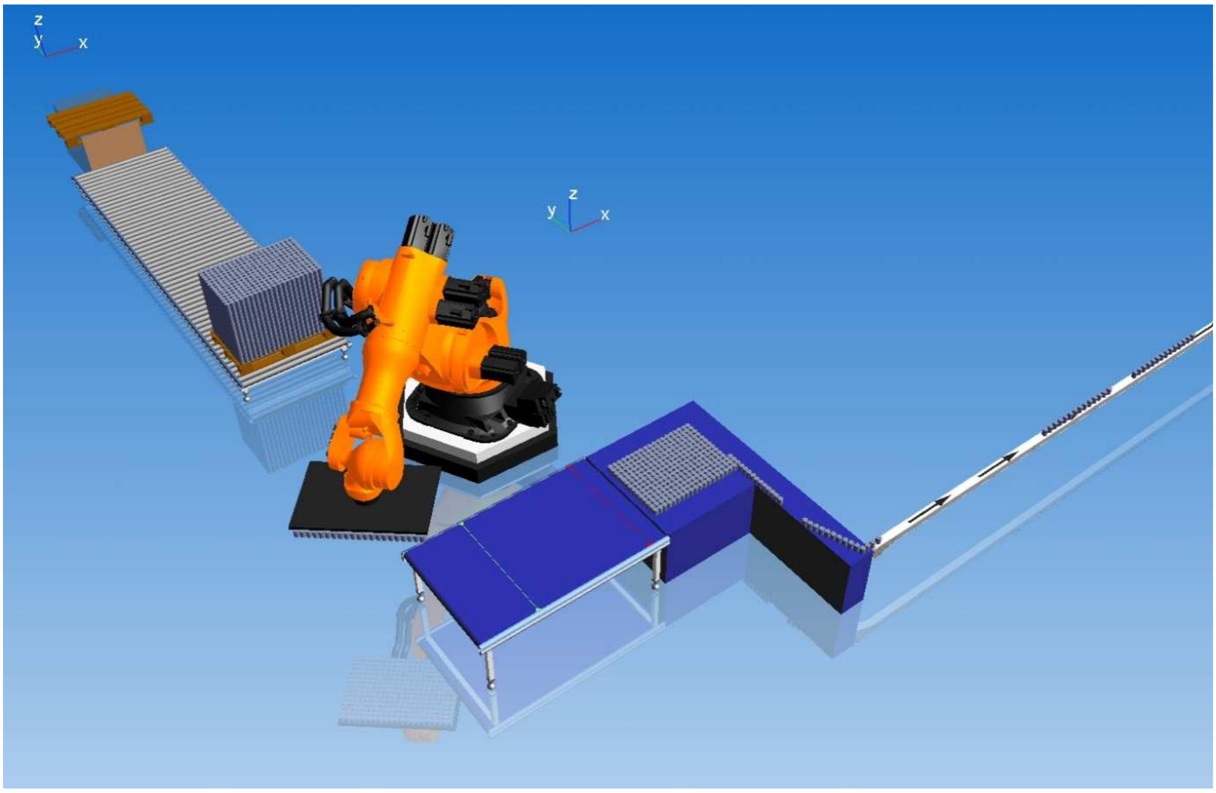

Figure 11. Model of the robot cell in the beginning of the line.

The packing process at the end of the line was built completely with standard modules from the 3DCreate library. To get a close enough representation of the functionality of the real packing process, the process is built with modules placed in a certain combination. There were no modifications done to the inner functionalities of any of these modules. The model of the packing process can be seen in Figure 13.

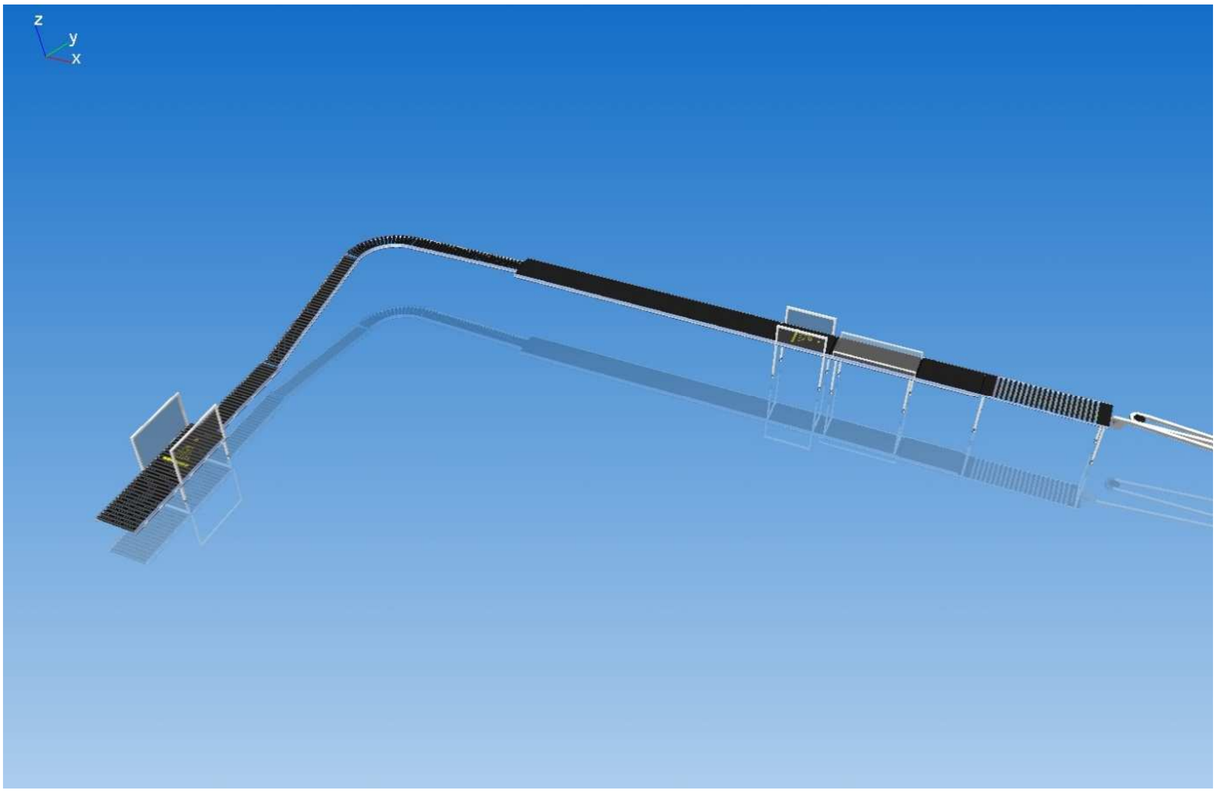

Figure 12. Model of the packing process.

3.5.1 Validation of model

When validating the model all machine specific parameters as stated in the framework for each machine at the line had been implemented based on the collected data, except for the failure and repair data. Validation of the model was performed together with experts on the production system, in the roles as line supervisor and technical responsible engineer.

3.5.1.1 Validation of logical connections

The actual validation process consisted of validating the logical connections controlling the start and stop of the machines. As those logical connections dictate the flow in the line, a valid behavior was fundamental both for the productivity analysis and the collisions analysis. As an indicator, accumulation was studied before each machine to make sure queues appeared on the correct places in the model.

3.5.1.2 Validation of diversion and merge points

The operation of the diverter was studied to ensure that the effects of priority of one of the herring fillers was modelled correctly. At the physical line there is a negative effect on the supply of products to the second herring filler, making it more likely to be starved. At the merge point the behavior when accumulation occur downstream was of great importance. For the second herring filler, the conveyor connecting it to the merger is very short so disturbances downstream will affect the productivity rapidly as there is small capacity for accumulation.

3.5.1.3 Validation of system boundaries

As a mean to speed up the simulation execution time, the intention is that simulation runs are performed only on a part of the production system that include the part including the filling process to the accumulation table after washing. To be able to do that, the assumption that the capacity is good enough upstream of the simulated part of the production system needed to be validated. Also, assumptions regarding the capacity downstream was of interest.

3.6 Experiments

To find suggestions to minimize the collision and at the same time at least maintain the level of productivity a series of experiments were conducted on the simulation model. The experiments were first applied on the base model one by one, and finally an experiment including all suggestions of improvement was conducted.

3.6.1 Collision detection

To be able to detect collisions, a built in function in 3DCreate called collision detection was used. The collision detector was set up to detect collisions in the model only between a chosen product and the two products sitting before and after on the conveyor. During the collision detection tests, one product was chosen to be sampled for collisions each 10 seconds. Then all collisions encountered by the product along the studied route was recorded. To record a collision, two events had to happen; first two products had to have contact, and then be released. The collected data consisting of one entry for each sampled product, chronologically ordered, was then exported to a spreadsheet and plotted.

3.6.2 Productivity

The productivity was measured using the OEE calculator built in the herring fillers. During the experiment, the accumulated OEE for each herring filler was recorded each 10 seconds. The collected data consisting of one entry for each sampled time, sorted per herring filler and chronologically ordered, was then exported to a spreadsheet and plotted.

4 Results

In this chapter all results from the project are presented. The project have been carried out in several smaller steps, all which have led to individual results. Together the results fulfill the aim of the project. Below follows a presentation of each of the results.

4.1 Created modules

When applying the framework three groups, defining similar machines, were identified. The most advance group of machines was the filling machines which are the only machines at the line that moves the products off from the conveyor system. Due to the complex behavior of these machines all subparts of the framework had to be taken into consideration when creating a module for that group. The other machines could be divided between simpler machines with and without infeed screw. For these machines only parts of the framework had to be taken into consideration. Apart from the machines also modules for diverters, mergers, loose infeed screw, twister and sensors were created. All the modules created in the project are presented more in detail below.

4.1.1 Filling machine

All filling machines at the production line were operating in the same way and had roughly the same behavior which made them suitable to model as one adjustable module. The machines are built up of three rotating disks, one smaller infeed disk, one bigger main disk and one smaller outfeed disk. The actual filling is performed at the main disk without stopping the constant motion of the disks. Products are fed into the machine by an infeed screw.

The module created for the filling machines required many adjustable features in order to imitate the design and behavior of the real machines. All dimensions had to be adjustable so that the module could be used for machines with different outer dimensions and different sizes for the disks. The module also had to be possible to mirror to accommodate being placed on either side of a conveyor. For the operational properties the speed, acceleration and deceleration had to be adjustable as well as time between failures, time to repair and disturbances in the outlet dynamics. An interface for connecting sensors to the machine also had to be in place to enable for the start/stop logics.

To aid usability in companion with FLDT the module has been given a functionality to snap directly to a conveyor and to support statistical gathering the module is operating in the PackML states and is equipped with an OEE calculator. There is also a possibility to manually take down the module during simulation. The full module specifications for the filling machine module can be seen in Appendix I.

4.1.2 Simpler machines

Except for the filling machines all other machines at the production line could be considered as rather simple. These machines where fitted on top or at the side of a conveyor which were running through the machine. This reduced the complexity of the flow going through them and meant that modules representing the machines could be built where only parts of the framework had to be taken into consideration.

There were two modules created to represent these simpler machines, one with an infeed screw and one without. Except for the infeed screw and start/stop logics the modules have the same features. They both have two different designs to choose between and are adjustable in all outer dimensions as well as made able to mirror so that they can represent machines with different sizes and design. For operational features they had to support adjustable parameters for speeds, time to failure and time to repair as well as outfeed disturbances. For the machines with infeeder screw, start/stop logics and the possibility to connect sensors was included. There were no identifiable acceleration or deceleration ramps in any of the simpler machines. This part of the framework could therefore be left aside.

Like the filling machine module these modules were given the functionality to snap directly to a conveyor, are operating in the PackML states, are fitted with an OEE calculator, and are possible to take down manually during simulation. Full specifications for these modules can be seen in Appendix II and Appendix III.

4.1.3 Diverters and mergers

There were two kinds of diverters and mergers at the food producing company. There were manual ones which were set to a fixed position during changeovers to direct the flow in a certain way and there were static ones that automatically changed position during production to divert the flow into two separate lanes or to merge the flow from two separate lanes into one lane.

The manual diverter and mergers work in the same way, as both basically divert products from one conveyor to another. One module was created to represent both functions. The only parts that are included from the framework are dimensions and switch logics. For the manual diverter, the switch logics is basically the manual movement of the diverter arm from one position to another.

The static diverter also include a stopper to provide inlet logics and is used together with a diverter controller unit providing switch logics and an interface to connecting the diverter, the stopper and the sensors.

A merger controller unit was used together with two manual diverters, two stoppers and a sensor for merging two flows on two conveyors to a third conveyor. The merger controller unit provided switch logics, and an interface connecting the diverters, stoppers and the sensor.

All diverter components are prepared for end-to-end connection to be able to use the built in function for space utilization, which means that they are not prepared to snap on to the conveyors. Full specifications for the diverter and stopper modules can be seen in Appendix IV and Appendix V.

4.1.4 Loose infeed screw

The loose infeed screw is operating in exactly the same way as the infeed screws fitted to some of the machine modules. The reason why this module is needed is because there were two machines in the production line where the infeed screws were placed some meters in front of the machine and the simplest way to replicate this is by having a loose infeed screw to place on the line together with a machine module without any infeed screw.

The loose infeed screw is a quite simple module. It contains functionality to snap it directly to a conveyor and to connect sensors to it for the start/stop logics. It also have an adjustable parameter for the speed of the module. Full specification of the module can be seen in Appendix VI.

4.1.5 Twister

The twister is an application where the jars are pushed through a spiral that turns the jars over to make sure they are empty. It has a simple functionality and thus is also represented by a simple module.

The module has adjustable parameters for length and speed and a functionality to connect sensors for the start/stop logics. It can also be snapped directly on a conveyor. Full module specifications can be seen in Appendix VII.

4.1.6 Sensors

There were a lot of sensors fitted to the production line. The sensors were controlling the flow of the line by detecting products and sending signals to either start or stop machines or other equipment. The sensors works in the same way throughout the entire line and were modeled as one module that could be connected to any kind of equipment.

To see when the sensor is detecting or not during simulation the module has been given a functionality to change color. When the sensor is detecting it turns red and else it is green. To aid usability the module has also been given a functionality to snap directly to a conveyor. Connecting a sensor to another module is done from the user interface of that module. Full specifications of the module can be seen in Appendix VIII.

4.2 Base model measurements

To form a base line to use for comparison of results from experiments, the productivity of the simulation line herring fillers was recorded, together with the total number of collisions encountered by sampled products along a part of the simulation model, from the start of the filling process until just before the diverter.

The simulation was run for 5430 seconds, and disturbances of 60 seconds each was induced on the first machine in the line after 1900 seconds and 3860 seconds. The measured productivity can be seen in Figure 14 and the measured number of collisions can be seen in Figure 15.

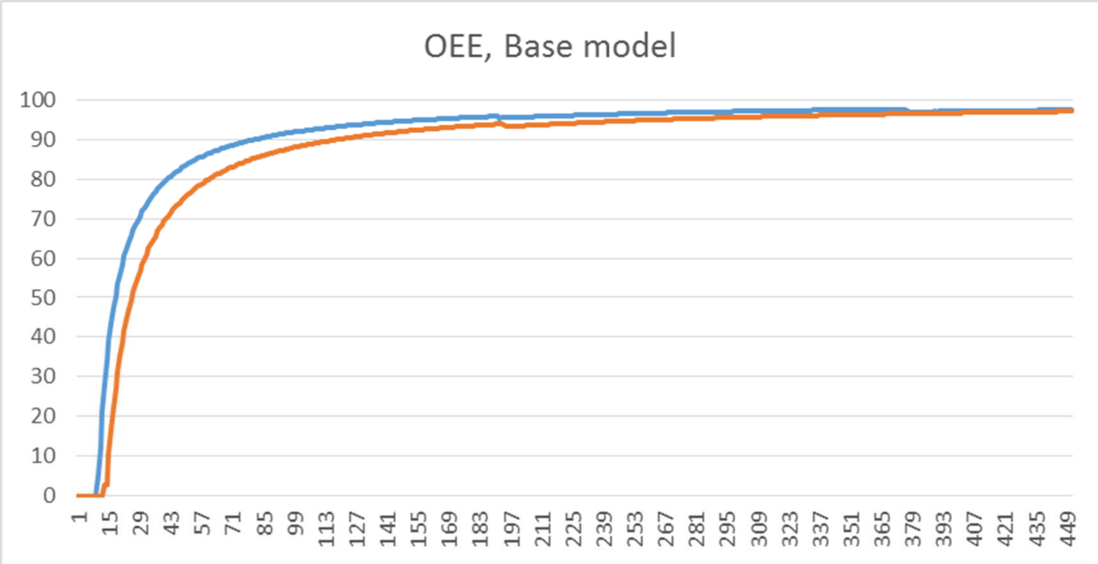

Figure 13. Accumulated OEE values per herring filler. The horizontal axis indicate sample number and the vertical axis indicates the OEE percentage.

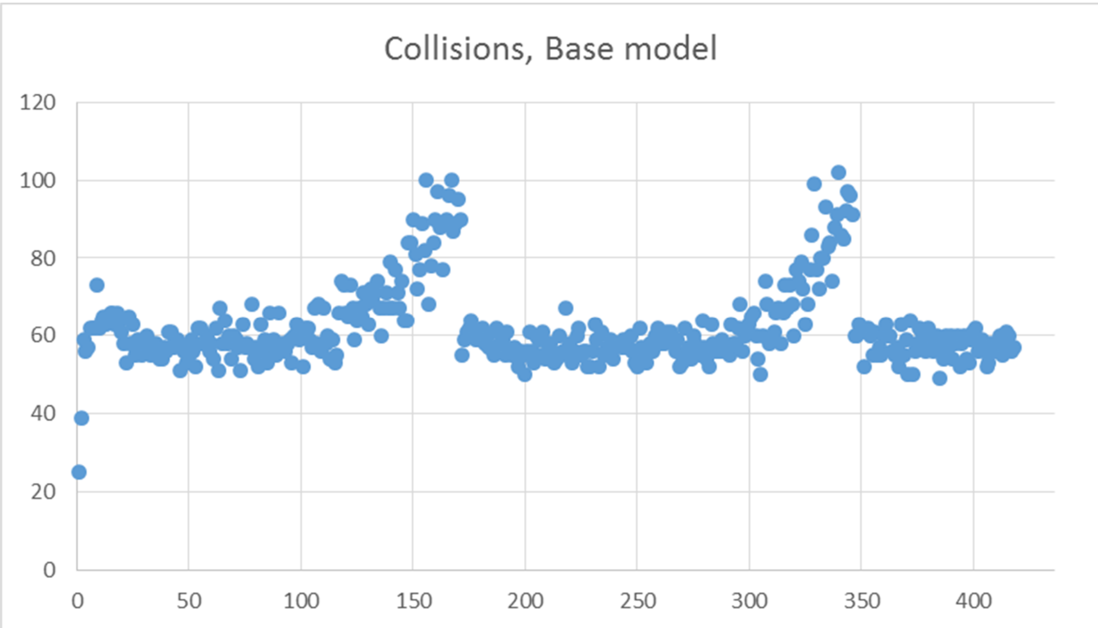

Figure 14. Number of collisions encountered for sampled product. The horizontal axis indicates the sample number and the vertical axis indicates the number of collisions encountered.

4.3 Experiments

Experiments were conducted in order to decrease the number of collisions and increase the productivity of the herring fillers.

4.3.1 Experiment 1, less inflow of products

To reduce the number of collisions, the first experiment was conducted using an inflow of products close to the maximum capacity of the herring fillers, as opposed to the significantly higher inflow of products than the capacity of the herring fillers experienced in the base model. The simulation was run for 5430 seconds, and disturbances of 60 seconds each was induced on the first machine in the line after 1900 seconds and 3860 seconds. The productivity measured can be seen in Figure 16. The measured number of collisions can be seen in Figure 17.

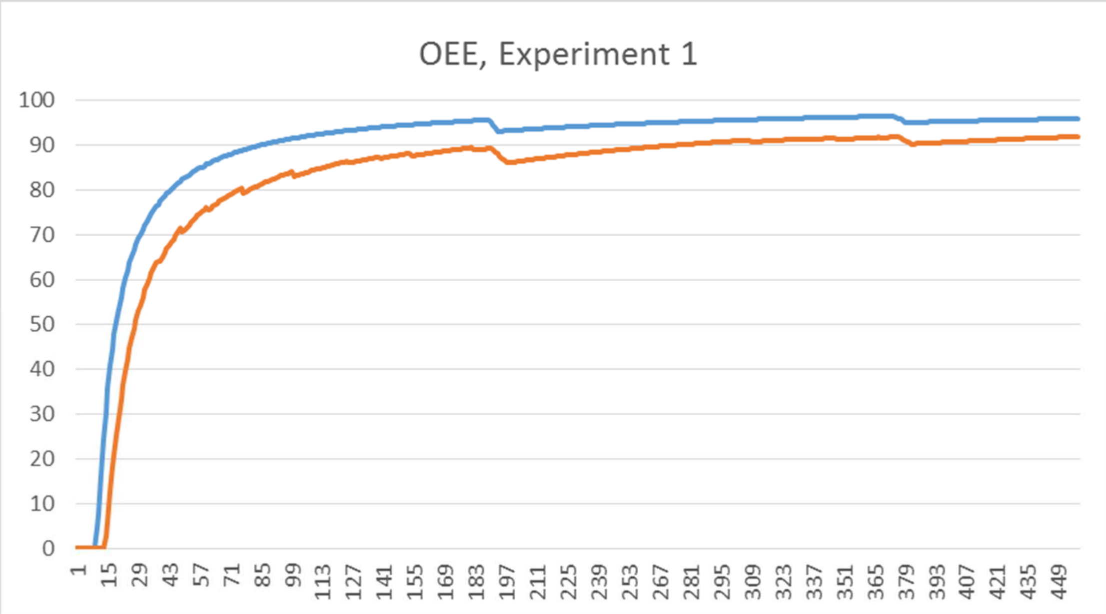

Figure 15. Measured OEE values per herring filler. The horizontal axis indicate sample number and the vertical axis indicates the OEE percentage.

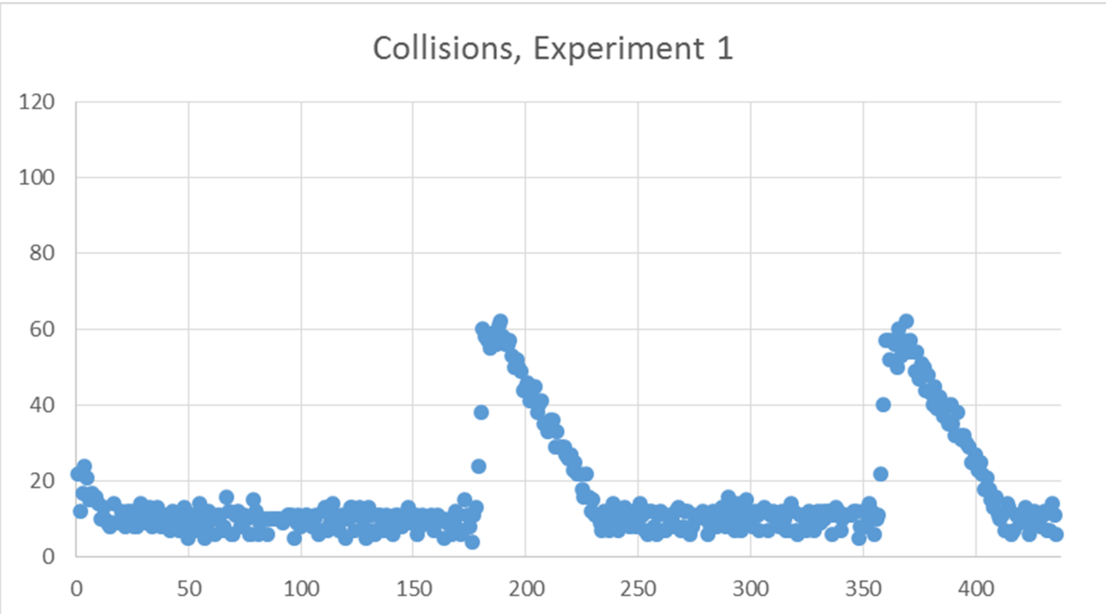

Figure 16. Number of collisions encountered for sampled product. The horizontal axis indicates the sample number and the vertical axis indicates the number of collisions encountered.

4.3.2 Experiment 2, faster diversion and merging

In the previous experiment, applying a strategy of less inflow of products, it was obvious that herring filler 2 tended to be starved due to the logics and functionality of the diverter dividing the flow between the herring fillers. To encounter this problem a concept with a fast switching, dynamic diverter were tested. This diverter changes position every 0.4 seconds without stopping the flow. The simulation was run for 5430 seconds and two disturbances of 60 seconds each were introduced, one after 1900 seconds and one after 3860 seconds. The productivity measured can be seen in Figure 18. The measured number of collisions can be seen in Figure 19.

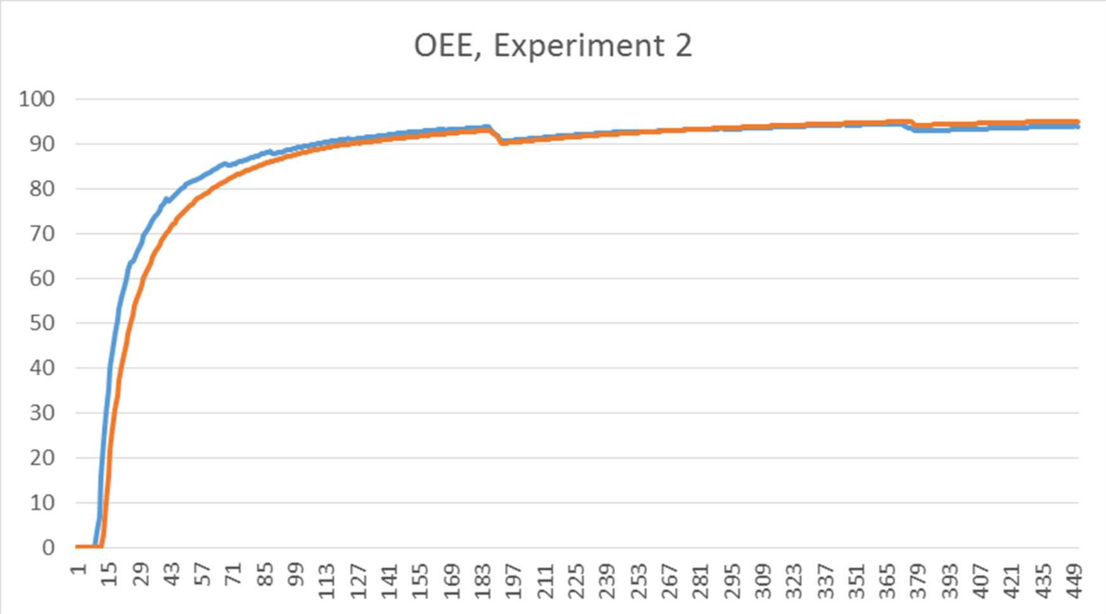

Figure 17. Accumulated OEE values per herring filler. The horizontal axis indicate sample number and the vertical axis indicates the OEE percentage.

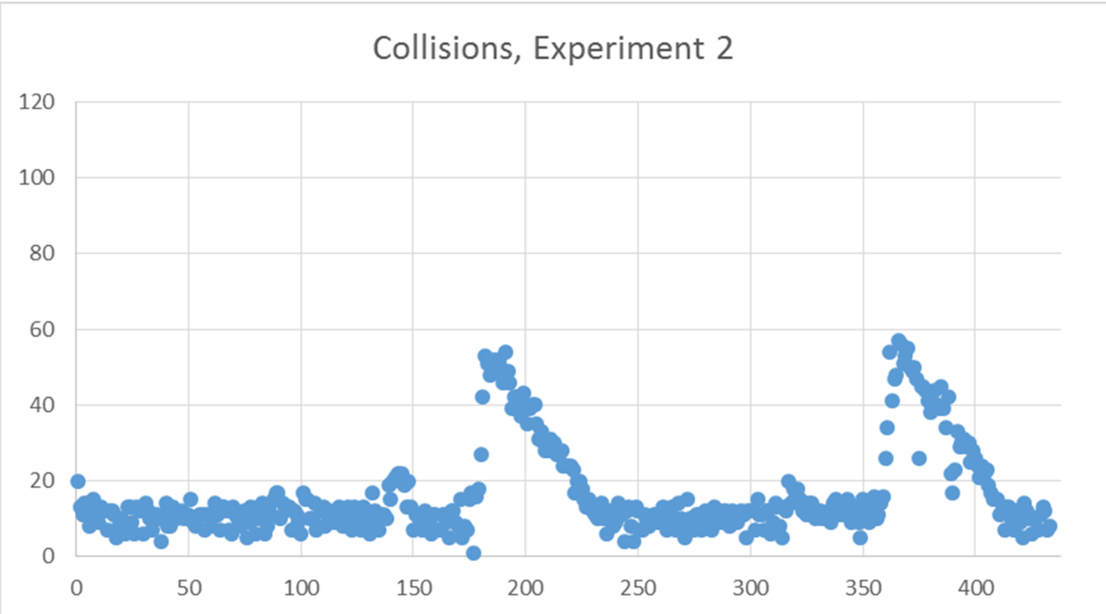

Figure 18. Number of collisions encountered for sampled product. The horizontal axis indicates the sample number and the vertical axis indicates the number of collisions encountered.

4.3.3 Experiment 3, dynamic control of machine speed

After experiment one and two the model still did not perform an OEE value as high as the one measured in the base model. In an attempt to increase the OEE value, controllers were applied to the machines in front of the herring fillers as well as for the infeed of products. The controllers looked one step ahead in the process and changed the speed of a machine to either be the same as the combined speed of the herring fillers, 20 % above that speed or 20 % lower than that speed. The speed was changed depending on if the machine downstream had normal, too much or too less accumulation. The simulation was run for 5430 seconds and two disturbances of 60 seconds each were introduced, one after 1900 seconds and one after 3860 seconds. The productivity measured can be seen in Figure 20. The measured number of collisions can be seen in Figure 21.

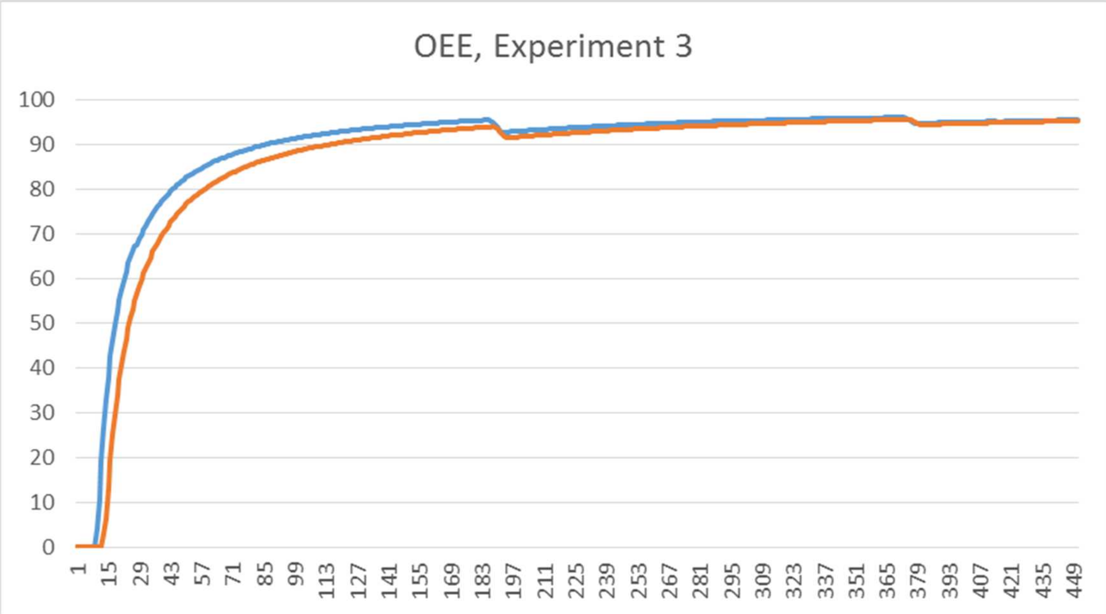

Figure 19. Accumulated OEE values per herring filler. The horizontal axis indicate sample number and the vertical axis indicates the OEE percentage.

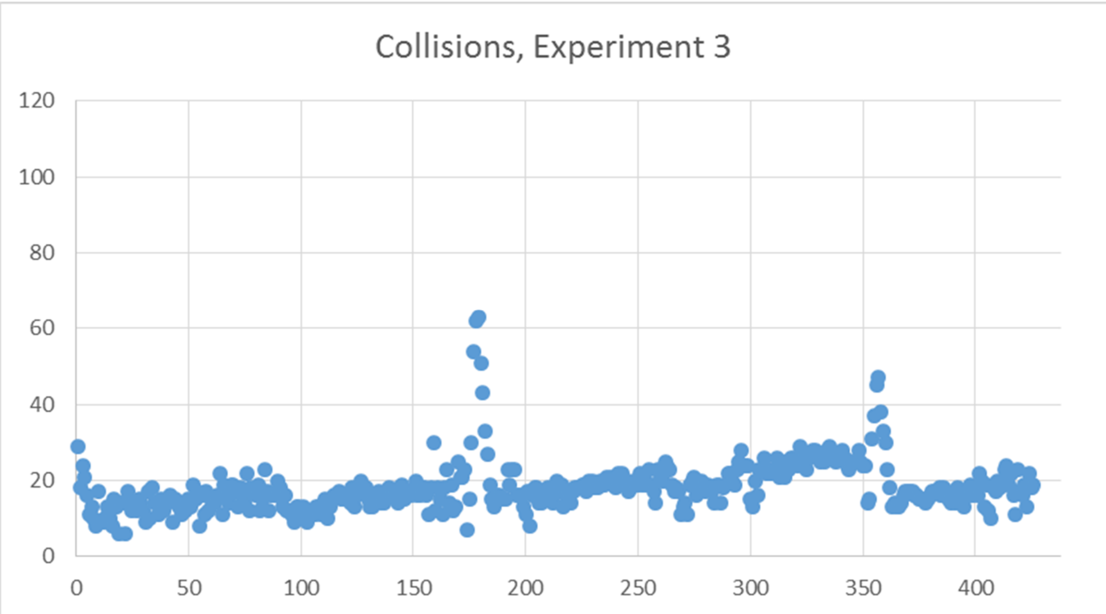

Figure 20. Number of collisions encountered for sampled product. The horizontal axis indicates the sample number and the vertical axis indicates the number of collisions encountered.

4.3.4 Experiment 4, added buffer in front of herring fillers

As a fourth experiment a buffer was added in front of the herring fillers. This was done to catch the OEE loss happening during a break down. In the base model there are accumulation on all conveyors in front of the herring fillers so when a break down occur on a machine the herring fillers are still fed with products from the conveyors. In the experiment model this is not the case as the heavy accumulation of products on the conveyors is taken away. Therefore the buffer is there to keep feeding the herring fillers with products during a break down upstream. In this experiment the machine controllers are set up to order the machines to run in high speed mode if the added buffer contains fewer than 175 products, which is the same amount of products as the conveyers are accumulating when full. The simulation was run for 5430 seconds and two disturbances of 60 seconds each were introduced, one after 1900 seconds and one after 3860 seconds. The productivity measured can be seen in Figure 22. The measured number of collisions can be seen in Figure 23.

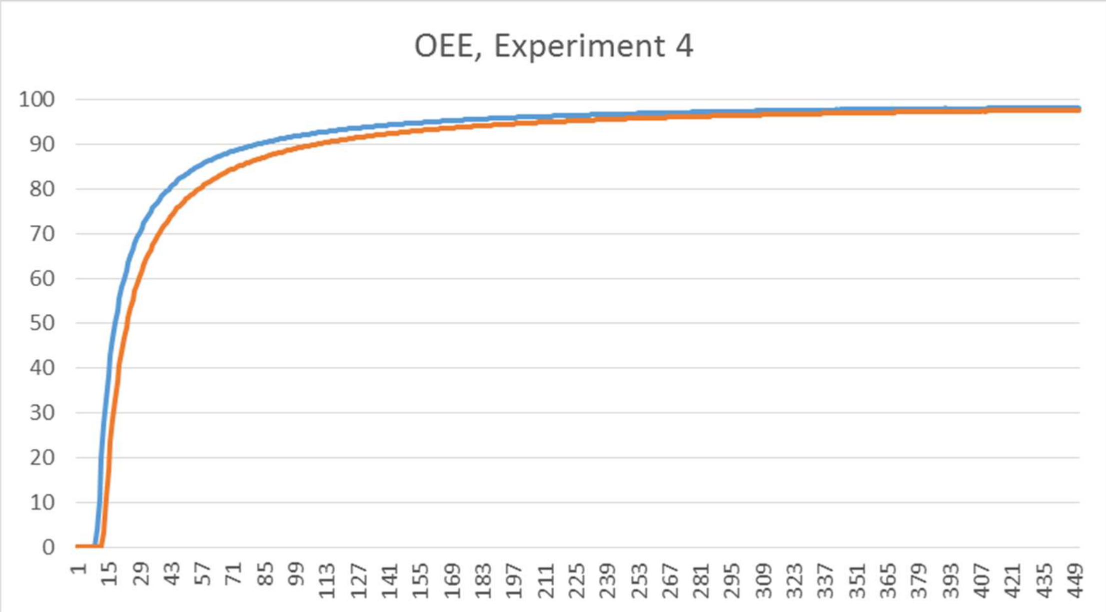

Figure 21. Accumulated OEE values per herring filler. The horizontal axis indicate sample number and the vertical axis indicates the OEE percentage.

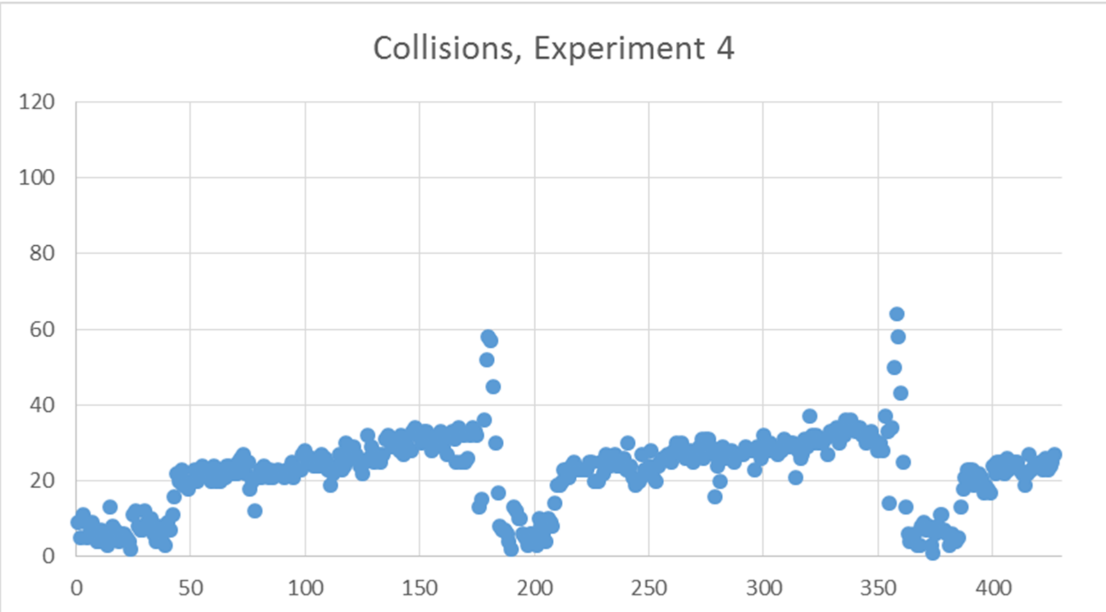

Figure 22. Number of collisions encountered for sampled product. The horizontal axis indicates the sample number and the vertical axis indicates the number of collisions encountered.

5 Discussion

The used method and the results from the simulation runs are discussed in order to draw conclusions on the goals for the project.

5.1 Simulation results

Each simulation run is discussed separately as each experiment was performed to address different problems. As the experiments were conducted in succession, where each experiment aimed to address issues identified in the last experiment, they are discussed in the same way. The final experiment is also discussed in relation to the base model results.

5.1.1 Base model

From the base model simulation run, it could be seen that both herring fillers had a high productivity, reaching OEE values over 90%. At the other hand, most products encountered over 60 collisions passing the area where collisions were monitored. The induced disturbance had a direct impact on the productivity, causing the OEE to drop slightly. As the number of products entering the system was higher than the number of products leaving the system, the accumulation on the conveyors increased until the disturbance. The accumulation led to more collisions, with some products experiencing more than 100 collisions.

5.1.1.1 Validity of the result

The base model is validated for productivity and collision detection purposes. As there are no stochastic disturbances included in the model, long time simulations are not useful but tests of different cases are. The results regarding productivity are to be seen as an indicator for the performance of the base model, and the absolute accuracy is not validated in relation to the production system. The collected collision data is only regarding the number of collisions. As collisions on the line happen in different relative speeds, some collisions are worse than other. The collected data does not include information of the relative speed of the products involved in the collision.

5.1.2 Experiment 1

As the number of collisions grew when the accumulation was spread in the system, the inflow of products to the system was decreased as a mean to lower the accumulation of products on the conveyors. As a result, a drop in productivity could be experienced on herring filler 2. Herring filler 1 is prioritized in the diversion of products and that can be seen in the productivity that is still over 90%. The lower inflow of products eliminated the accumulation of products upstream of the first machine, causing much lower numbers of collisions. Most sampled products encountered less than 20 collisions.

The effect on the collisions from the disturbances was obvious. When the first machine was not producing, the inflow of products accumulated upstream of the machine causing the numbers of collisions reaching what could be observed in the base model results.

5.1.3 Experiment 2

From experiment one an uneven workload between the two herring fillers were experienced. To deal with this an experiment with a faster diverter were conducted. The experiment was fruitful in terms of evening out the workload but also slightly lowered the amount of collisions in the system due to the fast diverter not stopping the flow when changing position. As the diverter changed position every 0.4 seconds the short conveyor leading from the diverter to herring filler 1 was no longer a problem. Thus the priority set for herring filler one in the diverter could be scrapped. This also lead to an increased OEE in relation to experiment 1, but still less productive than the base model.

5.1.4 Experiment 3

The third experiment conducted was to apply controllers to the machines in an attempt to increase the OEE value. The experiment showed a slightly higher OEE value than the previous experiment but still not as high as for the base model. What was more fruitful was the effect on the number of collisions. The model showed a much faster recuperation to a steady state with low number of collisions after a disturbance.

Another potential benefit with the dynamic control of the machines is that the machines are not starved or blocked during normal production as they are in the base model. This enables the machines to run without starting and stopping during normal production which can have a positive effect on the performance of the machines. The only scenario where the machines are forced to stop is when they are starved due to disturbances upstream or blocked due to a disturbance downstream.

5.1.5 Experiment 4

The fourth and last experiment showed an increase in OEE value to a level slightly above the level of the base model and a number of collisions not as low as for experiment 1, 2 and 3 but still significantly lower than for the base model. As the buffer constantly holds the same number of products as are accumulated on the conveyors in the base model during normal production, the protection of the herring fillers against disturbances upstream is at least the same as in the base model. For disturbances occurring to machines close to the herring fillers the protection of the herring fillers will be better than in the base model as more products are buffered near these machines.

As the accumulated products are situated in a defined place in the system in this solution, that gives greater possibilities to control the collisions and noise level derived from the accumulation. E.g. this can be done by using a collision free buffer or cover the buffer to reduce noise.

5.2 Using the chosen method

The used method include a way of working that has greatly affected the final result. The method also include the use of a specific software. These are the most significant factors regarding the method.

5.2.1 Modular discrete event simulation

When building the model with the pre-validated modules, there were some issues regarding the validation. As the modules are general, validation is harder as the models can be used in many ways. For example, when the filling machine module was tested the start/stop logics were tested to handle a high inflow of products making the upstream conveyor filled with accumulated products. The conclusion was drawn that the module could handle accumulated product. Later, when the module was used in the simulation model, in one case products were already accumulated when they reached the machine which caused the start/stop logics to fail. In a classical simulation approach, that problem may have been discovered and handled earlier. The positive was the reusability of modules, which was proven by the model which contained seven filler machines that used the same module, but with different parameters. Also, roughly 50 sensors are used in the model and all use the same module. The modular approach also meant that FlexLinks library of conveyors could be used. Specifically, the visual impression was enhanced as the time needed to create conveyors looking as close to reality was not available in this project.

5.2.2 Problems encountered regarding the software

While building the base model there were some problems encountered which required different sorts of workarounds. It was found that the processor fields used in the modules to enable for the snap on functionality were a little unstable. To get around this the conveyors were split up into smaller sections where machines and sensors were to be snapped on so that all sensors and machines were snapped on to individual conveyor sections. In this way it is still possible to use FLDT for building and configuring conveyor layouts but it takes some more planning to setup the layout in a special manner where the machines and sensors are supposed to be placed.

Another problem also related to the snap on functionality of modules on to the conveyors is the mismatch between processor fields and flow fields which cannot be combined in the same module without interfering with each other. To overcome this the flow field had to be abounded for the snap on modules and the flow of products into the module was instead handled using the grab function. Unlike the flow field however, the grab function do not consider if there is free capacity on the destination path or not which means that if it is full, the products will end up occupying the same space. This was no problem for modules that were stopped when accumulation was created downstream but for diverters and mergers where accumulation of products all the way through the module can happen the use of flow fields was necessary. Thus the possibility to snap these modules on to an existing conveyor system was lost. This meant that when creating the conveyor layout small gaps had to exist where the merger or diverter later could be placed. The easiest way to do this was found to be to build the conveyor as one full conveyor where small conveyor sections were placed roughly where the gaps needed to be. In this way all sections in the conveyor could be configured to have the correct speed through the end drive. When this was done the small sections were taken out of the conveyor and replaced with the merger or diverter module. To make it fit properly the length of the surrounding conveyer sections was adjusted as a last step.

A drawback discovered when building big layouts where lots of products are inside the system at the same time is that the highest possible speed to run the simulation in is significantly slower than for many other simulation software. For the layout created in this project the maximum simulation speed was roughly six to one compared to real world time. The number of products in the system also caused memory problems as products entering the system use system memory, but products leaving the system does not return the used memory causing the simulation environment to crash when a certain amount of system memory usage is reached. In the simulation model used for simulation runs, a closed loop of products is used. At the start of the simulation run a number of products is created into a buffer. Products are then introduced to the first conveyor at a controlled rate, and when they exit the last conveyor they return to the buffer. As the number of products introduced to the buffer is based on the space for products in the model, it does not affect the result.

5.2.3 Automatic data collection

The chosen approach to collection of low-frequent data was based upon the assumption if a machine is not producing and at the same time is not blocked or starved are good enough conditions to define an independent stop. The validation of the automatic data collection only covered the discovery of a stop. When the data collection was finished the collected data did not pass a sanity check; there were a significant amount of short stops recorded on all data collection points, which was not experienced in real life. The extra stops could be a result of poor implementation of the control logics in the machine causing the signal to bounce. There is also a suggestion that there is a superior system controlling the logics, which can cause the machine to stop even though not blocked or starved. As the data collected did not represent the reality, it could not be used in the simulation. This affected the result greatly as stochastic disturbances to the line could not be used. The base model herring fillers produce an OEE figure close to 100% on a one hour run, while the real system has a significantly lower OEE.

6 Conclusions

- Several reusable modular blocks representing machines present on FMCG lines were created in the project. As the modular blocks are based on a framework of features, a simulation user can determine the validity of the modules to the simulation task and also determine which data needs to be collected to support the simulation.
- A simulation model that can predict productivity and number of collision was created. The validity of the model in relation to the real production system could not be granted in absolute numbers, as fundamental data on disturbances could not be included in the model.
- The number of collisions in the model of the production line could be lowered by 65% and the productivity increased by 0.6% by producing only what is consumed in the next process and protect the bottleneck station with a buffer and a more effective diversion of flow to lower variation in supply.

7 References

Automation World 2014. Tactical brief: The power of PackML. http://www.automationworld.com/e-book-how-get-started-packml-and-improve-control-engineering-efficiency. Available 5 June 2014.

Badiger, Anil S., Gandhinathan, R. 2008. A methodology to enhance equipment performance using the OEE measure. *European journal of industrial engineering, vol. 2, no. 3, pp. 356 - 376.*

Johansson, B. 2006. On virtual development of manufacturing systems – proposal for a modular discrete event simulation methodology. Göteborg: Chalmers

Law, Averill M. 2007. *Simulation modeling & analysis*. 4th edition. New York: McGraw-Hill

OMAC. 2014. *PackML*. http://omac.org/content/packml. Available 5 June 2014.

Robinson, S. and V. Bhatia. 1995. Secrets of successful simulation projects. In: *Proceedings of the 1995 Winter Simulation Conference*. C. Alexopoulos et al. (eds). pp 61-67, IEEE, Piscataway, New Jersey.

Skoogh, A., Johansson, B. 2008. A methodology for input data management in discrete event simulation projects. *Proceedings of the 2008 winter simulation conference,* 7 -10 December, 2008, Miami. pp. 1727 - 1735.

Skoogh, A. 2011. Automation of Input Data Management. Göteborg: Chalmers

Skoogh, A., Johansson, B. and Stahre, J. 2012. Automated input data management: evaluation of a concept for reduced time consumption in discrete event simulation. In: *SIMULATION*. November 2012:88. pp 1279-1293.

Appendix I: Specifications for filling machine module

I. Description of module

The filling machine module is built to represent the advanced filling machines at the food producing company. These machines are built up of three spinning disks, one smaller infeed disk, one bigger main disk and one smaller outfeed disk. The products are held in compartments in the disks and are transferred from one disk to another where the disks intersect. The actual filling is performed at the main disk without stopping the constant motion of the disks. To make the products enter the machine at the right speed and time to match with an empty comportment in the infeed disk the machine operates in companion with a infeed screw. To decouple the infeed screw from variance in product arrival rate a queue is held upstream the infeed screw. This queue is controlled by sensors that sends signals to start and stop the machine. In the same way sensors after the machine sends signals to stop the machine if it is blocked and start it again when cleared. The module can be seen in Figure I-1

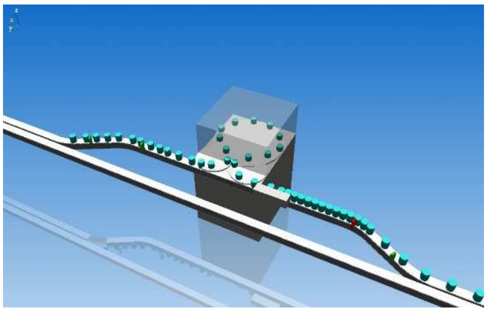

Figure I-1. Filling machine module.

II. Adjustable parameters

To be able to use the module for representation of several different machines it has been given a range of adjustable parameters. The parameters can be seen in Table I-1.

Table I-1. Parameters for a filling machine module.

Parameters	Value
Conveyor height	[mm]
Machine length	[mm]
Machine width	[mm]
Filler disc radius	[mm]
Infeed disc radius	[mm]
Outfeed disc radius	[mm]
Machine speed	[mm/s]
Mirror	[yes/no]
Use failures	[yes/no]

-MTBF	[distribution of [s]]
-MTTR	[distribution of [s]]
Use outfeed disturbances	[yes/no]
Use acceleration	[yes/no]
-Time 0-100 percent	[s]
Use deceleration	[yes/no]
-Time 100-0 percent	[s]

III. Layout building and simulation

For the module to integrate well with FLDT it has been given the functionality to snap directly onto an existing conveyor. To control starting and stopping of the module it needs to be connected to four sensors, one start sensor and one stop sensor in front of the module, and one start sensor and one stop sensor after the module. This is done through buttons implemented in the module. The distance between the start and stop sensors defines the start/stop ratio. The graphics of the module is done in a solid grey color for the base, solid white color for the discs, and a transparent top to visualize the flow of products in the machine. When the module goes down the top color changes to transparent red to indicate a failure. The module can be taken down manually during simulation.

IV. Output data

The module is operating in the PackML states and the current state is displayed during simulation. The output data is collected in terms of time spent in each state which together with a specified theoretical cycle time is used to calculate the OEE value. The OEE definition implemented in the module is the one suggested by Badiger *et al.* (2008) and values for OEE, availability, usability and performance efficiency is displayed during simulation. Scrap is not considered due to the case at the food producing company. The built in function in the software to generate charts over the state statistics can be used with the module.

Appendix II: Specifications for simpler machine module

I. Description of module

The simpler machine module is built to represent the most simple machines at the food producing company. These machines are built around a conveyor that goes through them. The conveyor is carrying the products through the machine and the machine is operating without taking the products off from the conveyor or changing the pitch between the products. Machines at the food producing company that can be represented by this module are the lid applicator, x-ray scanner, and the dishwasher. The module can be seen in Figure II-1 and Figure II-2.

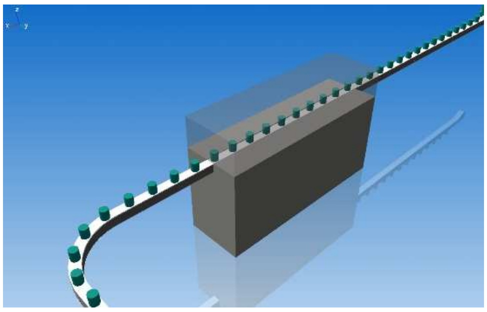

Figure II-1. Simpler machine set up to transport the products through the machine

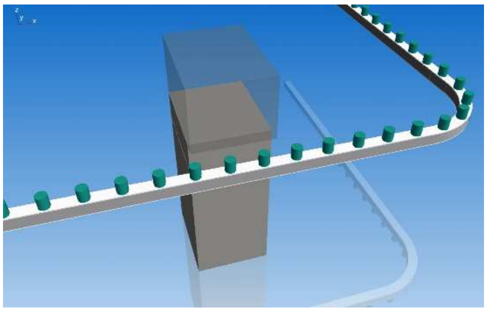

Figure II-2. Simpler machine set up on the side of the conveyor.

II. Adjustable parameters

To be able to use the module for representation of several different machines it has been given a range of adjustable parameters. The parameters can be seen in Table II-1.

Table II-1. Parameters for a simpler machine module.

Parameters	Value
Machine design	[standard/side mounted]
Conveyor height	[mm]
Machine length	[mm]
Machine width	[mm]
Use conveyor speed as machine speed	[yes/no]
-Machine speed	[mm/s]
Mirror	[yes/no]
Use failures	[yes/no]
-MTBF	[distribution of [s]]
-MTTR	[distribution of [s]]
Use outfeed disturbances	[yes/no]

III. Layout building and simulation

For the module to integrate well with FLDT it has been given the functionality to snap directly onto an existing conveyor. The graphics of the module is done in a solid grey color for the base and a transparent top to visualize the flow of products in the machine. When the module goes down the top color changes to transparent red to indicate a failure. The module can be taken down manually during simulation.

IV. Output data

The module is operating in the PackML states and the current state is displayed during simulation. The output data is collected in terms of time spent in each state which together with a specified theoretical cycle time is used to calculate the OEE value. The OEE definition implemented in the module is the one suggested by Badiger *et al.* (2008) and values for OEE, availability, usability and performance efficiency is displayed during simulation. Scrap is not considered due to the case at the food producing company. The built in function in the software to generate charts over the state statistics can be used with the module.

Appendix III: Specifications for simpler machine with feeder module

I. Description of module

The simpler machine with feeder module is built to represent the machines at the food producing company which have a conveyor going through them and operates with a infeed screw that creates a certain pitch between the products going in to the machine. To ensure a certain pitch between products in the machine a queue is held in front of the infeed screw to decouple it from variance in product arrival rate. This queue is controlled by sensors that sends signals to start and stop the machine. In the same way sensors after the machine sends signals to stop the machine if it is blocked and start it again when cleared. The products never leave the conveyor while inside the machine. Machines at the food producing company that can be represented by this module are the labelers, check weight, and some simple fillers. The module can be seen in Figure III-1 and Figure III-2.

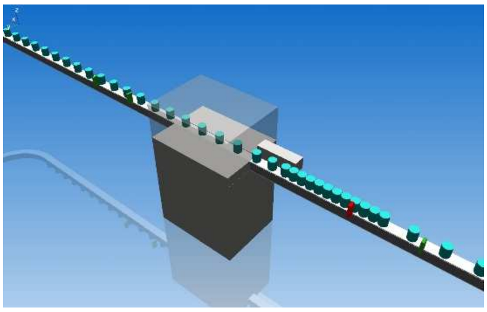

Figure III-1. Simpler machine with feeder module set up to transport the products through the machine

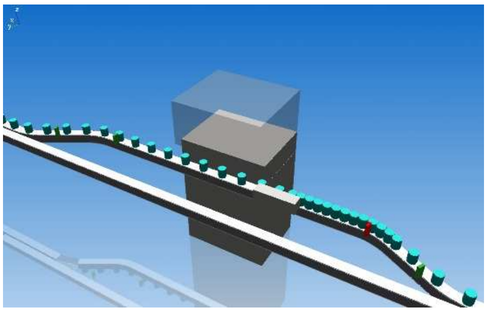

Figure III-2. Simpler machine with feeder module set up on the side of the conveyor.

II. Adjustable parameters

To be able to use the module for representation of several different machines it has been given a range of adjustable parameters. The parameters can be seen in Table III-1.

Table III-1. Parameters for a simpler machine with feeder module.

Parameters	Value
Machine design	[standard/side mounted]
Conveyor height	[mm]
Machine length	[mm]
Machine width	[mm]
Use conveyor speed as machine speed	[yes/no]
-Machine speed	[mm/s]
Mirror	[yes/no]
Use failures	[yes/no]
-MTBF	[distribution of [s]]
-MTTR	[distribution of [s]]
Use outfeed disturbances	[yes/no]

III. Layout building and simulation

For the module to integrate well with FLDT it has been given the functionality to snap directly onto an existing conveyor. To control starting and stopping of the module it needs to be connected to four sensors, one start sensor and one stop sensor in front of the module and one start sensor and one stop sensor after the module. This is done through buttons implemented in the module. The distance between the start and stop sensors defines the start/stop ratio. The graphics of the module is done in a solid grey color for the base and a transparent top to visualize the flow of products in the machine. When the module goes down the top color changes to transparent red to indicate a failure. The module can be taken down manually during simulation.

IV. Output data

The module is operating in the PackML states and the current state is displayed during simulation. The output data is collected in terms of time spent in each state which together with a specified theoretical cycle time is used to calculate the OEE value. The OEE definition implemented in the module is the one suggested by Badiger *et al.* (2008) and values for OEE, availability, usability and performance efficiency is displayed during simulation. Scrap is not considered due to the case at the food producing company. The built in function in the software to generate charts over the state statistics can be used with the module.

Appendix IV: Specifications for Diverter module

I. Description of module

The diverter is a component that diverts flow of products from one conveyor line to another. In the real world system, that is performed physically by an arm that guide the product from one conveyor to another. The diverter can be used both with manual operation and static operation. In the static case it is combined with a stopping function that stop the flow of products during a time to avoid collision between products and products or guide arm. A diverter operating in both of its states can be seen in Figure IV-1 and Figure IV-2.

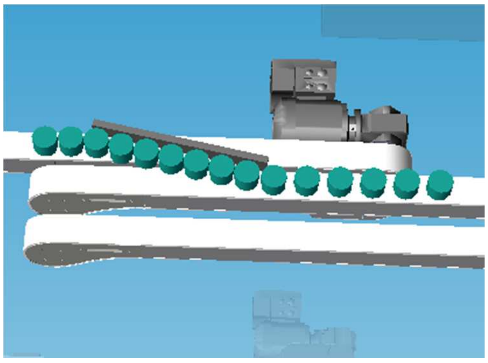

Figure IV-1. Diverter module diverting the flow to output 1.

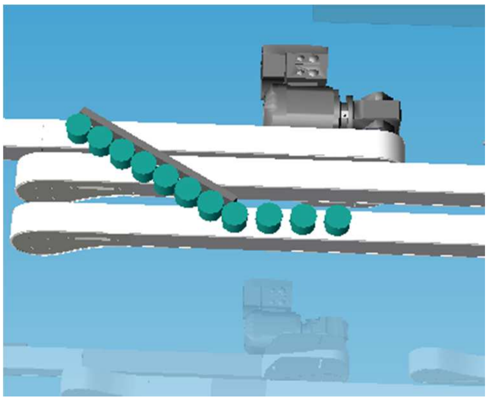

Figure IV-2. Diverter module diverting the flow to output 2.

II. Adjustable parameters

To be able to use the module for representation of several different machines it has been given a range of adjustable parameters. The parameters can be seen in Table IV-1.

Table IV-1. Parameters for diverter module.

Parameters	Value
Conveyor height	[mm]
Arm length	[mm]
Output 1 offset	[mm]
Output 2 offset	[mm]
Divert	[yes/no]

III. Layout building and simulation

To use the model together with FLDT, the conveyor needs to be broken up and connect to the diverter. The operation of the diverter is controlled by the Divert parameter. When the value is false the flow of product go to output 1, and when the value is true the flow of products go to output 2. To have static operation, the diverter has to be connected to a diverter controller. The diverter arm is visualized in a grey color, and the visualization shows where the flow is diverted.

IV. Output data

The diverter module has no output data.

Appendix V: Specifications for Stopper module

I. Description of module

The stopper module is built to represent the stoppers at the food producing company that stops the flow of products when diverting or merging flows. The module is built as one generic stopper that can be connected to any of the modules built in this project that needs stoppers in order to operate. The stopper is communicating with the connected module through Boolean signals. The module can be seen in Figure V-1 and Figure V-2.

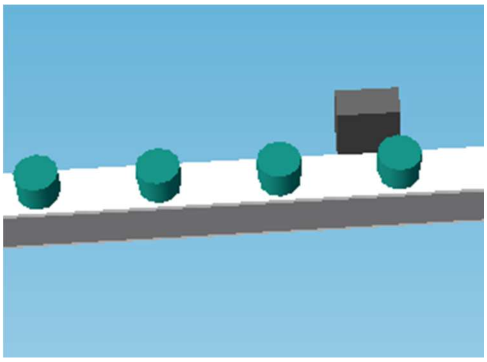

Figure V-1. Stopper module not stopping the flow.

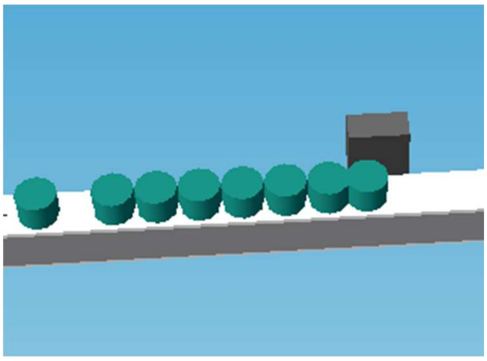

Figure V-2. Stopper module stopping the flow.

II. Adjustable parameters

The stopper module does not have any adjustable parameters

III. Layout building and simulation

For the module to integrate well with FLDT it has been given the functionality to snap directly onto an existing conveyor. Connecting the sensor module to another module is done through the other module. The graphics of the module has a solid green color when the sensor is cleared and a solid red color when the sensor is blocked.

IV. Output data

The stopper module gives no output data.

Appendix VI: Specifications for loose infeed screw module

I. Description of module

The infeed screw module is built to represent the functionality of the infeed screws at the food producing company. The module operates in exactly the same way as the infeed screws included in some of the machine modules but is built as a separate module. The module is intended to use in companion with a machine module without infeed screw for those occasions where the machine needs a certain pitch between the products but there is not enough space on the line to place the infeed screw in connection to the machine. To ensure a certain pitch between products leaving the infeed screw a queue is held in front of the infeed screw to decouple it from variance in product arrival rate. This queue is controlled by sensors that sends signals to start and stop the infeed screw. In the same way sensors after the machine sends signals to stop the infeed screw if the machine is blocked and start the infeed screw again when the machine is cleared. The module can be seen in Figure VI-1.

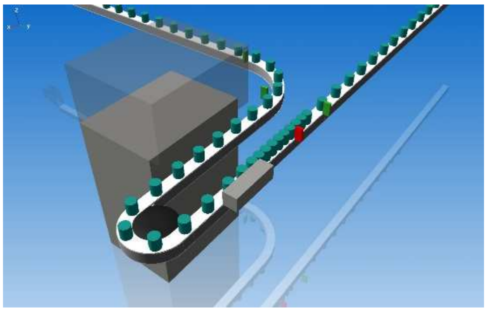

Figure VI-1. Loose infeed screw in operation.

II. Adjustable parameters

To be able to use the module for representation of infeed screws with slight differences it has been given a couple of adjustable parameters. The parameters can be seen in Table VI-1.

Table VI-1. Parameters for loose infeed screw module.

Parameters	Value
Conveyor height	[mm]
Use conveyor speed as machine speed	[yes/no]
-Machine speed	[mm/s]
Mirror	[yes/no]

III. Layout building and simulation

For the module to integrate well with FLDT it has been given the functionality to snap directly onto an existing conveyor. To control starting and stopping of the module it needs to be connected to four sensors, one start sensor and one stop sensor in front of the module and one start sensor and one stop sensor after the machine module. This is done through buttons implemented in the module. The distance between the start and stop sensors defines the start/stop ratio. The graphics of the module is done in a solid grey color and the flow of products going by the module is visualize.

IV. Output data

The module is operating in the PackML states and the current state is displayed during simulation. The output data is collected in terms of time spent in each state. The built in function in the software to generate charts over the state statistics can be used with the module.

Appendix VII: Specifications for twister module

I. Description of module

The twister module is built to represent the functionality of the twister at the food producing company. The twister consists of a stop unit and a spiral. The twister is not driven which is why it needs a queue upstream that pushes the products through the twister. This queue is controlled by sensors that sends signals to start and stop the twister. In the same way sensors after the twister sends signals to stop the twister if the twister is blocked and start it again when cleared. This is to avoid too high pressure on the products in the twister. The module can be seen in Figure VII-1.

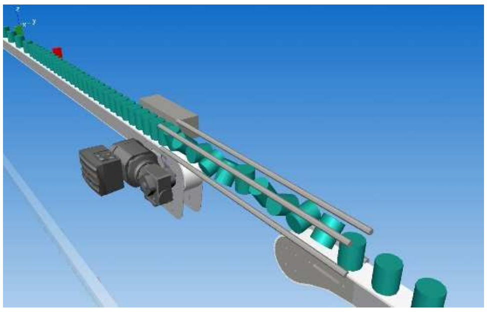

Figure VII-1. Twister module in operation.

II. Adjustable parameters

To be able to use the module for representation of more than one unique twister it has been given a couple of adjustable parameters. The parameters can be seen in Table VII-1.

Table VII-1. Parameters for twister module.

Parameters	Value
Conveyor height	[mm]
Twist length	[mm]
Use conveyor speed as machine speed	[yes/no]
-Machine speed	[mm/s]
Mirror	[yes/no]

III. Layout building and simulation

For the module to integrate well with FLDT it has been given the functionality to snap directly onto an existing conveyor. To control starting and stopping of the module it needs to be connected to four sensors, one start sensor and one stop sensor in front of the module and one start sensor and one stop sensor after the machine module. This is done through buttons implemented in the module. The distance between the start and stop sensors defines the start/stop ratio. The graphics of the module is done in a solid grey color and the flow of products going through the module is visualize.

IV. Output data

The module is operating in the PackML states and the current state is displayed during simulation. The output data is collected in terms of time spent in each state. The built in function in the software to generate charts over the state statistics can be used with the module.

Appendix VIII: Specifications for sensor module

V. Description of module

The sensor module is built to represent the sensors at the food producing company that detects products as they go by at the line. The module is built as one generic sensor that can be connected to any of the modules built in this project that needs sensors in order to operate. The sensor is communicating with the connected module through Boolean signals. The module can be seen in Figure VIII-1.

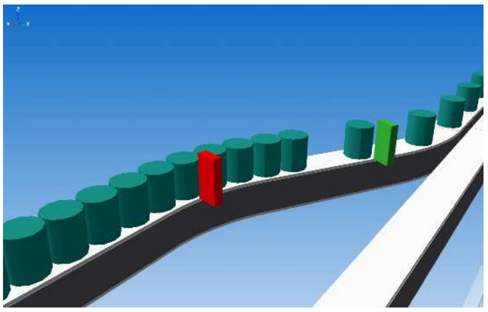

Figure VIII-1. Sensor module in operation.

VI. Adjustable parameters

The adjustability for the sensor module is limited to one parameter. This parameter is the threshold value which is a scaling factor used to decide when the sensor should signal that it is blocked. The sensor automatically calculates the time it takes for one product to pass the sensor at normal speed and by assigning a low or high threshold value to scale this time value the sensitivity of the sensor is determined. The threshold value cannot be lower than one.

VII. Layout building and simulation

For the module to integrate well with FLDT it has been given the functionality to snap directly onto an existing conveyor. Connecting the sensor module to another module is done through the other module. The graphics of the module has a solid green color when the sensor is cleared and a solid red color when the sensor is blocked.

VIII. Output data

The module gives no output data.