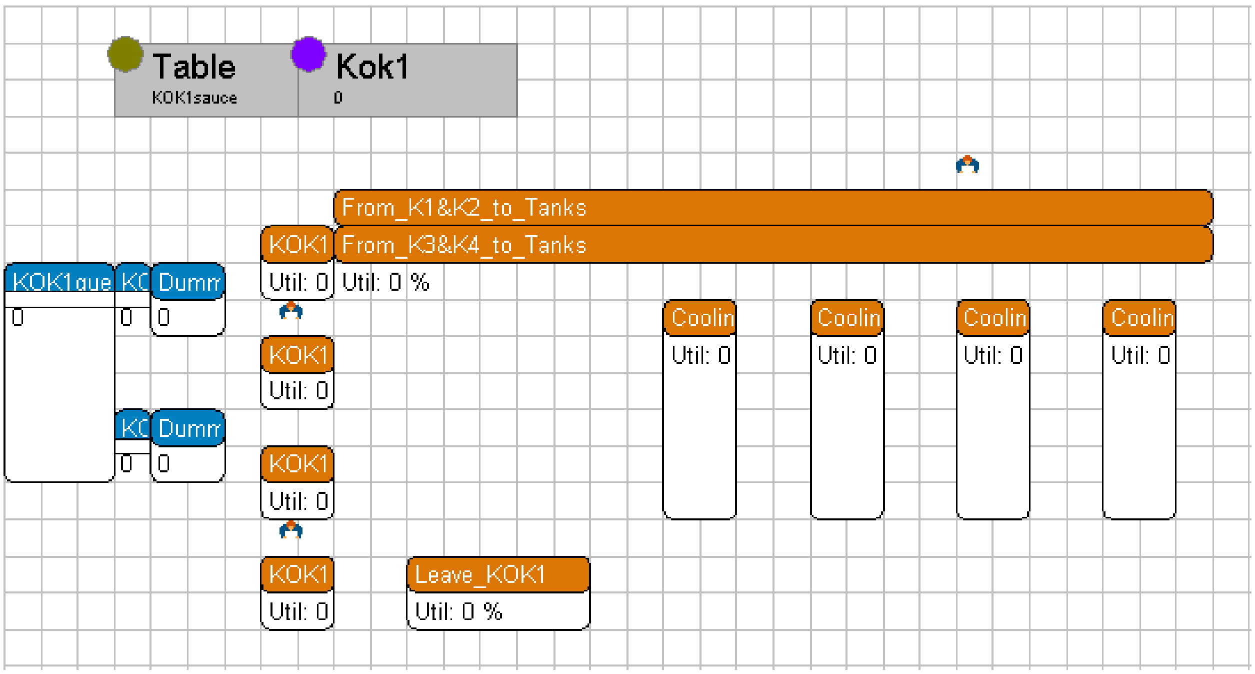


Simulation of a boiling station

Master of Science Thesis in the Master Degree Program, Product and production development

Martin Fredriksson

Department of Product and production development
Division of Production system
CHALMERS UNIVERSITY OF TECHNOLOGY
Göteborg, Sweden June 2010

Simulation of a boiling station

Master of Science Thesis in the Master Degree Program, Product and production development

Martin Fredriksson

Department of Product and production development
Division of Production system
CHALMERS UNIVERSITY OF TECHNOLOGY
Göteborg, Sweden June 2010

Editor of Simulation of a boiling station
Martin A. Fredriksson

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

Examinator: Björn Johansson

The cover is a picture of the simulation program Enterprice Dynamics displaying one of the boiling stations in this report.

Göteborg, Sweden June 2010

Thanks to

Nils Bengtsson from PMC for the help with the programing in the simulation program.

Henrik Kovács, Roland Berggren and Eva Roos from Findus for the help with the organisation and direction during the whole project.

Annette Lucas from Findus for the help with sorting the articles.

Richard Larsson and Daniel Rydberg fom Findus for the help with explaining the project to the employees and testing the simulation program.

Björn Johansson from Chalmers for the direction and for the help with starting this project.

Abstract

Findus is a food company and it has a factory that produces finished dishes. There are three boiling stations that boil different kinds of sauces. This report explains how the boiling stations work and how to implement this to a simulation.

To create a simulation, data is needed from the factory. This data was collected with three different methods. The model was built in the software ED (Enterprise Dynamics 7 Studio) and has Excel as a user interface.

The sauces were divided into different data groups. The data of the groups was run in the software Expertfit to get different distributions on the tasks.

When the simulation is done, ED prints the results to an Excel sheet. Excel calculates the time from the start to the end of the sauce batch and presents this in two ways.

Contents

Objectives

When this project started some meetings about the objectives where held at Findus(1). At the meetings there were discussions about how a simulation program can help Findus and where this simulation can have the highest influence. Some suggestions were developed and the alternative with the boiling station was the winning solution. The objective from the start of the project was to find the bottleneck of the boiling stations and the goal was to give Findus suggestions on investments that could improve the boiling stations. The objective of this project was changed over time because Findus guided the project in a different direction.

The new objective became to show the company how a simulation tool can help them to plan their production in a better way. The new goal is that this project should lead to a program that the employees who plan the production on Findus can use to prevent the boiling stations from becoming a bottle neck in the production flow. To reach this goal the simulation tool has to be simple and should be easy to use. A way of fulfilling these conditions could be to use a program that is well known on Findus to interact with the simulation program.

Introduction

This part introduces Findus and displays the limitations for this project. The last part is an introduction to show how the boiling station works.

Background

Findus is a food company that works with the entire chain from growing vegetables to distributing the completed food to companies that sell the food to costumers for example ICA and Coop. The central office of Findus in Sweden is located in Bjuv, where the main part of this project is carried out.

In Bjuv Findus has three different plants and those plants have different production lines. Three of the production lines are called finished dishes lines and they produce different kinds of food in one-serving boxes, it can be beef or fish together with mashed potatoes and vegetables served with various sauces. There are two fish lines, one line produces the fish dishes and the other produce some of the fish that is used by the finished dishes lines. There is also one meat line that produces all kinds of meat and some of the meat products are sub products to the finished dishes lines.

The finished dishes lines and the fish lines have different products that are boiled in the three boiling stations. This simulation is used to provide the planners with a tool to check if the planned production of the next coming week could be feasible.

Findus introduced the lean production (2) one and a half years ago and this could be a good starting point on this project, but have the lean production reached the boiling stations? In some way it has by introduction of the method 5S and group meetings, but they hadn’t collected data from any boiling station which is what was needed for this project.

Limitations

Due to the time and the massive amount of articles the following assumptions are made;

- The lines are always going at 100%.
- The boilers have infinite supply of ingredients, but a time for collecting them is added.

Boling stations

In the factory there are three boiling stations where Findus produces sauces. The first is the KVK station which is the oldest one, that station has three 450 liter kettles. The kettles in KVK are used to produce sauces that don´t need to boil like mashed potatoes and sauces that don’t need to boil very long like spinach. The station has 18 different types of sauces and some of the products need to cool down when they have been boiled, this is done by putting a cap over the kettle and use vacuum to cool the product. The station serves both the fish lines and the finished dishes lines.

The second is the KOK1 station, this station produces sauces to different products on the finished dishes lines. In KOK1 there are 40 different sauces belonging to 85 different dishes. The station has four 750 liter kettles and four tanks that are used to cool down the sauces.

After a kettle of sauce have been boiled the cleaning of the kettle starts, if the next sauce is not the same the employees use chemicals to clean the kettle but if it is the same sauce they clean the kettle with a water hose.

The third station is the KOK2 and this station has two sides. One where they boil sauces to the fish lines and one where they boil pasta and fry couscous for the finished dishes lines. On the fish side the station has two kettles to boil the sauces to one of the fish lines which are connected to a big cooling tank. When the sauce has cooled off the sauce is pumped to a bigger tank and from this tank there are pipes in the ceiling to the fish line. When they sometimes make a warm sauce they connect the pipes in a different way directly to the big tank. Beside those two kettles there are two kettles that can be tipped which are used for both fish lines. On the other side (the pasta side) there are five kettles and those are used for boiling pasta and fry couscous. Now there is a new project that uses one of the five kettles to melt butter for the new mashed potatoes. This is not included in the simulation model because it is temporal and in the future there will either be a new procedure on how to make the mashed potatoes or it will be the same procedure as before.

There is one more station where the boilers combine butter with three different kinds of flavors. The time-consuming part in this procedure is not the time mixing the butter with the flavors it's the transportation time for getting the butter and leaving the butter with flavors. A picture of the simulation model in ED (Enterprise Dynamics) (3) is displayed in Figure 1.

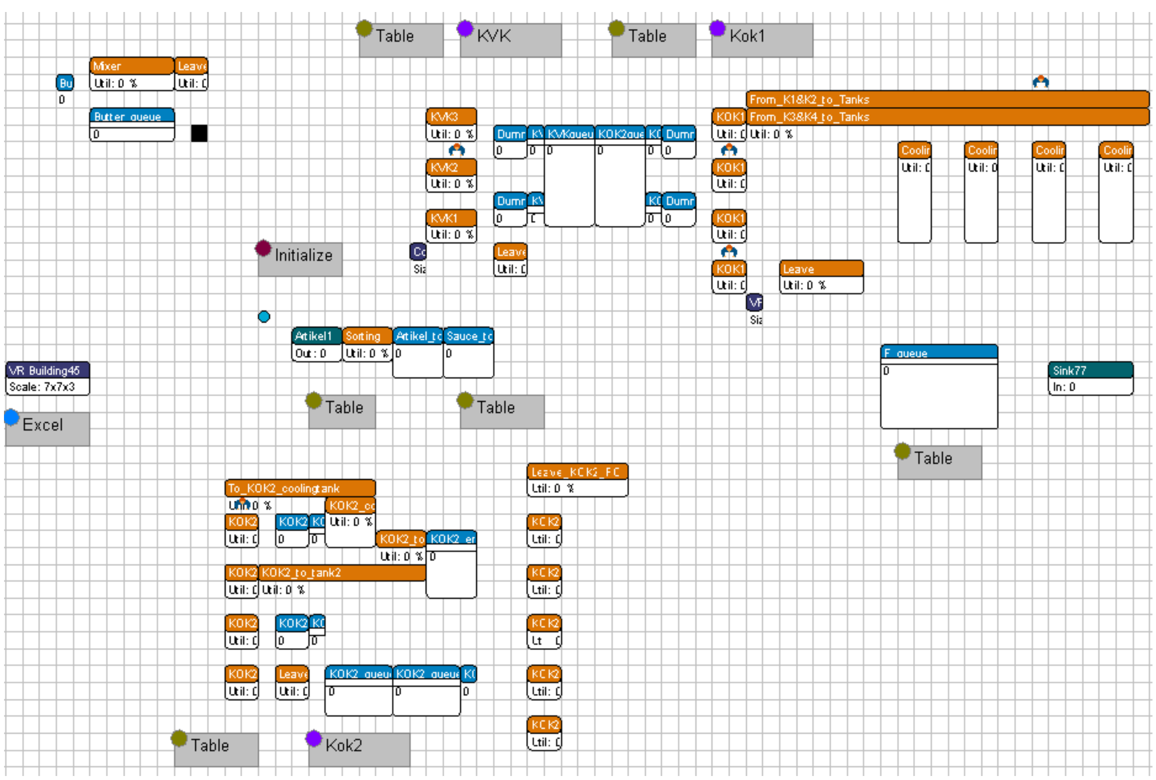


Figure 1, the simulation model

Flow analysis

A flow analysis was created with help of interviews with the employees on the different boiling stations. The flow in Figure 2 was created, this flow describes the route that every sauce takes through the boiling stations.

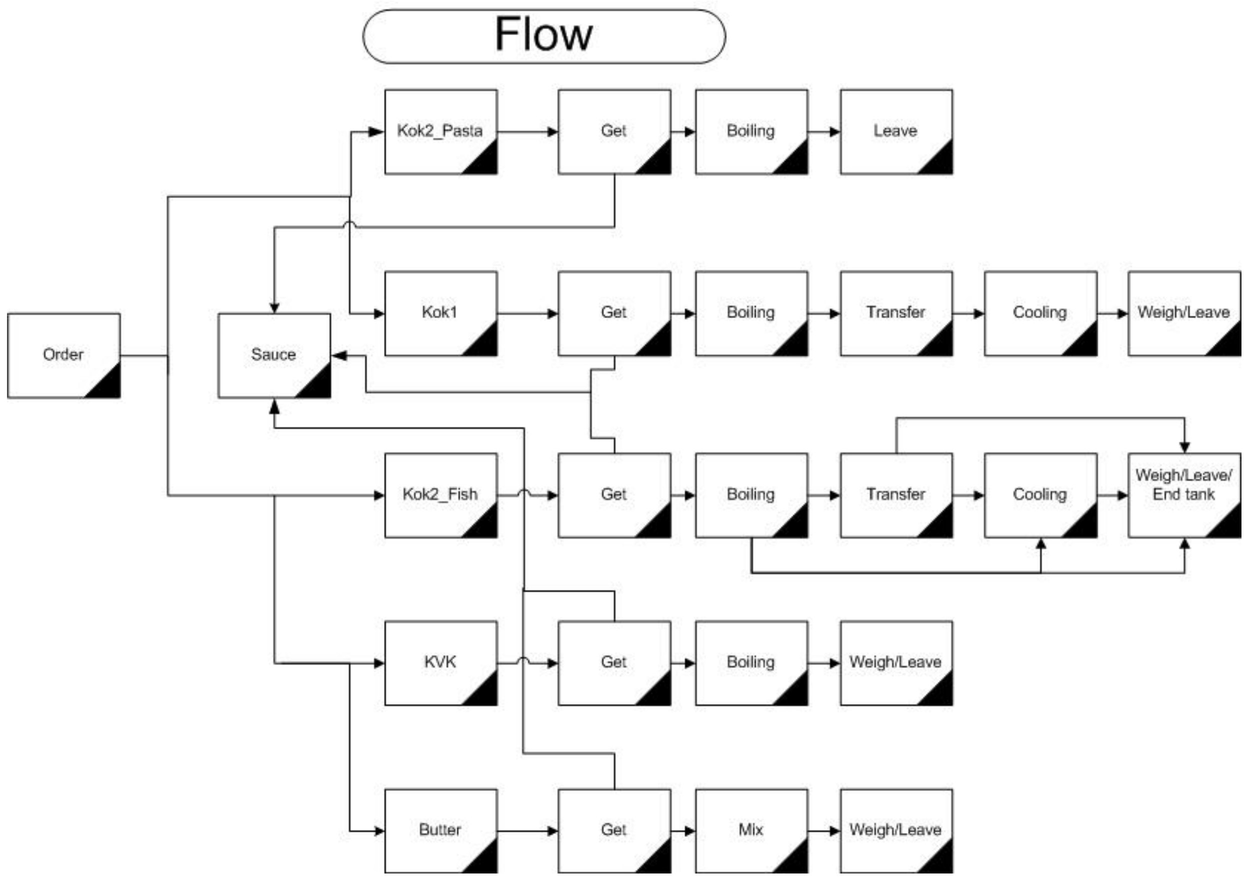


Figure 2, the flow

When an order comes to Findus the order is split into small suborders that every station gets. Then they have to get the ingredients for the sauces, usually the ingredients are already sorted and weighed to the sauces. When they have checked that they have all the ingredients they start to boil the sauce or mix the butter, then the task depends on which station the sauce is in and what kind of sauce it is. For example if it is a hot cheese sauce on the Kok2_Fish station the sauce should be boiled and then transferred directly to the end tank.

Methods

Three different methods to collect data where tested and one method was used for the programming.

Interviews

This method uses questions about the employee's work tasks and defining the max and min time on those. This method was very easy to use on the KVK station because the station does not have too many different sauces. To use this method on the Kok1 station would be very difficult because there are so many different sauces. The drawback of this method is that the times are not following a distribution that represents the real production and the programmer must know much about the boiling procedure to choose a distribution for it. To check if the times received from the interviews were in the range of the real production times the second method was used.

Stopwatch

This method uses a stopwatch to collect times. The time is started when an employee starts a new task and is registered when he finishes. These tasks on KVK could be reduced to only three:

- Time to get the ingredients and clean the kettle
- The time for the sauce in the kettle
- Time to leave the product in a refrigerator

The method was very easy to implement with these tasks, but when the number of tasks start to increase the method could be very confusing and it was hard to keep track of which time should start and which was finished. This is the case in the Kok1 station where the tasks are:

- Time to get the ingredients and clean the kettle
- Time to boil
- Time to transfer to cool tank
- Time to cool
- Time to tap, weigh and leave the sauce

With those tasks, four boiling stations, four cooling tanks and the fact that the amount of collected times was five on an eight our shift it would take 1.5 years to collect five samples of each sauce. The solution was the third method.

Form

This method uses a form to collect times. This form should be handed out each day and the question is about starting and stopping time for each task. The article and sauce type should also be written down. This was implemented in a structured way by introducing the form to the boilers in groups. The form is displayed in Appendix A.

To save time for the employees they only had to write down the time a task started and the time it ended, they didn't have to calculate the amount of minutes the tasks took. The forms were handed in every day by putting them in a binder. From the times on the forms the time for every task was calculated and put in an Excel (4) sheet.

The next station was the KOK2 station where the tasks were more spread out than in the KOK1 station. The tasks are:

- Time to get the ingredients and clean the kettle
- Time to boil
- Time to transfer to cool tank
- Time to transfer from kettle to end tank
- Time to cool in the tank
- Time to cool in the kettle
- Time to transfer from the cooling tank to the end tank
- Time to tap, weigh and leave the sauce

Some tasks on this station where very hard to collect sufficiently many times at so the amount of data was not enough to create time distributions. One of them was the time from the cooling tank to the end tank and instead of a distribution a constant time was inserted.

Programming method

There are three steps in the programming (5) (6):

- The first step is programming the model, this is done with the program Enterprise Dynamics (ED) and the knowledge that has been collected.
- The second step is to validate the simulation model. This is accomplished by running the simulation with different inputs and check if the output is in the range of a feasible solution.
- The third is verification and this is done by comparing the result from the simulation with the results from the real production.

Data Analyses

The data was first collected and put in a table for each station, thereafter the data was sorted by the sauce name. Then the min, max and average times of the different tasks where calculated in Excel, because there was not enough data collected to create a distribution for each sauce. That's why the sauces must be divided into groups.

KVK Data

Only one of the tasks on KVK could be used to create a distribution, namely the time a sauce spends in the kettle. There were totally 150 times collected and 60 % of them were for mashed potatoes. There were 18 sauces which belongs to this station in total, but the collected times covered only 10 of them. This was because some of the sauces were unusual and some have been erased from the production. There were three different groups in the data and the groups can be seen in Table 1.

Table 1, sauces KVK

Sauce type	Group
Mos152433	1
Mos152437	1
Rödkål	2
Lingon	2
Salsa	3
Chili con carne	3
Tacofyllning	3
Spenat	3
Primatopping	3
HollandaiseTop	3

KOK1 Data

There are four tasks on this station boiling, transferring to cooling tank, cooling and the leaving procedure. There were 350 times collected and they were distributed over 40 sauces, but as the same reasons as in the KVK station there were only 34 sauces that had been boiled. The sauces that were not boiled are sorted into a group where the viscosity of the sauce is nearly the same as a sauce that was registered. The groups are listed in Table 2.

Table 2, groups of the tasks in the KOK1 station

Sauce Type	Boiling Groups	Sauce Type	Transfer Groups	Sauce Type	Cooling Groups	Sauce Type	Leaving Groups
Kryddsås	1	Kryddsås	1	Kryddsås	1	Kryddsås	1
Rödvinssås	1	Rödvinssås	1	Rödvinssås	1	Rödvinssås	1
Alfedosås	2	Tomat Lök	2	Ost/Skinksås	2	Bechamelsås	2
Gurksås	2	Örtagårdsås	2	Cajunggryta	2	Gurksås	2
Kycklningsås	3	Bechamelsås	2	Stuv Maksås	2	Grillsås	2
Youghurtsås	3	Jerksås	2	Tikka Masalasås	2	Kycklningsås	2
Gratängsås	3	Gratängsås	2	Ostsås	2	Tikka Masalasås	2
Ostsås	3	Gurksås	2	Chorizosås	2	Gratängsås	2
Ost/Skinksås	3	Tikka Masalasås	2	Remouladesås	2	Moulin Rouge sås	2
Skysås	3	Remouladesås	2	Grillsås	2	Currysås	2
Örtagårdsås	3	Youghurtsås	2	Gurksås	2	Örtagårdsås	2
Vattenkrassesås	3	Alfedosås	2	Alfedosås	2	Gräddsås	2
Gräddsås	3	Indonesiansås	2	Gratängsås	2	Portersås	2
Portersås	3	Bacon/purjolöksås	3	Champinjonsås	3	Youghurtsås	3
Currysås	4	Kycklningsås	3	Ajvar Relish	3	Champinjonsås	3
Grillsås	4	Skysås	3	Örtagårdsås	3	Indonesiansås	3
Champinjonsås	4	Moulin Rouge sås	3	Kycklningsås	3	Cajunggryta	3
Remouladesås	4	Gräddsås	3	Indonesiansås	3	Skysås	3
Bacon/purjolöksås	4	Köttbullsås	3	Moulin Rouge sås	3	Bacon/purjolöksås	3
Bearnaisesås	4	Portersås	3	Skysås	3	Ostsås	3
Indonesiansås	4	Vattenkrassesås	3	Bacon/purjolöksås	3	Bearnaisesås	3
Paprikasås	4	Currysås	3	Portersås	3	Stuv Maksås	3
Stroganoffsås	4	Stroganoffsås	3	Vattenkrassesås	4	Jerksås	3
Chorizosås	4	Stuv Maksås	3	Stroganoffsås	4	Paprikasås	3
Moulin Rouge sås	4	Ostsås	3	Currysås	4	Ost/Skinksås	4
Stuv Maksås	4	Champinjonsås	3	Gräddsås	4	Ajvar Relish	4
Tikka Masalasås	5	Grillsås	3	Köttbullsås	4	Hollandaisesås	4
Köttbullsås	5	Ost/Skinksås	3	Bearnaisesås	4	Stroganoffsås	4
Ajvar Relish	5	Hollandaisesås	4	Youghurtsås	4	Köttbullsås	4
Hollandaisesås	5	Ajvar Relish	4	Hollandaisesås	4	Tomat Lök	4
Cajunggryta	5	Cajunggryta	4	Tomat Lök	4	Alfedosås	4
Tomat Lök	5	Chorizosås	4	Bechamelsås	4	Remouladesås	4
Jerksås	5	Paprikasås	4	Jerksås	4	Chorizosås	4
Bechamelsås	5	Bearnaisesås	4	Paprikasås	4	Vattenkrassesås	4

KOK2 Data

There are 15 sauces that should be boiled on this station, but there have only been seven registered and there are only 70 times of them collected. The distributions were made for those seven sauces and with some analyses the other sauces were sorted as well, but to make a simulation model that is more reliable, more time samples should be needed on this station. The groups of this station are listed in the Table 3.

Table 3, groups of the tasks in the KOK2Fish station

Sauce type	Boiling Groups	Sauce type	Transfer Groups	Typ Cooling	Cooling Groups
Mbordelaise	1	Citron/dillsås	1	Citron/dillsås	1
Mmandel	2	Dillsås	1	Dillsås	1
Ostmassa	2	Vostsås	2	Vvin/räksås	1
Citron/dillsås	3	Vvin/räksås	1		
Dillsås	3				
Vostsås	3				
Vvin/räksås	3				

The data from the different groups were later used as an input to the software ExpertFit(7). You can find the results from this program in the section ExpertFit results.

Programming

This section starts in the same order as the real programming by connecting to Excel. The section ends with explaining how the results are programmed. An example of the code is found in Appendix B.

Connect Excel to ED

To feed the model with input the software Excel was chosen. Since Excel is used for other purposes in Findus the employees already know how it works which makes it easier for them to use the simulation program. The main idea was to give the simulation program the articles and the amount of the finished product in kilograms that should be produced the coming day. To convert the finished product in kilograms into number of kettles that should be produced, Findus uses a list of variables which tell us how many kilograms of finished product there are per kettle.

In ED there is only one entry of products and to choose how many products that should enter the program multiplies the number of sauces on the article with the number of kettles that should be boiled. One kettle of sauce is one product in the ED simulation program.

When the production volume is for example 1.5 kettles the planners on Findus use two 75% full kettles instead of one full and one half. In this simulation the time of 75% full kettles is included in the time distributions of the boiling stations. The ratio between the finished product in kilograms and the amount of finished product per kettle is always rounded upwards to get the right amount of kettles.

After you enter the article and the amount of finished product Excel calculates the number of kettles, how many sauces and which row the article has in the list of all articles. When the Simulation program starts or resets, a program code runs in ED which resets all counters, pointers and some tables. When the ED simulation begins, a pointer is pointing at the first row in the Excel Sheet and the first product gets the labels “Article”,” Sauces” and “Row”. When there is a new article the pointer moves to the row under.

The total amount of kettles is also calculated in Excel and put in a global variable in ED. this variable is tested against the amount of kettles that are leaving the simulation program. When they are equal the simulation run stops.

Sort the sauces

A table “Sauces” was added to the simulation model to divide the articles into their different sauces. The table has four columns, the first tells us the article number and the next three tells us which sauces that article has. Each kettle of sauce that should be boiled receives a label “Sauce”. This table points out what kind of sauce the product has.

In the next part of the flow the products are given a label “Station”. To be able to make this functional, a table named “Boiling stations” is added. This table has two columns; sauce and station. The simulation program checks which row the sauce has in the table and puts the value in the right column to the label “Station”. This label is later used to send the sauce to the correct station.

KVK

The starting of the programming on the KVK was done after some interviews with the boilers that work on this station. Some important observations was made, one was that kettle three only produces mashed potatoes. This was because the cooling system of that kettle does not function very well and the other sauces needs to boil and then cool, the mashed potatoes only need to be mixed. To include this in the programming, two dummy queues were introduced. One that only gets and handles the products with mashed potatoes. The other dummy queue gets and handles the distribution of the other sauces.

The dummy queues also have two other important functions. One is to check what sauce the product has and then sets a cycle time for the kettle by finding the sauce in the table with the time distributions. The second is to check if the sauce is different from the sauce in the last product. If it is, an extra time is added to the setup time because of the extended cleaning time. The distributions are located in a table called KVKsauce. A leave process represents the time for the weighing and leaving the wagons in the refrigerator, see Figure 3.

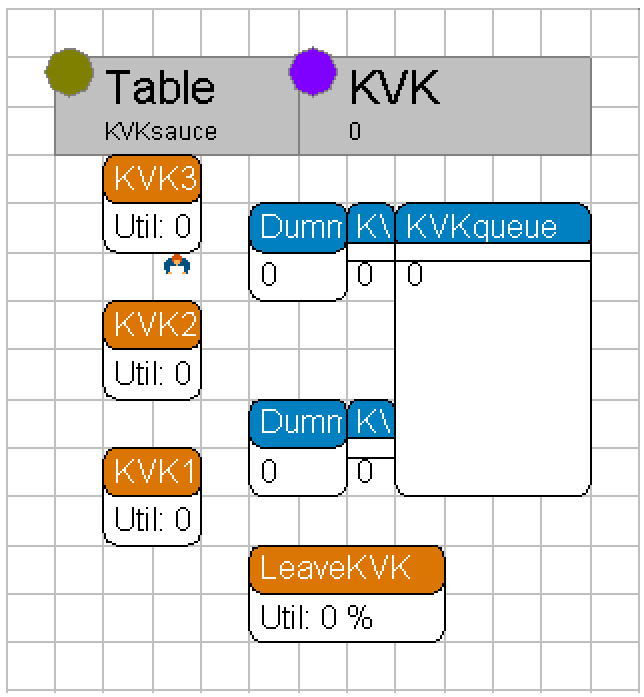


Figure 3, the KVK station

KOK1

When programming the KOK1 station there was much to think of, one obvious thing was that the station must have more control over how the sauces should move. To start with, the station got two dummy queues like KVK.

To imitate the real procedure the first queue is used to split the sauces up into two separate dummy queues and give every product a tank number where the sauce should cool down, if the sauce is the same as the one before, the tank number will be the same.

The dummy queues has many tasks, the first is to check if the sauce is the same as the one before, if the sauce is the same, the dummy queue should give the sauce a label “Kettle” and then shift a global variable to the next kettle. The kettle is given a cycle time and a setup time to clean and get more ingredients. This setup time is based on the cleaning time between the same sauces.

If it is not the same sauce the dummy queue sets the label “Kettle” to the same as the one before and then closes the output for the dummy queue. The setup time in this case is based on the time to clean between two different sauces. There are four kettles in this station and they work in pairs. They are connected by a pipe system to the cooling tanks. When the sauce has been boiled, a code checks which sauce that have been boiled and sets the cycle time for the transportation of that specific sauce. When the transportation is finished the product is sent to the tank that the product has on its tank number label.

The products in the two kettles have to be done and transferred to the cooling tank before the new sauce can enter the kettles. This lock system is included in both the dummy queues and the transfer to cooling tanks process. One other thing in the dummy queues is that a label “Row number” is added to the product, the code checks where the sauce is in a table where the distributions for the cycle times and setup times for the stations different moments are located.

When a product enters a cooling tank the setup time is given to the tank if it is a new sauce. Then there is one phase left and that is to tap and weigh the sauce and put it in the refrigerator, the cycle time for that is given when a sauce leaves one of the cooling tanks, see Figure 4.

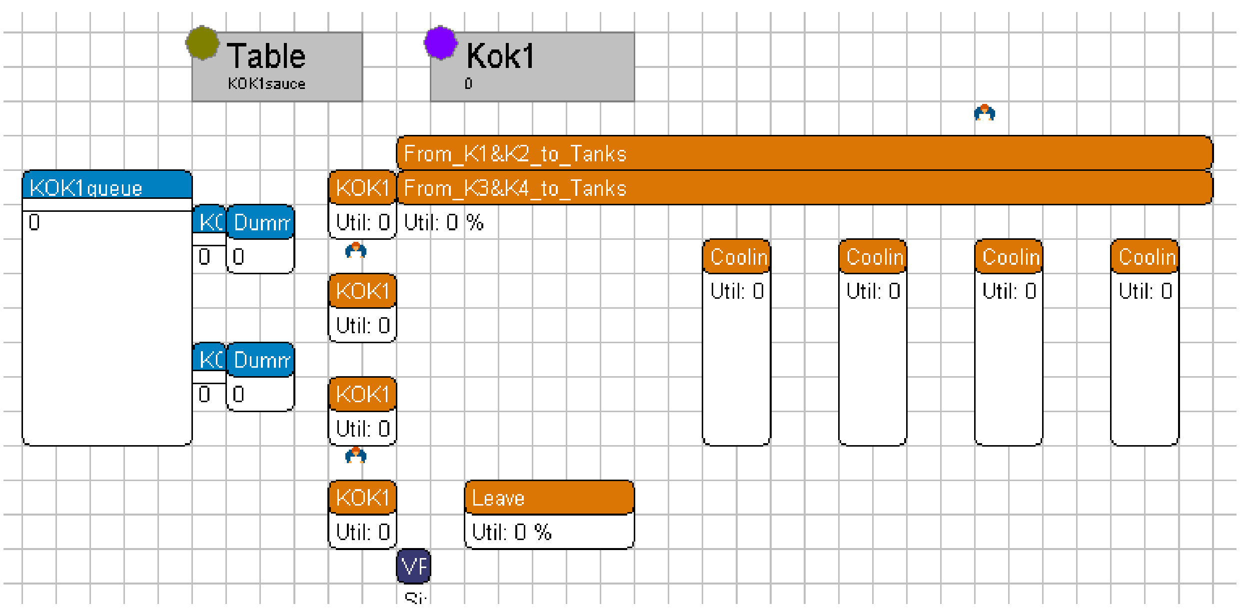


Figure 4, KOK1 station

KOK2

The programming in the KOK2 station was almost the same as in the KOK1 station. The main idea was to separate the sauces into two different dummy queues, one that uses the two kettles that can tip and the other one that uses the kettles that are connected by pipes to the cooling tank or to the end tank, see Figure 5.

The two dummy queues are nearly the same as in the programming, they check if the sauce that comes in to the dummy queue is different from the one that was there before. If the sauce is different, the cleaning time should increase and in the case with the kettles connected with pipes to the cooling tank, the dummy queue closes its output. Then it waits for the sauces in the kettles to be pumped over to the cooling tank. When the sauce is pumped to the cooling tank the dummy queue opens the output again.

The time for the boiling and cleaning is given to the kettle when the sauce product leaves this dummy queue. When the product enters the dummy queue, labels are added to the product to set the right time for the boiling and cleaning.

The most important of the labels is “Row_kok2”. When the sauce is found in the KOK2 sauce table, the label “Row_kok2” is given a row number. This is used to find the time for the boiling, transferring and cooling for the sauce by combining the label “Row_kok2” with a column in the KOK2 sauce table.

The two kettles which can be tipped over gets a boiling time and a cleaning time if it’s a new sauce, otherwise only a boiling time. When the sauce is done they use water to cool the sauce. Instead of using two stations for tow tasks, there is one station for boiling and cooling. The times for those are added together in the boiling time. After this time the sauce is put in a leaving station that illustrates the tipping of the sauces into wagons, registering of their weights and brings them to a refrigerator.

The two other kettles are different in the sense that they have two outputs and this is because they sometimes make warm sauces that should be pumped directly to the end tank. This program code tests if the label “Sauce” is the sauce that should go directly to the end tank. If the sauce is going directly to the end tank, the label “Kettle” is set to two and otherwise this label is set to one. The kettles are checking the label “Kettle” and will send the sauce trough this output.

The transfer time for the warm sauces from the kettles to the end tank is given directly to the transfer2 station. If the sauce goes to the cooling tank instead, the transfer times is located in the KOK2 sauce table and are given to the transfer1 station before the sauce leaves the kettle.

The cooling tank gets its cycle time when the sauce leaves the “To_KOK2_coolingtank” station. When the sauce has entered the cooling tank, the output on the kettles that are connected by pipes are closed. This is because the sauce needs to cool down and leave the cooling tank before a new warm sauce will be able to enter. A cleaning time is added if the sauces are different. There is a fixed transfer time from the cooling tank to the end tank and when the sauce has left this transfer station, the outputs in the kettles are reopened.

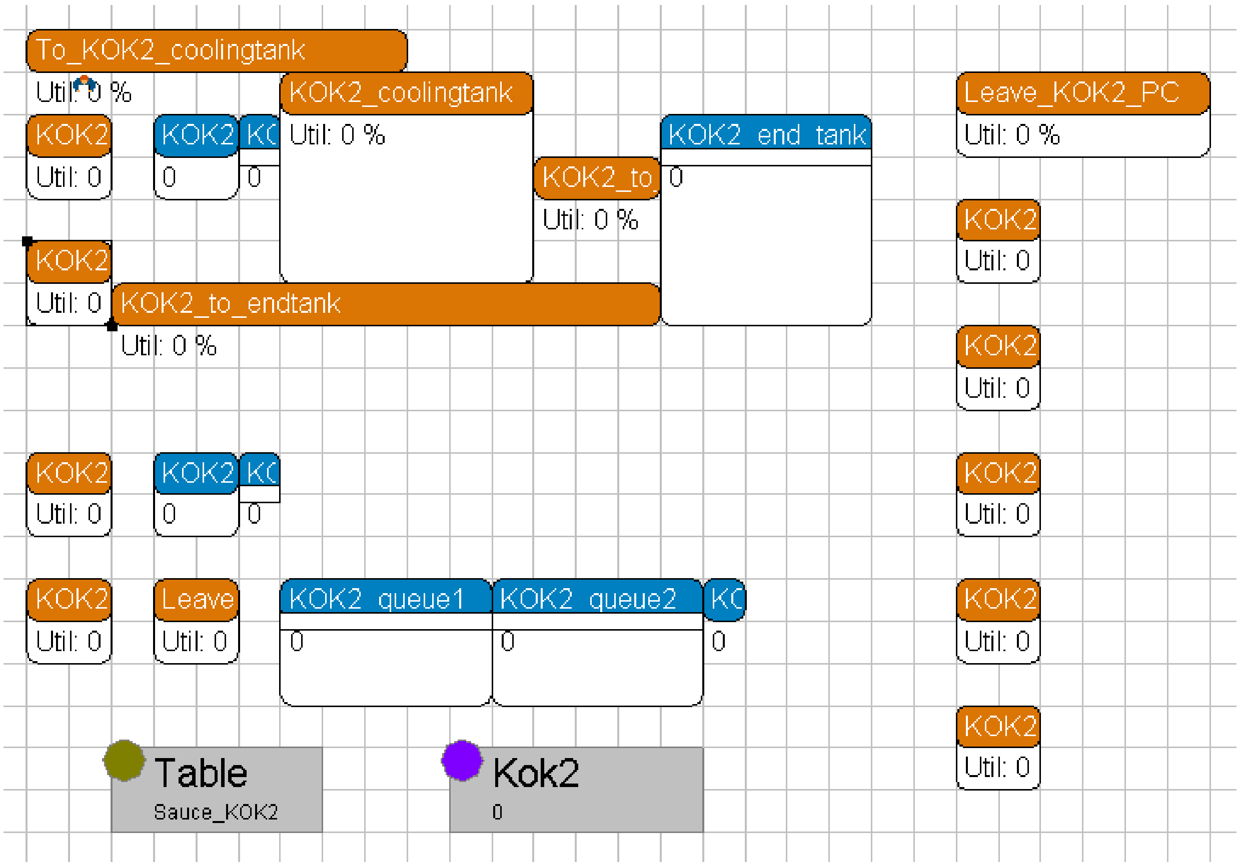


Figure 5, KOK2 station

Results

To get the time from the start of a sauce batch to the last sauce in that batch, some time clocks were added in all dummy queues. When a new sauce leaves the dummy queue to a kettle a time stamp, the sauces name and the article are printed out in a table named “Data”. This dummy queue also puts a stamp on the last sauce in the batch. When a sauce leaves the last station the program inserts the end time for that sauce batch into the “Data” table. When the simulation stops the “Data” table is printed out to Excel.

Validation

By testing the simulation model with different inputs, some bugs were found. One that was found with this method was that when a sauce came in to the dummy queue the sauce executed a code that set the cycle time on the kettle. If this was a new sauce the sauce in the kettle got the wrong boiling time. The solution was to put the code that sets the setup time and cycle time when the sauce leaves the dummy queue instead.

The second problem that was found was that if the product had more than four kettles to boil the sauces became a plug in the main queue. By adding a queue between the main queue and the dummy queue the problem was solved and there can now be 23 kettles of the same sauce before that problem appears again.

After the simulation model were updated with the mathematical distributions of the different sauce groups the model was tested by the employees to check if the times where valid. The test was done by letting the employees run the simulation and check if the simulated times could be the same as the real times in the boiling stations. The employees couldn’t tell that the simulated times were longer or shorter than the real times.

So, how close was the model to the real boiling stations? This question is not answered yet but the model is under a testing phase and one idea to check how close the model is was added in one of the presentations on Findus. This idea is to give out the forms over a production week and then run the simulation parallel with the real production. Then the result from the forms and the simulation runs can be compared to find out how close the model is.

Results and manual

ExpertFit results

The times that were measured and divided into different sauce groups were inserted to ExpertFit and the output can be seen in Table 4.

Table 4, the mathematical distributions of different tasks

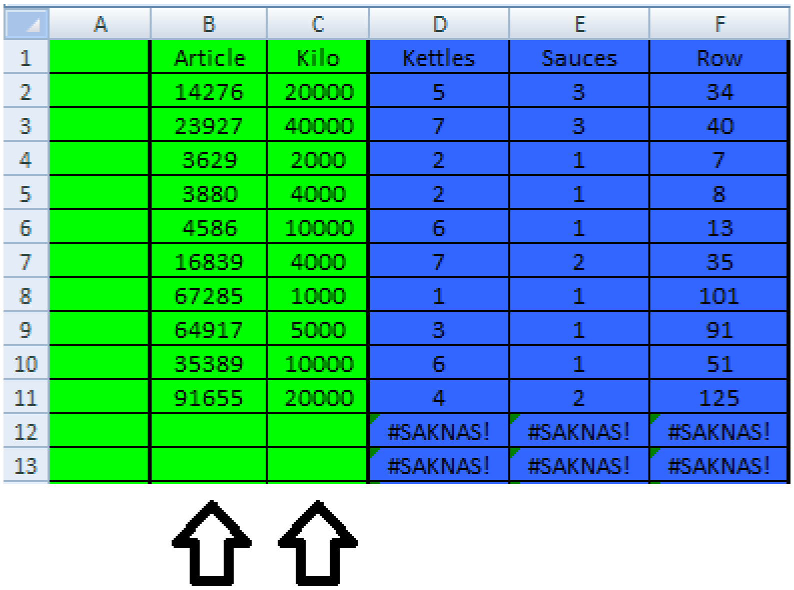
KVK	
KVKTime1	1.0/weibull(0.052853, 3.671839, <stream>)
KVKTime2	33.964286 + weibull(7.426219, 1.041330, <stream>)
KVKTime3	52.783784 + weibull(31.908247, 1.379899, <stream>)
KOK1	
Boiling1	3.738200 + beta(7.846416, 4.164532, 5.050384, <stream>)
Boiling2	pearsont5(19.908978, 18.468528, <stream>)
Boiling3	pearsont5(25.853665, 9.341688, <stream>)
Boiling4	pearsont5(33.185960, 10.584382, <stream>)
Boiling5	19.921569 + weibull(22.418773, 1.182535, <stream>)
Transfer1	0.500000 + weibull(2.274828, 2.490831, <stream>)
Transfer2	gamma(5.475610, 12.000000, <stream>)
Transfer3	2.909091 + gamma(3.818850, 3.489073, <stream>)
Transfer4	pearsont6(10.360253, 78.974260, 8.622812, <stream>)
Cooling1	22.324696 * exp(ln(1.0/random(1.0, <stream>) - 1.0)/6.292821)
Cooling2	11.781705 + beta(26.241023, 2.923095, 3.389243, <stream>)
Cooling3	39.422765 * exp(ln(1.0/random(1.0, <stream>) - 1.0)/8.410955)
Cooling4	pearsont5(43.834019, 22.102800, <stream>)
Cooling5	pearsont5(53.689374, 12.914450, <stream>)
Leave1	1.666667 + gamma(2.250000, 1.000000, <stream>)
Leave2	3.900000 + gamma(3.643210, 2.509928, <stream>)
Leave3	8.815634 * exp(ln(1.0/random(1.0, <stream>) - 1.0)/6.199070)
Leave4	3.471484 + 38.009336/(1.+lognormal(6.576996, 31.631869, <stream>))
KOK2	
boiling1	weibull(106.945221, 5.988894, <stream>)
boiling2	55.681288 + 34.858756/(1.+lognormal(2.201639, 227.409298, <stream>))
boiling3	23.285819 * exp(ln(1.0/random(1.0, <stream>) - 1.0)/8.010062)
Transfer1	8.928571 + weibull(4.923066, 1.256399, <stream>)
Transfer2	7.868660 + beta(2.931340, 0.561569, 0.675891, <stream>)
Cooling	weibull(57.425303, 4.382163, <stream>)

The table consists of different mathematical distributions with different variables. Every distribution corresponds to a task for a group of sauces. The stream variable that is in every distribution is a variable that chooses which random generator should be used. If this variable is empty the ED simulation program chooses the default generator.

Excel manual

Combining Excel and ED to a powerful tool can help Findus to easily test the production from an Excel interface. Excel is a tool that Findus employee uses daily. To use this tool should not be harder than the daily work.

The result is a software mix that uses Excel as an interface to Enterprise Dynamics. The user should insert the article and how much of that article that should be produced, see Figure 6.



	A	B	C	D	E	F
1		Article	Kilo	Kettles	Sauces	Row
2		14276	20000	5	3	34
3		23927	40000	7	3	40
4		3629	2000	2	1	7
5		3880	4000	2	1	8
6		4586	10000	6	1	13
7		16839	4000	7	2	35
8		67285	1000	1	1	101
9		64917	5000	3	1	91
10		35389	10000	6	1	51
11		91655	20000	4	2	125
12				#SAKNAS!	#SAKNAS!	#SAKNAS!
13				#SAKNAS!	#SAKNAS!	#SAKNAS!

Figure 6, insert the article and kilograms of finished goods

A button "Reset" is added to the user interface. This Button is used when the user wants to run another simulation. Excel erases data and results when the user presses the"Reset" button.

The user has two ways to run the simulation. One is only a single run and to do this, the user only has to press the "Start" button in Excel, see Figure 7. The second way is to run the simulation 10 times and get the max and min times on each station. This is done in 3 steps:

1. Press the Reset button
2. Press the Start_10 button 10 times (the cell that displays the runs must decrease by one before pressing the Start_10 button again, the runs takes approximately 2 seconds each).
3. Press the Max_10 button

Figure 7, start buttons

Results from Excel

The results from both start methods are displayed in sheet two in Excel. When the user has pressed "Start" and the simulation has ended Excel changes sheet from one to two. Sheet two displays the work stations. Every sauce batch that runs on the work stations are displayed with an article, name and time, see Figure 8. The Total time on each work station is also calculated.

D	E	F	G	H	I
KVK_Sauce			KOK1_1-2		
4541	HollandaiseTop	4.3	24539	Rödvinssås	1.6
4541	Spenat	3.8			
27597	Balitoppning	3			
8629	Spenat	4.4			
Total		15.5	Total		1.6

Figure 8, result from one simulation run

When using the "Start _10" the Excel ends up in the same sheet. When the runs is equal zero the user presses the "MAX_10" button and sheet two shows the result from the 10 runs and the max and min times are calculated, see Figure 9.

D	E	F	G	H	I	J	K	L
KVK_Sauce			KOK1_1-2			KOK1_3-4		
14726	Lingon	1.1	10618	Örtagårdsås	4	14726	Köttbullsås	3.8
						10691	Ost/Skinksås	4.9
Total		1.1	Total		4	Total		8.7
10 Runs			10 Runs			10 Runs		
max		1.3	max		5.7	max		8.9
min		1.1	min		3.7	min		7.6
Total		1.1	Total		4	Total		8.7
Total		1.3	Total		5.7	Total		8.4
Total		1.1	Total		3.7	Total		8.9
Total		1.2	Total		4.9	Total		8.8
Total		1.1	Total		4.5	Total		7.8
Total		1.1	Total		4.3	Total		8.7
Total		1.2	Total		4.6	Total		8
Total		1.2	Total		5.1	Total		8
Total		1.1	Total		5	Total		7.6

Figure 9, Max_Min of 10 runs

The time unit in the table is hours. The result from the simulation run should be used to check if they can produce the sauce before a given time. The min and max times are use to capture the variance in the production.

Discussion and Conclusion

There are many different angles to consider when using a simulation. Is ED the right software to choose for this type of production flow? Are the methods used to collect time accurate? The answers to those questions are very hard to find and this is because it's very time-consuming to build models and the models have to be created in the same way so that the results can be compared.

One thing that could be implemented in the simulation that could be a bottle neck is the wagons that the sauces are transported in. I choose not to use them in the simulation because it is usually only problems with them in two of the 80 different sauces and the employees know which sauces this applies to. It had also been hard to implement the wagons in the simulation because the wagons are distributed all over the factory. A solution could be a new simulation that only consists of the wagons.

Risk analysis

In some cases the employees that boil the sauces have been suspecting that the collected data should be used as basic data to firing employees. That and the fact that the company gives notice that they are going to fire employees could possibly lead to exaggerated times in the forms. This is one thing to take in consideration when this simulation is used.

One more thing that could inflict with the data is that the cooling time on the KOK1 station should start when all of the sauce was pumped in to the tank and stop when the sauce is at 10C°. Sometimes the sauce was down at 8C° when the time was registered and this means that the cooling time could be too long. This should not cause any problem with the simulation though, on the contrary it includes in the simulation the problem that an employee must note that the temperature is at 10C° before the sauce can be tapped. Sometime the employees are on a coffee break and when they come back the temperature could be much lower than 10C°. The drawback is that the simulation is not adapted to use when you want to look only at the machines and not consider the human factor.

Conclusion

The goal was fulfilled by giving Findus a program that the planners can use to simulate the production on the boiling station. The software is installed on one of the computers on Findus and two of the employees have been taught how to use it. The routine to use the software is only on a test face so far but after some time it will be evaluated and Findus will decide if they want to use this type of simulation or not. If they want to continue with the program there are still some things that can be made to improve the model. Suggestions on such improvements are;

- Implement an interface to update the model with new articles
- Include some sort of on/off mechanism on each of the kettles and cooling tanks so the program can be used to see how the production is affected if a kettle or if a tank stops working

References

1. Findus. [Online] May 2010. http://www.findus.se/.

2. **Larsson, Jonas Karlsson & Robert.** *En Lean kartläggning och ett arbetssätt för att öka involveringen på ASKO Cylinda AB.* Göteborg : Chalmers Tekniska Högskola, 2006.

3. Incontrol Simulation Solutions. [Online] january 2010. http://www.incontrolsim.com/.

4. Microsoft Office. [Online] May 2010. http://office.microsoft.com/sv-se/excel/FX100487621053.aspx.

5. **Sargent, Robert G.** *VALIDATION AND VERIFICATION OF SIMULATION MODELS.* U.S.A. : Syracuse University, 1999. NY 13244.

6. **Banks, J., Carson, J.S. & Nelson B.L.** Discrete-event system simulation. 1996.

7. ExpertFit. [Online] May 2010. http://www.averill-law.com/ExpertFit-distribution-fitting-software.htm.

8. **Dynamics, Enterprise.** *ANNEXES TUTORIAL.* The Netherlands,3528 BJ Utrecht : s.n., 2007.

9. —. *TUTORIAL ED.* The Netherlands,3528 BJ Utrecht : s.n., 2007.

10. —. *QUICKSTART GUIDE TO XML.* The Netherlands,3528 BJ Utrecht : s.n., 2007.

Appendix A

Appendix A contain of the formerly that where handed out to the workers on Findus. The headlines are in Swedish.

KOK2

Datum:

Artikel	Typ	GrytaNR	Start kok	Stop kok	Start överföring	Stop överföring	Start kyl	Stop kyl

Övrigt

KOK1

The boiling side:

Datum;

Artikel	Typ	GrytaNR	Start kok	Stop kok	Start överföring	Stop överföring

Övrigt

The cooling side:

Datum: ________________

Artikel	Typ	TankNR	Start kyl	Stop kyl	Start Töm/Väg	Stop Töm/Väg

Övrigt

KVK

Datum;

Artikel	Typ	GrytNR	Start kok	Stop kok

Övrigt

Appendix B

This appendix contains an example of the code for the simulation program. The example is from the programming of the KVK station.

```
{----------- KVKqueue-----------trigger on entry------------}
{Send the Mash to output one that goes to kettle three}
 If(Inlist(Label([Sauce],i),([Mos152433]),([Mos152437])) <> 0,
    setLabel([Kettle],1,i),
    setLabel([Kettle],2,i)
  )

{----------- DummyQueue1-----------trigger on entry------------}
  SetExprAtt(
                [CycleTime],              {set the mash cycle time}
                KVKsauce(7,3),
                AtomByName([KVK3],Model)
)
{----------- DummyQueue1-----------trigger on exit------------}
Do(
     KVKq1_InAr:= ValueToString(label([Ar],i)),
     if(CompareText (KVKq1_inAr ,KVKq1_OutAr),
        KVKq1_OutAr:=ValueToString(label([Ar],i)),                    {Check if there is a new artikle}
        KVKq1_OutAr:=[New]
  ),
     KVKq1_InSauce:= label([sauce],i),
     if(CompareText (KVKq1_InSauce ,KVKq1_OutSauce),
        KVKq1_OutSauce :=label([sauce],i),                  {Check if there is a new sauce}
        KVKq1_OutSauce:=[New]
  ),
  if(And(CompareText (KVKq1_InSauce ,KVKq1_OutSauce),CompareText (KVKq1_INAr ,KVKq1_OutAr)),
    Do(
         KVK_data1_caunt_q1:= KVK_data1_caunt_q1+1,         {counts how many of the same sauce}
         if(Label([Kok],i)=KVK_data1_caunt_q1,            {if the amount of sauces is the same as the label kok (kettles),}
           setlabel([Last],1,i)                         {this is the last sauce of that kind}
      )
    ),

    Do(
        SetCell(KVK_data2_caunt_q1,3,Time,DATA(0,0)),               {new sauce set start time in table DATA}
        SetCell(KVK_data2_caunt_q1,1,Label([Ar],i),DATA(0,0)),
        SetCell(KVK_data2_caunt_q1,2,Label([sauce],i),DATA(0,0)),
        KVK_data2_caunt_q1:=KVK_data2_caunt_q1+1,
        KVK_data1_caunt_q1:=1
     )
  ),
  KVKq1_OutSauce :=label([sauce],i),
  KVKq1_OutAr :=ValueToString(label([Ar],i)),
  setlabel([KVKM],1,i)
)


{----------- DummyQueue2-----------trigger on entry------------}
Do(
  KVK_InSauce:= label([sauce],i),
  if (CompareText (KVK_InSauce ,KVK_OutSauce),
     KVK_OutSauce :=label([sauce],i),                   {Check if there is a new sauce}
     KVK_OutSauce:=[New]
  ),
   {Select a kettle for the sauce}
    if(CompareText (KVK_InSauce ,KVK_OutSauce),
       Do(
          KVK_Same_Sauce_q2:=1,
          If(KVK_kettle=2,
```

```
                Do(
                    setLabel([Kettle1],2,i),
                    KVK_kettle:=1,
                    For(KVK_caunt := 1, KVK_caunt < 20, Inc(KVK_caunt),
                         If(Inlist(Label([Sauce],i),KVKsauce(KVK_caunt,1)) <> 0,
                            Label([Row_KVK],i):=KVK_caunt
                         )
                    )
                 ),
                 Do(
                    setLabel([Kettle1],1,i),
                    KVK_kettle:=2,
                    For(KVK_caunt := 1, KVK_caunt < 20, Inc(KVK_caunt),
                         If(Inlist(Label([Sauce],i),KVKsauce(KVK_caunt,1)) <> 0,
                         Label([Row_KVK],i):=KVK_caunt
                         )
                    )
                 )
              )
           ),
           Do(
               KVK_Same_Sauce_q2:=0,
               If(KVK_kettle=2,
                 Do(
                     setLabel([Kettle1],1,i),
                      KVK_kettle:=2,
                      Do(
                           For(KVK_caunt := 1, KVK_caunt < 20, Inc(KVK_caunt),
                                If(Inlist(Label([Sauce],i),KVKsauce(KVK_caunt,1)) <> 0,
                                   Label([Row_KVK],i):=KVK_caunt
                                )
                           )
                   )
               ),
               DO(
                    setLabel([Kettle1],2,i),
                    KVK_kettle:=1,
                    Do(
                        For(KVK_caunt := 1, KVK_caunt < 20, Inc(KVK_caunt),
                             If(Inlist(Label([Sauce],i),KVKsauce(KVK_caunt,1)) <> 0,
                                Label([Row_KVK],i):=KVK_caunt
                             )
                        )
                    )
               )
           ),
           KVK_OutSauce := label([sauce],i)
      )
 )

{----------- DummyQueue2-----------trigger on exit------------}
Do(
    Do(
        KVKq2_InAr:= ValueToString(label([Ar],i)),
        if(CompareText (KVKq2_InAr ,KVKq2_OutAr),
          KVKq2_OutAr:=ValueToString(label([Ar],i)),          {Check if there is a new artikle}
          KVKq2_OutAr:=[New]
    ),
        KVKq2_InSauce:= label([sauce],i),
        if(CompareText (KVKq2_InSauce ,KVKq2_OutSauce),
          KVKq2_OutSauce :=label([sauce],i),          {Check if there is a new sauce}
          KVKq2_OutSauce:=[New]
    ),
{counts how many of the same sauce if the amount of sauces is the same as the label kok (kettles), this is the last sauce of that kind }
      if(And(CompareText (KVKq2_InSauce ,KVKq2_OutSauce),CompareText (KVKq2_INAr ,KVKq2_OutAr)),
        Do(KVK_data1_caunt_q2:= KVK_data1_caunt_q2+1,
```

```
            if(Label([Kok],i)=KVK_data1_caunt_q2,
               setlabel([Last],1,i)
            )
      ),
     Do(
       {new sauce set start time in table DATA}
       SetCell(KVK_data2_caunt_q2,8,Time,DATA(0,0)),
       SetCell(KVK_data2_caunt_q2,6,Label([Ar],i),DATA(0,0)),
       SetCell(KVK_data2_caunt_q2,7,Label([sauce],i),DATA(0,0)),
       KVK_data2_caunt_q2:=KVK_data2_caunt_q2+1,
       KVK_data1_caunt_q2:=1
     )
  ),
   KVKq2_OutSauce :=label([sauce],i),
   KVKq2_OutAr :=ValueToString(label([Ar],i))
),
 if(KVK_kettle=1,  {check witch kettle the sauce should boil in and set the cycletime and setuptime for  this kettle}
    Do(
        if(KOK1_Same_Sauce_d3=1,
        Do(
            SetExprAtt(
                          [CycleTime],
                          KVKsauce(Label([Row_KVK],i),3),
                          AtomByName([KVK1],Model)
            ),
             SetExprAtt(
                          [Setuptime],
                           KVKsauce(Label([Row_KVK],i),2),
                           AtomByName([KVK1],Model)
             )
        ),
        Do(
            SetExprAtt(
                          [CycleTime],
                          KVKsauce(Label([Row_KVK],i),3),
                          AtomByName([KVK1],Model)
            ),
             SetExprAtt(
                          [Setuptime],
                          [Normal(mins(15),Mins(2))],
                           AtomByName([KVK1],Model)
             )
         )
      )
),
Do(
    if(KOK1_Same_Sauce_d3=1,
       Do(
           SetExprAtt(
                         [CycleTime],
                         KVKsauce(Label([Row_KVK],i),3),
                         AtomByName([KVK2],Model)
           ),
           SetExprAtt(
                        [Setuptime],
                        KVKsauce(Label([Row_KVK],i),2),
                        AtomByName([KVK2],Model)
           )
       ),
        Do(
           SetExprAtt(
                         [CycleTime],
                         KVKsauce(Label([Row_KVK],i),3),
                         AtomByName([KVK2],Model)
           ),
           SetExprAtt(
```

```
                              [Setuptime],
                              [Normal(mins(15),Mins(2))],
                              AtomByName([KVK2],Model)
              )
           )
        )
      )
   )
)

{-----------LeaveKVK-----------trigger on exit-----------}
{this cod check if the sauce is the last of that sauce batch, if it is the end time is put in the DATA table }
If(and(
     Label([Last],i)=1,
     Label([KVKM])=1
    ),
     For(KVK_data3_caunt:= KVK_data2_caunt_q1+KVK_data2_caunt_q2, KVK_data3_caunt <> 0,  dec(KVK_data3_caunt),
          If (
              and(
                   Inlist(Label([Ar],i),DATA(KVK_data3_caunt,1)) <> 0,
                  Inlist(Label([Sauce],i),DATA(KVK_data3_caunt,2))<> 0
              ),
               SetCell(KVK_data3_caunt,4,Time,DATA(0,0))
            )
        ),
         For(KVK_data3_caunt:= KVK_data2_caunt_q1+KVK_data2_caunt_q2, KVK_data3_caunt <> 0, dec(KVK_data3_caunt),
              If (
                  and(
                        Inlist(Label([Ar],i),DATA(KVK_data3_caunt,6)) <> 0,
                        Inlist(Label([Sauce],i),DATA(KVK_data3_caunt,7))<> 0
                  ),
                  SetCell(KVK_data3_caunt,9,Time,DATA(0,0))
              )
           )
)
```