Free University of Bolzano-Bozen
Faculty of Computer Science

Implementation of a scheduling component in a production management system

Author:
Michael Mairegger

Supervisor:
Prof. Johann Gamper

Academic Year 2010/2011

Abstract

The project of this thesis comprises the planning, implementation and the testing of a scheduling algorithm which optimizes a production plan according the selected order. An attempt was made to optimize this so far that the production costs, production time and the entire production effort are minimized. The data is managed in a program developed by mine called “Schwer Verwaltung”, the company name was eponymous, which covers all the company’s needs like inventory-, production-, storage-management including creation of delivery and invoice documents. The program is still being worked out in many areas in the future. In the background of the application the latest version of Microsoft SQL Server 2008 R2 is working and LIN2SQL similar dialect is used to query data. The algorithm itself analyzes the existing production plans and compares them with the newly inserted productions. These are then scheduled under consideration of minimizing the production costs, production time and adherence of the delivery date, required by the contracting authority, to the machines into their existing production plans.

The biggest challenge during the project was to find a way how the available data is analyzed properly and the computed result is as accurate as possible.

All implemented algorithm were tested using “nUnit” for their proper working. Through various tests and comparisons has been confirmed that the applied algorithms optimizes the production cycle time. In addition, a reduction in production costs and production relief has been achieved.

Zusammenfassung

Das Projekt dieser Diplomarbeit besteht aus der Planung, Implementierung und dem Testen eines Produktionsalgorithmus welcher anhand der ausgewählten Bestellungen einen optimierten Produktionsplan erstellt. Es wurde versucht diesen soweit zu optimieren, dass die Produktionskosten sowie die Produktionszeit und der gesamte Produktionsaufwand minimal sind. Die Daten werden in einem von mir entwickelten Programm namens „Schwer Verwaltung“, der Firmenname war namensgebend, verwaltet welches die gesamte Lagerverwaltung angefangen bei der Bestellung, Produktion, Lager- und Rohmaterialbestand über Lieferschein und Rechnung alles in der Firma notwendige abdeckt. Das Programm wird in Zukunft noch in vielen Bereichen ausgearbeitet. Im Hintergrund der Anwendung arbeitet die aktuellste Version des Microsoft SQL-Server 2008 R2 an welchem durch das Programm durch LINQ2SQL ähnlichem Dialekt die Daten abgefragt werden.

Der Algorithmus an sich analysiert die vorhandenen Produktionspläne und gleicht diesen mit den neu einzufügenden Produktionen ab. Diese werden dann unter Berücksichtigung der Minimierung der Produktionskosten, Produktionszeit und Einhaltung des vom Kunden gewünschten Lieferdatums an den vorhandenen Maschinen in dessen vorhandenen Produktionsplan eingeplant.

Die größte Herausforderung während des Projektes war es einen Weg zu finden wie die vorhandenen Daten korrekt analysiert werden und das Resultat so genau wie möglich vorhergesagt wird. Alle implementierten Algorithmen wurden durch „nUnit-Tests“ auf ihren korrekten Ausgang hin überprüft.

Durch verschiedene Tests und Vergleiche wurde bestätigt dass die angewandten Algorithmen den Produktionszyklus zeitlich optimierten. Zusätzlich wurde eine Produktionskostensenkung und eine Entlastung der Produktion erreicht.

Sommario

Il progetto di questa tesi di laurea consiste nella pianificazione, nell`implementazione e nel sottoporre ad un test un algoritmo di produzione, che crea un piano di produzione ottimizzato in base alle ordinazioni scelte. Si è tentato di ottimizzarlo a tal punto, che sia i costi che il tempo di produzione e l`intero carico di produzione siano minimali.

I dati vengono gestiti in un programma da me sviluppato, chiamato "Schwer Verwaltung" - la ragione sociale ne ha dato il nome - il quale comprende l´intera gestione del magazzino, iniziando dall´ordine, dalla produzione, dalla scorta disponibile e dalla giacenza di materiale grezzo alla bolla di consegna ed alla fattura, e quindi tutto ciò che è necessario all´interno della ditta.

In futuro il programma sarà ancora elaborato in molti campi. Sullo sfondo dell´applicazione lavora la versione più attuale di Microsoft SQL-Server 2008 R2 nella quale vengono consultati i dati per mezzo del programma simile al dialetto attraverso LINQ2SQL.

L´algoritmo in sè analizza i piani di produzione disponibili e li adatta alle nuove produzioni da inserire. Queste vengono poi previste nel piano di produzione delle macchinari disponibili tenendo conto della riduzione ai minimi termini dei costi e dei tempi di produzione e del rispetto della data di consegna richiesta dal cliente.

La sfida più grande nel corso del progetto era quella di trovare un modo come analizzare correttamente i dati disponibili e come pronosticare il resultato più precisamente possibile. Tutti gli algoritmi implementati sono stati esaminati attraverso test-nunit in merito al loro risultato corretto.

Attraverso vari test e confronti è stato confermato che gli algoritmi applicati ottimizzano in tempo il ciclo di produzione. Inoltre è stato raggiunto un abbassamento dei costi di produzione e un alleggerimento della produzione.

Contents

1. Introduction

1.1.Motivation

Having an optimized production scheduling plan is a very important factor of every company that is specialized in production. Often this machine scheduling is done via the FCFS (First Come First Serve) principle. But this FCFS principle has the drawback that not every order that is given into commission by the contracting authority can be produced until to the date that the contracting authority wants to. Therefore the aim of this project is that the company, a metal-processing company, I work for, needed a new solution for managing and optimizing their production plans.

1.2.Problem Description

Until now the scheduling of their production plan was done by hand. This was done by the user by calculating for each order whether he would be able schedule it in order to produce it on any of his owning machines until to the due date. The next step the user was faced was the question where put this order into production. The possible solution varies. The new order could be placed:

- Single Placement
  - At the end of the production queue of any machine
  - On any position of the production queue of any machine
- Split Placement
  - At the end of the production queue of at least two machines
  - On any position of the production queue of at least two machines where the position in the queue may vary.
- Expand Placement
  - Expand a production of a Single Placement on any machine
  - Expand a production of a Split Placement on at least two machines

As you can imagine the problem of manual computing is that this process is very time consuming and error prone if you are not concentrated. If you work in a company that owns three such production machines this computation process might be easy but if you have ten or more machines you will not be able to compute the production scheduling plan without any help. Looking at the previous enumeration we can see that there are so many factors that an order can be scheduled. Many factors like production time and production costs, to mention only two of them, depends on that scheduling and therefore the result can vary dramatically from producing on one machine or on the other. So the company tutor asked whether I could build this computation into their administrative software, which I develop, in order to automate this production plan scheduling.

1.3.Proposed Solution

The question that arises was how to ensure that every order is scheduled in the existing machine production plan in such a way that it will produced not later than the due date the contracting authority gave. I decided to expand the software that is in use to implement the requested features. I implemented a new scheduling algorithm that schedules an order so that the order can be delivered on time regarding following factors and solution steps and with respect to the enumeration seen in 1.2:

- Production costs
- Production time
- Same production element scheduled on others machines scheduling plans
- Split order to produce it on many machines
- Change production order of existing scheduling plans
- Expand production with the new order

The algorithm should come to a solution in that the production costs and production time of the scheduling is minimal in order to allow the company to get the biggest revenue. According to the time and cost factor another important factor is that I always have to ensure that every order given into commission by the contracting authority will be produced under any circumstances not after the given due date.
We propose in this thesis an algorithm that takes care of the problems listed in 1.2. The algorithm analyzes all of the existing machine production plans and tries to schedule the new order on a machines production list or splits the order on many machines. After that computation the desired outcome would be that the production costs and production time is minimized and that the user does not need any time to calculate the production plan by hand.

1.4.State of the Art

Nowadays each company is forced to lower their production costs more than ever before. The pressure of competition is very high. Starting with the development of the computers people also started to develop algorithms that makes time consuming task as easy as possible. Prior the era of computers people were forced to manual compute everything because they had no computer that did this work. They had some instruments that helped in computation but no independent working machine that computes something according the given input.
This changed after developing the computer. Complex algorithms were made to compute tasks, which previous took hours to finish and now with help of computers only a few seconds. Also companies began to think how they could improve their complex production plans, which took very long time to compute, in order to compile it in a fast, cost saving and in a reliable way. They started to develop complex scheduling algorithms that try to optimize even the unimaginable factors. Very complex scheduling algorithms were developed. Each of them were optimized and adjusted to the company's needs. This is because each company works in a different kind of industry. Car industry has other scheduling algorithms than computer industry and so has the metal industry. Also

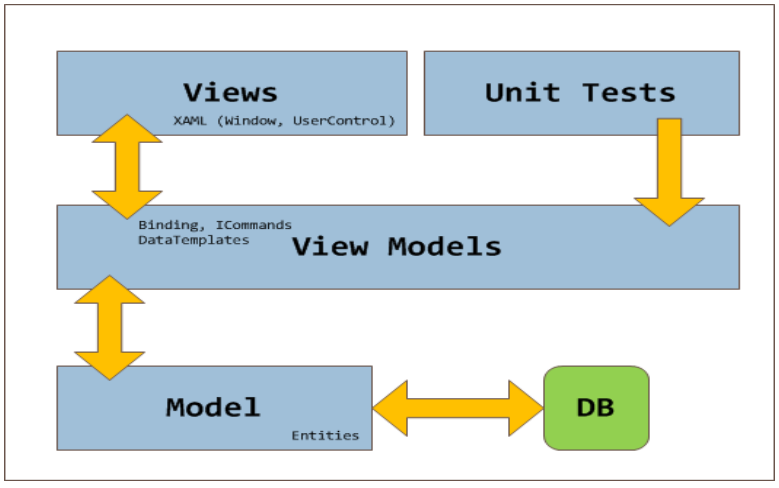
the company I work for decided to optimize their production scheduling algorithm. How this is done is discussed in this thesis.
In the last years there has been much research on scheduling algorithms and optimizing production plans (1) (2) (3) (4) (5) (6).
A related work to this thesis which presents in a very detailed way the problems in scheduling is (1). Peter Brucker, the author, writes especially in chapter 4 “Single Machine Scheduling Problems”, chapter 5 “Parallel Machines” and chapter 7 “Due-Date Scheduling” a lot about the main problems in scheduling that also I covered.

2. System Architecture

The application is developed using a three tier application layer. The three tiers are as follows:

- Data Layer (Database)
- ViewModel Layer (Logic)
- View Layer (GUI)

Therefore the data layer is represented by the Microsoft SQL Server 2008 R2 and all data is stored there. The ViewModel part contains all the logic, algorithm, user handles the user input. The View part contains all stuff about displaying what the ViewModel delivers. Furthermore it delivers all user input back to the ViewModel so that this layer can handle it and if necessary store data using the data layer.



I used following frameworks:

- *.NET Framework 4.0*, a framework based on .NET Technology by Microsoft that provides many base classes for developing this Project.
- *Cinch v2*[1], a MVVM (Model-View-ViewModel framework), that provides all classes and interfaces for ensuring the application to be a three tier application. Cinch is a MVVM framework that exposes a number of helper classes to allow

[1] http://cinch.codeplex.com/

the developer to quickly get to grips with creating scalable testable MVVM frameworks as quickly as possible. (7)

- *MEFedMVVM*[2], a dependency injection container, service locator and framework for developing application with MEF.
  The Managed Extensibility Framework (MEF) is a composition layer for .NET that improves the flexibility, maintainability and testability of large applications. MEF can be used for third-party plugin extensibility, or it can bring the benefits of a loosely-coupled plugin-like architecture to regular applications. (8)
- *LINQKit*[3], a free set of extensions for LINQ to SQL and Entity Framework power users (9).

For development of the SQL-Queries I used T-SQL, short for Transact-SQL, a proprietary extension to SQL developed by Sybase and Microsoft. Using T-SQL the developer is able to write IF-statements, ***DELETE*** statements with ***FROM*** and ***JOIN*** clause, ***BULK INSERT*** and has the possibility to handle errors by using ***TRY CATCH***.

3. Design

3.1. Order and stock valutation

For placing an order that was given in commission which has a specific due date, some conditions need to be satisfied prior that it can placed onto a machine schedule. At the beginning the contracting authority requests a specific element as an incoming order. After the confirmation of the company, the order is confirmed and therefore ready for production. Prior the production there are some steps that have to be processed.

- At the beginning of the process it is checked whether the ordered goods are already in stock. If so, it is checked whether they are not already reserved by another order. If the amount of requested goods is in stock and not reserved for any other company the goods can be released for delivery. Then we do not have to proceed to further steps.

If we have not enough goods in the stock we have to produce them. We will see in this chapter all steps that are necessary for an optimal production.

3.2.Select preferred machine

The first step after inserting an order and enabling them for the production process is to search all of the machines on which the specific element given into commission can be produced. To determine these machines we have to search the corresponding element in the production elements table. In order to find the element we have to consider that an element might be one single element or might also be mounted by other elements and those elements with others and so on. Thus producing one element might result in a

[2] http://mefedmvvm.codeplex.com/
[3] http://www.albahari.com/nutshell/linqkit.aspx

flow of production of many elements. All those elements have to be produced unless they are in stock already. Therefore we need to find for any elements, where we do not have the needed amount in stock, the corresponding machine, preferred position, etc. For better understanding the process, the initial element will be called *parent element*, since it might be mounted further on and therefore it might be the parent of a lot of child items, the so-called *child elements*. Accordingly all elements that are used for the mount process are called *child elements*. Once found the right parent element we can traverse the database table that holds the connection between an element and the corresponding machine on which the element can be produced and put all machines into a list. With that step we have immediately a list containing all machines that can produce this or any descendant element. Furthermore we have to select for each child-item the corresponding machine and to choose to produce it there. We will not deepen in detail, because the selection process of the machine corresponding to the production element is always the same.

3.3.Machine Selection Process

Now we are in the state that we have found all corresponding machines that are able to produce this and any descendant element. But a new problem arises: if we found ten machines, which could produce the desired part, we do not know if also their production rate and production cost is the same. They could be different since the machines might be of different type. Therefore…machine. Now this process will be discussed in detail Therefore we can say that the possibility for a machine to be able to produce an element does not ensure that it is optimal to produce on that specific machine. We will discuss this thought in detail now.
In this chapter we will always refer to following types of machines:

- lathe
- turning lathe
- milling machine

3.3.1. Selection by time

In order to get a good result, the consideration of the production time is an essential factor. It is in the company's interests that the production time on the machine, the algorithm selected, is as short as possible. Since a shorter production time results in a higher production rate. In order to find the machine with the best production rate we have to compute for every machine, previously selected in chapter 3.1, the total amount of production time that is needed for the element. If we have an element that can be produced on any type of any machine we have the problem that on a milling machine and on a lathe we have no automation. That means that after producing a single piece of the selected order a person has to extract it manual from the machine. Therefore we need additional to the machine a person that is working on it all the time. On a turning lathe this element extraction is done automatically and no additional person is needed for the production process, unless for inspection and sampling measurements. Furthermore extracting a produced element from a machine and putting back in raw

material takes some amount of time. Often this extraction may take longer than the production process of the single element.
Therefore it is clear that if we choose a turning lathe as production machine, which is automatic, over a manual served milling machine or a lathe we will have a better production rate. A better production rate is always preferred. Regarding the time issue in extracting the produced element it applies that using a manual served machine in general we will always have lower production rates and therefore higher production costs.
Another additional point of interest is the already mentioned extracting time. This is the time needed after the element was produced, removed from the machine and inserted a raw material to for the next element. As already mentioned on a lathe and milling machine this process is done by hand. On a turning lathe this is done automatically. In order to retrieve the amount of time needed to produce the element we need to sum up the production time and the extraction time. On each machine the production time and extraction time may be different.
So we have to find for each machine the corresponding total production time. This is done by looking in the corresponding table that maintains all production time data. After collecting all the production times for a specific element we have to evaluate them. This is generally done by ordering the retrieved data ascending according the sum of total production time.

Machine lists

In the following examples we consider the following machine lists:

- Machine #1 Machine Type: lathe
- Machine #2 Machine Type: milling machine
- Machine #3 Machine Type: turning lathe
- Machine #4 Machine Type: turning lathe

Now we may have following situation:

Situation 1

- Machine #1: T = 35 sec Machine Type: lathe
- Machine #2: T = 40 sec Machine Type: milling machine
- Machine #3: T = 65 sec Machine Type: turning lathe
- Machine #4: T = 70 sec Machine Type: turning lathe

If we have distinct production times we have no problems. We can choose the machine with the smallest amount of production time. But does this always be the best choice? No, it is not. It may be the fastest option, regarding the production time, but not the cheapest. As we have seen above, the extracting time is faster on an automatic machine than on a machine where the extracting process is done by hand. Therefore we do not consider the situation 0 because it will generally never appear. But if this situation arises

it will be corrected later thanks to the algorithm for computing the production costs per machine.

Such a situation may arise because of faster extraction on automatic machines:

Situation 2

- Machine #4: T = 50 sec Machine Type: turning lathe
- Machine #3: T = 52 sec Machine Type: turning lathe
- Machine #1: T = 80 sec Machine Type: lathe
- Machine #2: T = 85 sec Machine Type: milling machine

3.3.2. Selection by production costs

The next factor that we need to consider, is, that each machine has, depending on different factors, different production costs. Some machine needs more electrical power, or was more expensive at purchase time; the production creates more waste material, or it requires a permanent attendee etc. All these costs are fixed costs and were computed already in advance. The fixed costs are indicated as cost per hour. These costs are defined as constant amounts. That means that variations in energy consumption, wasted material and so on are not considered. Therefore we consider the wasted material as constant amount. Fluctuations in material waste are also not considered as a factor in the production cost calculation. After determining each part of the costs we can sum up all those cost factors that are stored in the database. After summing up all those factors up we have a list that we order ascending by the sum of the production costs to find the machine with the lowest production costs.

3.3.2.1. Situation 3

- Machine #3: C = 23€ / hour Machine Type: turning lathe
- Machine #4: C = 25€ / hour Machine Type: turning lathe
- Machine #1: C = 50€ / hour Machine Type: lathe
- Machine #2: C = 53€ / hour Machine Type: milling machine

As a result of the fact that the last two machines are manually used we have higher production costs caused of the permanent attendee. We can consider that in general manually operated machines have lower acquisition costs than automatic ones.

3.3.3. Selection by evaluating time and production cost

We have seen in 3.3.1 and 3.3.2 that we now have two separate lists: one list ordered by production time and the second one by production costs. The next challenge is to merge those two lists into one. Doing that we get the following situation:

Merging Situation 1 (see at 0) with Situation 3 (see at 3.3.2.1) results in the following situation:

3.3.3.1. Situation 4

- Machine #1: C = 50€ / hour T = 35 sec Machine Type: lathe
- Machine #2: C = 53€ / hour T = 40 sec Machine Type: milling machine

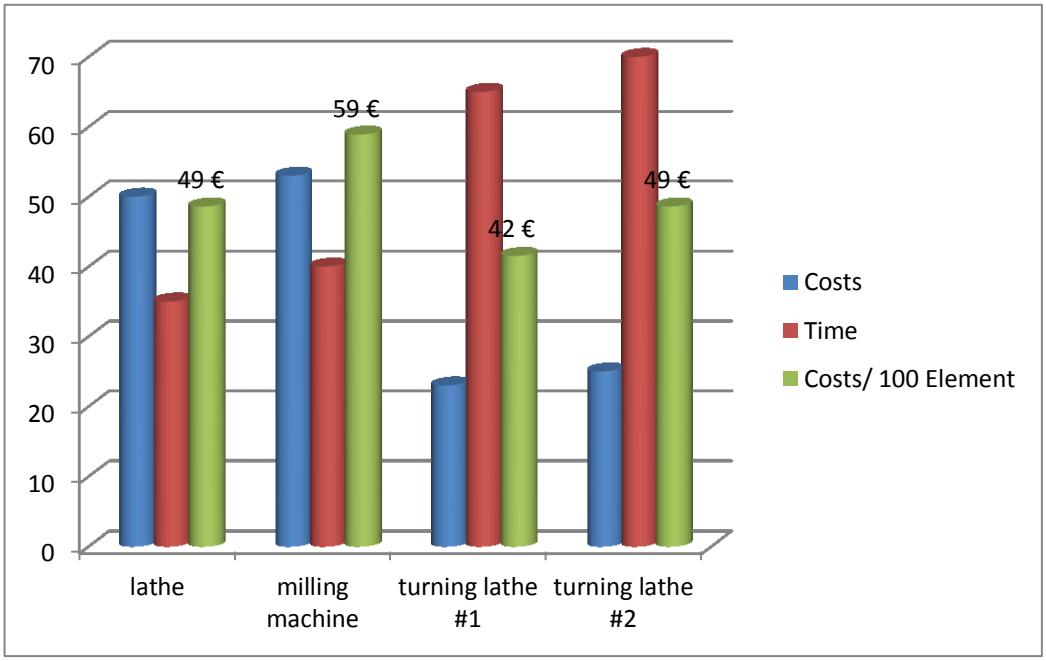
- Machine #3 C = 23€ / hour T = 65 sec Machine Type: turning lathe
- Machine #4: C = 25€ / hour T = 70 sec Machine Type: turning lathe

The next step which follows is doing some short calculation, like computing the total production costs per produced piece on every possible machine. This is achieved with following computation:

$$Cp = \frac{Cm * t}{3600}$$

Where **Cp** stands for "Production costs per element" and **Cm** stands for "Cost for machine per hour". Calculating the result we get the following costs.

Note that for better visibility the green bar indicates the production costs for 100 elements.



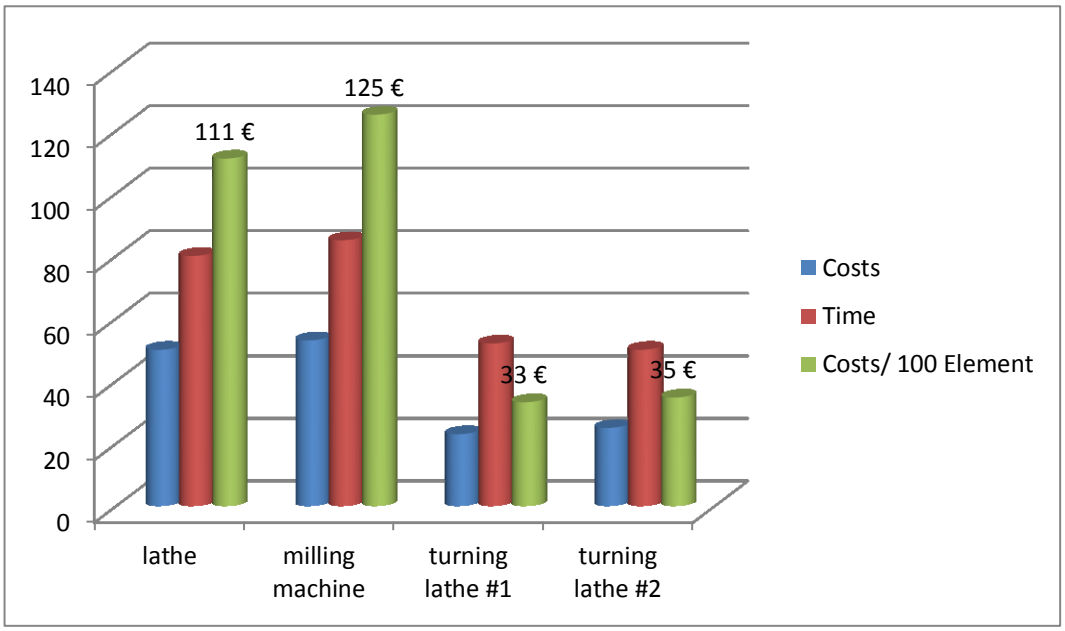
Graph 3-1 Production costs comparison

As shown in the graph above, although the first two machines have a lower production time, which will never occur because they are manually operated, the resulting costs is higher than the others because of the higher fixed production costs. The next situation we have, which is more realistic, is the situation when we merge Situation 2 (see at 0) with Situation 3 (see at 3.3.2.1) resulting in the following situation:

3.3.3.2. Situation 5:

- Machine #1: C = 50€ / hour T = 80 sec Machine Type: lathe
- Machine #2: C = 53€ / hour T = 85 sec Machine Type: milling machine
- Machine #3 C = 23€ / hour T = 52 sec Machine Type: turning lathe
- Machine #4: C = 25€ / hour T = 50 sec Machine Type: turning lathe

Note that for better visibility the green bar indicates the production costs for 100 Elements.



Graph 3-2 Production costs comparison

Now we have a more realistic situation where the automatic operated production machines have a lower production time and therefore the result that the fixed production costs is also lower than on the manual operated machines. The graph shows, that the desired machine on which we will start producing the element will be either the turning lathe #1 or the turning lathe #2. The better choice will be the turning lathe #1. The question that arises is why do we need a lathe and a milling machine when they are so expensive and slow in production and therefore obviously never chosen? Well as fast that this question arise as fast it is answered. There are some elements that cannot be produced on a turning lathe. Therefore if we have the same situation like above without both turning lathes we have to choose between the lathe and the milling machine. In that situation the lathe will win over the milling machine. Well, now we have selected the preferred machine but it is not sure that it will be selected as production machine. There are also other factors that depends if a machine is finally selected.

3.3.4. Selection by being able to produce in time

After selecting the best machine related to the total production costs we have to find out whether the machine with the lowest production cost would also be able to produce the element, which was placed in order, until to the due date that the contracting authority proposed. If the machine that was previously selected is not able to produce the total amount of element until to the due date, then the next cheapest machine will be selected and checked whether it is able to produce the entire order on time and so on and so forth.

The procedure that checks whether the production fulfills on time is done as follows:

3.3.4.1. Trying append on end of queue

The first and easiest way to check whether the order fulfills the due date is to append the order to the end of the existing production queue. This is done by searching the last element in the production queue and by remembering the end timestamp as starting point for the new production. If the queue would be empty the actual date is selected as start date. After that we have found the exact starting point of the production, we can calculate the time needed for production. This is achieved by multiplying the production amount of produced elements with the total production time of a single element. As discussed in the previous section the total production time is the sum of the production and extraction time.

$$Tp = Tpe * n$$

Where ***Tp*** is the Total production time, ***Tpe*** the production time of a single element and ***n*** the number of elements to produce.

We can now add this calculated timespan to the starting time stamp in order to get the end point of the production. But this would be inaccurate because we do not produce any element on weekends and during night. Therefore we store for every single machine a daily production time span in where the machine runs. Knowing that, we can easily compute the real estimated end time of the production:

$$TimeEnd = TimeStart + HoursOfFirstDay + n * \ fullDays + HoursOfLastDay$$

$$HoursOfFirsDay = MachineStopAtTime - StartTime$$

$$FullDays = Floor\left(Round\left(Tp - \frac{HoursOfFirstDay}{MachineStopAtTime - MachineStartAtTime}\right)\right)$$

Now that we have completed this calculation we have the amount of full days we of production. These amounts of days have to be added to the start date plus one day, because we have already subtracted the ***HoursOfFirstDay***. We add those days avoiding Saturdays, Sundays and Holiday. All the holiday dates are stored in the database, except those that have to be computed like Easter and Pentecost. After getting the last whole production day, the last step still remains is to add the ***RemainingTime*** to the ***MachineStartAtTime*** in order to get the exact end time stamp of the production.

Now we have the exact end time of the production or to say it with better words the estimated end time. There are many factors which might postpone the end time. This is caused by machine defects, production difficulties, material bearing is empty and that the person that is assigned to the specific machine, if it is a manual operated machine, is in vacation or otherwise prevented from doing the work. Well these are factors that we could not avoid, and therefore we do consider that all goes well. In other words if we consider this downtime we could possibly increase the total production time of one element.

Under consideration of the productions end time we are now able to see whether we finish the production on time or we delay the order. If we delay we try the same

calculation on every other possibly machine that follows the order seen in Graph 3-2.
If we move so on we may find a machine that is able to produce the order on time. But sometimes if the machines are very busy for a very long time we might have the situation that we do not find any machine that is able to finish the order until the due date. This would be a problem. We now have three possible solutions to fix the issue.

1. The first and easiest solution would be to notify the user of the problem that the order could not processed until the desired date. This will result in the risky problem that the user has to notify the contracting authority that he is not able to produce the order on time. If this happens once the contracting authority will have no problem to postpone the order until that day the program has calculated. If this happens another time the contracting authority will maybe also have no problem. But if we get regularly production end dates that are not on time the contracting authority will search another production company.
2. The second and trickier solution is also the preferred one. If we find no machine where we can append the production we try to insert the production on a position within the production list. We will discuss this in the chapter 3.3.4.2.
3. The last solution would be to split the production to produce it on more than one machine. In that way we have more than one machine that produces a part of the total production and we might have the case that the end date of the production is earlier than expected. This will be discussed in chapter 3.3.4.3.

	Hours Needed	Start Time Stamp	End Time Stamp	Due Date	Production On Time
Production A	5	0	5	10	True
Production B	3	5	8	9	True
Production C	2	8	10	11	True
Production D	8	10	18	19	True
New Production	5	18	23	20	False

Table 3-1 Machine on which we cannot append

	Hours Needed	Start Time Stamp	End Time Stamp	Due Date	Production On Time
Production A	5	0	5	10	True
Production B	3	5	8	9	True
New Production	5	8	13	20	True

Table 3-2 Machine on which we can append

3.3.4.2. Trying inserting on every other position

As a second solution for the problem if we could not find any machine on which we could produce the elements on time is that we try to move the production step by step towards the beginning of the production list until the production satisfies the desired end date. This is done by swapping the last production, which is the new one, with the penultimate. If we then check the productions end date we can see whether we can produce on time or not. If not we swap again until we find a position in the existing

production plan of the machine where we are able to produce the whole order on time. Well now we might have found a position in the production plan where we produce the order on time but what about the other productions that were already in the list. Could it be that we have it affected and moved it behind the due date of it because of the swapping process? Every other production that we swapped might now be in the "unstable" case where that production does not finish on time. We might have luck and have not affected other productions but we cannot risk that. Swapping in such a so inconsiderate is very bad. So what is the solution?

If we start swapping we can do that but we have to check the production item that we moved back, immediately. This is done by swapping both production elements. Then we check the production end date of the new production. This end date would correspondingly be the start date of the production that we moved backwards. Starting with this date we can calculate regarding the production time the end date of the production and check if we end previous the desired date given by the contracting authority or not. If this is not the case we are not able to move the new production any further in the production chain and swap it back. So we have to check on other machines if we would be able to move the production in the production chain.

	Hours Needed	Start Time Stamp	End Time Stamp	Due Date	Hours to due Date	Production On Time
Production A	5	0	5	10	5	True
Production B	6	5	8	9	1	True
Production C	2	8	10	11	1	True
Production D	8	10	18	19	1	True
New Production	6			15		False

Table 3-3 Production Plan where we cannot insert on another position

	Hours Needed	Start Time Stamp	End Time Stamp	Due Date	Hours to due Date	Production On Time
Production A	3	0	3	10	7	True
Production B	3	5	8	9	1	True
New Production	6			15		True

Table 3-4 Production Plan where we can insert on another position

Hours Needed	Start Time	End Time	Due Date	Hours to due	Production On Time

		Stamp	Stamp		Date	
Production A	3	0	3	10	7	True
New Production	6	5	11	15	4	True
Production B	3	11	14	17	1	True

Table 3-5 Situation of Table 3-1 after inserting the new production on the new position

3.3.4.3. Trying to split the order on different machines

If we do not find any position processing 3.3.4.1 and 3.3.4.2 we are not done. The last step is that we split a production in two or more sub productions. In such a way we maybe have that splitting the production may result in the situation that we finish the production on time. In order to check whether we are able to finish on time we have to compute some steps.

First of all we have the machine list ordered by costs as seen in Graph 3-2. Now we begin by checking whether the last production element in the production lists has its production end date before the desired end date of the order. If this is not the case we do the same with the next machine. If this would be the case we compute the possible production days that lie in the time span between the end date of the last production element in the list and the due date of the order. We will get a specific amount of time that we might be able to produce on this machine to fulfill the end date of the order. With this amount of time we can now compute the amount of elements that we might produce. So we notice that amount of elements that we get in the computations result. With that result we compute the remaining amount of elements that we have to produce for the order. This is done by subtracting the computed elements from the total amount of elements.

$$ComputedElementsForMachine = \frac{RemainingProductionTimeTillDueDate}{ProductionTimePerElement}$$

$$RemainingElements = TotalElements - ComputedElementsForMachine$$

With those remaining elements we proceed computing on the next machine and so on. If we finish the preferred machine list seen in Graph 3-2 and we have ***RemainingElements > 0*** we have a problem. In this case we are not able to split the order on different machines and we have the case that we have to notify the contractual authority that we do not manage to produce the order on time.

The disadvantage of splitting the order on more than one production machine is that we do not have necessary following points that we tried to achieve in section 3.3.1 and 3.3.2:

- Minimized production costs
- Minimized production time

But this does not necessarily results in a bad solution, which means not being able to deliver on time. So we do consider a lower production rate resulting in higher production costs to satisfy the contracting authority.

	Hours Needed	Start Time Stamp	End Time Stamp	Due Date	Hours to due Date	Production On Time
New Production	20			25		

Table 3-6 The new production

	Hours Needed	Start Time Stamp	End Time Stamp	Due Date	Hours to due Date	Production On Time
Production A	5	0	5	10	5	True
Production B	6	5	8	9	1	True
Production C	2	8	10	11	1	True
Production D	8	10	18	19	1	True
New split production	7	18	25	25	0	True

Table 3-7 After appending the new production on machine 1

	Hours Needed	Start Time Stamp	End Time Stamp	Due Date	Hours to due Date	Production On Time
Production A	3	0	3	10	7	True
Production B	3	5	8	9	1	True
New split production	13	8	21	25	4	True

Table 3-8 After appending the new production on machine 2

3.4.Position Selection

As seen in 3.3.4.2 we can move a production in the production list.
We can decide to move a production in the production list as we can see in the following two scenarios:

- Move them if we cannot append it on the end of production chain as seen in 3.3.4.2.
- Move them immediately

Because we have already discussed the first point in 3.3.4.2 we now discuss the second point.
We can try to optimize the production list in such a way that the sum of all time differences of each production item in the production list until to the due date of every

single production item is maximized. If we maximize that we have ensured that on every position in the production chain we have enough time left for the case that if some unexpected error occurs we will always be able to deliver on time.
In order to be sure we are anytime in such a situation we first have to search the latest possible productions start date. We find that date by subtracting the total production time from the due date with consideration of the machines daily production time and the weekends where we do not produce. After that we have calculated the latest possible production start date we search in the production list the first order that starts closest prior the just found date. So we try inserting the production prior that other production and check whether we affect the backward-shifted production. If so we roll back and try on another machine. If we are in the lucky case that we do not affect that production we sum up all time spans that are after each element until to the due date.

$$\sum DueDate - ProductionEndDate$$

The next step we do is to shift the new production one position toward the start of the list. Doing that we have to check another time whether we affect the newly backward-shifted order or not. If not we sum up all time spans that are after each element until to the due date again. If that sum is bigger than the previous computed we remind this result otherwise the old one.
At that time we have traversed the whole production list we take the position where we have the maximal sum of remaining time until the due date as position where we insert the new selected order.

	Hours Needed	Start Time Stamp	End Time Stamp	Due Date	Hours to due Date	Production On Time
New Production	12			13		

Table 3-9 The new production

	Hours Needed	Start Time Stamp	End Time Stamp	Due Date	Hours to due Date	Production On Time
New split Production	7	0	7	12	5	True
Production A	3	7	10	10	7	True
Production B	3	5	8	9	1	True

Table 3-10 After splitting a part of the new production on Machine 1

	Hours Needed	Start Time Stamp	End Time Stamp	Due Date	Hours to due Date	Production On Time
New split	5	0	5	12	7	True

Production						
Production A	5	5	10	10	0	True
Production B	6	5	8	9	1	True
Production C	2	8	10	11	1	True
Production D	8	10	18	19	1	True

Table 3-11 After splitting a part of the new production on Machine 1

3.5.Expand a production

Until now we have seen following solutions to optimize the production queue:

- Insert on end of production queue
- Insert on every other position in the production queue to optimize the time window until the end of production

Now we have seen many solutions to optimize the production queue but we have left over a very important point. Until now we do not have considered that we might have another production in any machine that produces the same element that we try to insert into any machines production queue. Think about that. Would it not be nice if we try to expand this production, if possible with the new one?
Yes, I think so. Now, let me explain why this would be a possible solution and the steps to that solution.
You have to consider following situation. If you produce any element on a machine we have seen previously that it takes some production time. That is clear. But what we have not considered until now is that after finishing producing an element we need another specific amount of time to prepare that machine for the next element. In general this is not done in minutes but hours or days. Preparing a machine for the next production means that we have to clean it if the material of the element differs from the material of the element that was produced lately. I.e. if the last production was produced of stainless steel and the new production element of copper, we have to clean all stainless steel shavings to not mix them with copper shavings. Why not? Well it is easy to explain. Shavings are sold to the shelving-dealer and therefore they should be separated. Another reason is that some elements are made of silver, and therefore we have to accurate take all shaving out of the machine because of the value of silver.

3.5.1. Try expanding production with same item

At the time we insert a new item in the production list we consider that the list on every machine is already optimized, because we have applied the shown steps any time before. So if we want to expand or merge a production that already exists in the production list first we have to find such a production in any of the possible machines. In order to find existing production in any machines production list we have to traverse

each production list of every machine and search for the ID of the production element. If we do not find any production with the same element-ID we cannot expand any production on any machine. If we find one or many productions on a machine they are either in running or waiting state. Saying that in other words means that the production is already in production or in queue. Not being in production means that another production is going on and the production waits until the other production is processed or all previous production have been completed. If there is such a production in the waiting queue or already processing we can try if by adding the new production to that found production whether we affect other productions that are after that one in such a case that they could not be finished on time. If we do not affect any other production we can easily expand it with the desired item count. Clearly this is done after a previous check whether by expanding the production we can deliver on time. If not we check on another machine. We check also on another machine if we cannot expand the production because the problem mentioned above that we influence another production.

	Hours Needed	Start Time Stamp	End Time Stamp	Due Date	Hours to due Date	Production On Time
New Production of type B	8			20		

Table 3-12 The new production

	Hours Needed	Start Time Stamp	End Time Stamp	Due Date	Hours to due Date	Production On Time
Production A	5	5	10	10	0	True
Production B	6	5	8	19	11	True
Expand B	8	8	16	20	4	True
Production C	2	16	18	21	3	True
Production D	8	18	26	39	13	True

Table 3-13 After expanding the new production on a machine

3.5.2. Expand by splitting

As we have seen in chapter 3.3.4.3 we can split a production in such a way that we can produce an order in parallel. The difference to chapter 3.3.4.3 is that there we tried to insert a new production item to the list and now we try to split and expand already existing productions in waiting or producing state. This step is applied if the step in chapter 3.5.1 fails and we do not find any production that can be fully expanded by the new order.

We now apply almost the same as we have already seen in chapter 3.3.4.3. We first try to find if there is any machine where there is a production with the same Element-ID. If

there is none we cannot continue and we have to insert a new production. If we find at least two (otherwise we cannot split) we can determine whether we are able to split or not. In order to calculate this we take the first machine where we could produce and try to insert as many elements of the new production as we could produce on time. Then we calculate the remaining one like we have seen in 3.3.4.3 and try to insert the remaining count of elements in the remaining machines. If we could insert all the preferred element count, which are needed for the whole production, we succeeded. If not we could not split-expand the order on any machine. So we fail producing the order on time.

	Hours Needed	Order income Time Stamp	End Time Stamp	Due Date	Hours to due Date	Production On Time
New Production of type B	9	7	16	15	-1	False

Table 3-14 The new production

	Hours Needed	Start Time Stamp	End Time Stamp	Due Date	Hours to due Date	Production On Time
Production A	5	5	10	10	0	True
Production B	6	5	8	19	11	True
Expand B	7	8	15	0	4	True
Production C	2	16	18	21	3	True
Production D	8	18	26	39	13	True

Table 3-15 After expanding a part of the new production on machine 1

	Hours Needed	Start Time Stamp	End Time Stamp	Due Date	Hours to due Date	Production On Time
Production A	5	5	10	10	0	True
Production C	2	10	12	21	3	True
Production B	6	12	14	19	11	True
Expand B	2	8	10	15	5	True
Production D	8	10	18	20	2	True

Table 3-16 After expanding a part of the new production on machine 1

4. Evaluation

In order to prove that our implemented code and algorithm works as expected we have to do some comparison with situations that we had before and situations that we do have now.

4.1.What do we compare?

In order to have a good result it is important that we compare essential factors. What we used as comparison is:

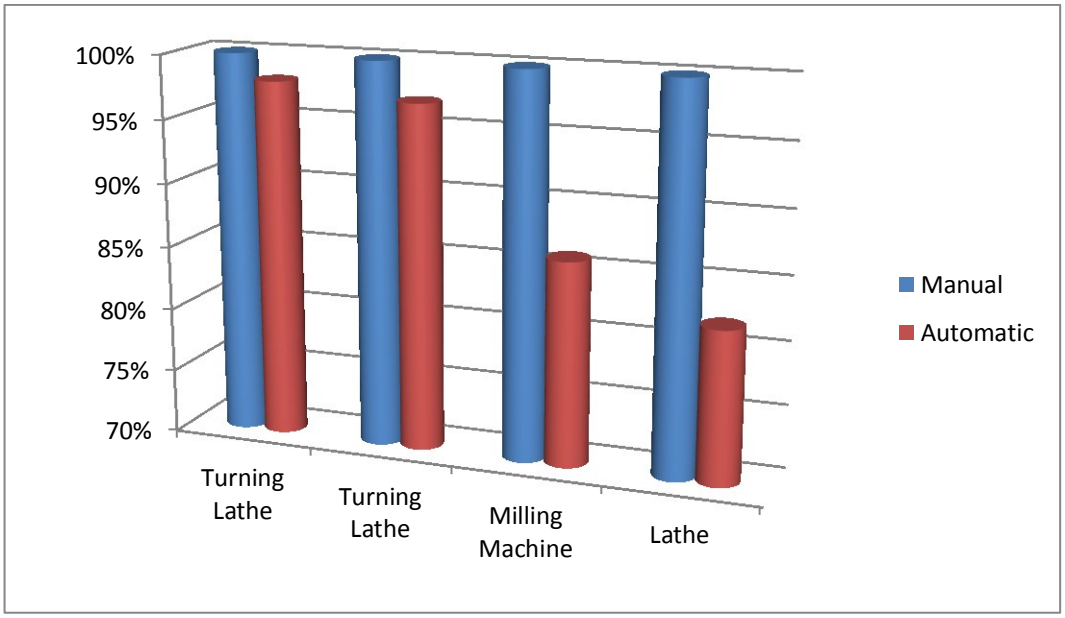
4.1.1. Time needed for compiling the production plan

A very important factor for the user besides the correct working and trustful algorithm is that compiling the production plan by inserting new orders takes less time than it does before. From experience that we collected during the development of the application we can say that improving the application response time does affect the most user satisfaction. The user will immediately notify that a step that finishes one second earlier than the last time the user uses the program. Therefore it must be guaranteed that compiling the production plan by hand should be much slower than selecting the order and letting the program decide where to put in production queue. In order to monitor the time needed for manual compilation I implemented a stopwatch that measures the elapsed time from when the order has been selected until when the production plan gets saved. Because I run this monitoring almost half of a year I think that I have enough points to evaluate. In order to evaluate the time needed for inserting automatically by program I started and ended the stopwatch in the same time as for manual compilation. As expected the automatic assign takes less time. The manual insertion takes at average 34 seconds where the automatic insertion takes in average 136 milliseconds. Surprised? No, I would not. It is a machine computation that does the same that the user does but a lot faster and more accurate.

4.1.2. Production costs for production

The next step I compared is the production cost. By not taken into consideration of the time needed for compiling the production plan we compare now the costs that were needed for a single order that goes into production. Inserting manual the user in general knows the fixed costs of each single machine. Otherwise he can look up. Additional he also exactly knows how many waste material each machine produces. Therefore he can insert and move the production element until he has found a place where the production element in his opinion fits best.

For this comparison I have monitored the costs for each production. I have monitored the costs each time the production was completely produced. Doing so I got following result:

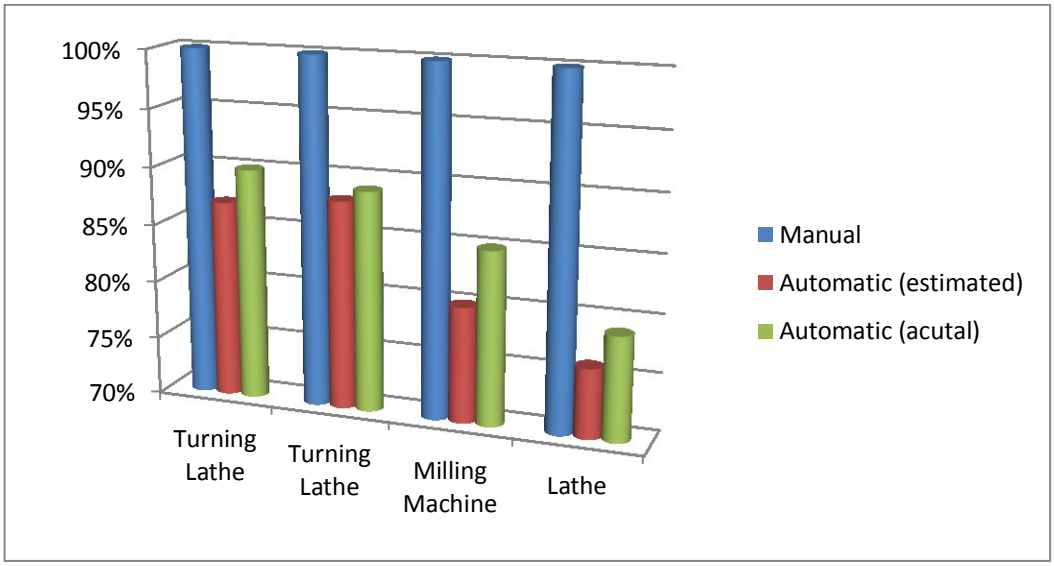


Graph 4-1 Evaluation of the production costs

As you can easily see the automatic process costs less than the manual. The reason why this difference is not so huge is because the user planned as accurate he is able to and therefore the costs were already low.

4.1.3. Production time for production

Monitoring the production time was also an important factor because applying the algorithm and all my solution steps should at least lower down the production time. If so we have more slots available to produce further orders. I compared for each, manual and automatic, the predicted and actual production time that was used to produce a single element. This was done by computing the real total production time of the production and dividing it through the count of the elements that we had to produce.



Graph 4-2 Evaluation of the production time

In this graph we can see that we improve the production time. We can see very well that on a manual operated machine this improvement is much better than on an automatic machine. This is because an improper scheduling on a manually used machine impacts much more than on an automatic used machine. We can also see that the real value of needed time is always higher than the estimated time. This difference between estimated and actual production time is bigger on manual operated machined because there could be more human failures than on an automatic one. For future work I can build in some threshold that the algorithm adds in order to get better real results.

4.1.4. Being able to produce

A next step in the comparison is that we consider the ability to be able to produce on time. I count all production that did not succeed and got a factor of 2% meaning that 2 out of 100 productions did not finish on time.

5. Conclusion and Furture Work

This thesis introduced an implementation of a scheduling algorithm with all the requirements that the company gave. It shows that scheduling a production plan for a great amount of machines is very time consuming if it is done by hand. During the development of the algorithm we have seen that there are so many factors that we have to take into account. The main focus of the algorithm was to assign a new order to a machine that already has a very complex and optimized production plan. We were not allowed to change, by assigning a new production item to a machine, the existing production plan if we affect another production in such a way that it will not finished on time. Otherwise we may change the order.

We have seen many possibilities to assign a new order to an existing production plan in order to ensure that order finishes until to the due date. Additional we have seen how to ensure that the production costs and the production time of the order is as less as possible.

For future work I could include the optimization of the algorithm in such a way that the system stores all previous data about the machine assignment, like previous real production time, real production cost etc. This would give us the possibility to optimize the algorithm in such a way that assigning a new order to a machine we can consider all the decision made in past. Another future work would be that we do consider the down time of a machine in the calculation of the end time point of the production.

6. References

6.1. Bibliography

1. **Brucker, Peter.** *Scheduling Algorithms.* Osnabrück : Springer, 2006. ISBN 978-3-540-69515-8.

2. **Conway, Richard W., Maxwell, William L. and Miller, Louis W.** *Theory of Scheduling.* 2009. ISBN 0486428176.

3. **Lawler, L. Eugene.** Scheduling a Single Machine to Minimize the Number of Late Jobs. *University of California at Berkley.* [Online] October 1983. [Cited: January 15, 2012.] http://www.eecs.berkeley.edu/Pubs/TechRpts/1983/CSD-83-139.pdf.

4. **Mohammadi, E. and M., Haydari.** *Single machine problem with a minimax criteria and preemtion penalties.* Shanghau : s.n., 2011. Print ISBN: 978-1-4244-8727-1 .

5. **Pinedo, Michael.** *Scheduling: Theory, Algorithms, and Systems.* s.l. : Springer, 2008. ISBN 9780387789347.

6. **Sukul, Shailen.** Musings of a Software Architect. *Musings of a Software Architect.* [Online] May 29, 2011. [Cited: December 29, 2011.] http://www.shailen.sukul.org/2011/06/applying-mvvm-principles-to-sharepoint.html.

7. **Barber, Sacha.** Codeplexx. *A Rich Feull Featured WPF/SL MVVM Framework.* [Online] September 06, 2011. [Cited: March 05, 2012.] http://cinch.codeplex.com/.

8. **nblumhardt.** Codeplex. *Managed Extensibilit Framework (MEF).* [Online] December 19, 2011. [Cited: March 05, 2012.] http://mef.codeplex.com/.

9. **Albahari, Joe, Albahari, Ben and O' Reilly Media Inc.** In a nutshell. *LINQ In a nutshell.* [Online] 2012. [Cited: March 05, 2012.] http://www.albahari.com/nutshell/linqkit.aspx.