TAMPEREEN TEKNILLINEN YLIOPISTO
TAMPERE UNIVERSITY OF TECHNOLOGY

HOW TO IMPROVE SOFTWARE ROBOTS' PRODUCTION EFFICIENCY WITH SCHEDULING?

Master thesis

ABSTRACT

Erik Jokela: HOW TO IMPROVE SOFTWARE ROBOTS' PRODUCTION EFFICIENCY WITH SCHEDULING?
Tampere University of Technology
Master of Science Thesis, 47 pages, 3 Appendix pages
March, 2018
Master's Degree Programme in Automation technology
Major: Process control
Examiner: Professor Matti Vilkko

Keywords: RPA, production scheduling

In this thesis it is investigated on how to improve production efficiency on processes that have been automated by using RPA (Robotic Process Automation) solutions. Usage of software robots has been increasing rapidly during last couple of years. Currently, they are operated manually by having process controllers operating and supervising them. Also schedulers has been used. Since the number of robots is increasing rapidly and they require plenty computers or virtual computers it is getting hard for human to operate them efficiently.

This thesis focuses on how different schedulers could be done, and what are the things that has be considered when developing schedulers. Two different approaches were selected, which were selected by their qualities. First one was the earliest due date and the second one was the using cost functions to minimize waiting times of the processes. For testing a three processes environment was used. The first solution worked quite well, however it was switching from process to process quite often, which decreased the efficiency of solution, because of the long setup time. The second option was even better, but as the set up was quite simple the results were quite trivial. Thus, the limitations and problems of this solution are still unknown.

TIIVISTELMÄ

Erik Jokela: KUINKA KEHITTÄÄ SOVELLUSROBOTIN TUOTANNON TEHOKKUUTTA AIKATAULUTUKSELLA?
Tampereen teknillinen yliopisto
Diplomityö, 47 sivua, 3 liitesivua
Maaliskuu, 2018
Automaatiotekniikan diplomi-insinöörin tutkinto-ohjelma
Pääaine: Prosessien hallinta
Tarkastaja: professori Matti Vilkko

Avainsanat: RPA, Tuotannon aikataulutus

Tässä työssä tutkittiin kuinka sovellusrobotiikan keinoin automatisoitujen prosessien tuotannonhoidon tehokkuutta voitaisiin parantaa. Sovellusrobottien käyttö on kasvanut nopeasti viimeisten parin vuoden aikana. Tällä hetkellä robotit aikataulutetaan manuaalisesti siten että prosessien valvojat ajavat niitä. Myös aikatauluja käytetään. Koska robottien määrä kasvaa nopeasti, on robottien ja virtuaalikoneiden määrä, joilla prosesseja ajetaan, kasvanut vauhdilla. Tämän takia prosessien huoltaminen tehokkaasti ihmisvoimin vaikeutuu koko ajan.

Työ keskittyy erilaisten aikataulutusratkaisujen tekemiseen ja siihen mitä asioita pitää ottaa huomioon tämän kaltaista ratkaisua tehtäessä. Työssä käytettiin kahta erilaista aikataulutus ratkaisua, jotka valikoituivat niiden luonteen perusteella. Ensimmäinen ratkaisutapa oli tehdä aikaisimman eräpäivän työ ensin. Toinen ratkaisutapa oli käyttää kustannusfunktiota ja minimoida prosessien aiheuttaman odotuksen kustannusta. Testipenkkinä käytettiin kolmen prosessin ympäristöä, jolla ratkaisua testattiin käytännössä. Ensimmäinen vaihtoehto toimi hyvin, mutta ratkaisu vaihteli prosessien välillä turhan usein, mikä heikensi tehokkuutta, koska prosessin ylösajoaika on kohtuullisen pitkä. Toinen vaihtoehto toimi hieman paremmin, mutta koska testijärjestelmä oli liian yksinkertainen, oli jaksotukset melko triviaaleja. Näin ollen ratkaisun todellinen käyttäytyminen jäi vielä osittain hämärän peittoon.

CONTENTS

1. INTRODUCTION

Trying to produce more value with less resources leading to profit has been one of the key purposes of organizations. Automation has been seen as a solution to reduce the need for human resources for a long time. Moreover, the reduction of human labor may have led to higher productivity and lower costs, hence higher profit. Nowadays automation is not implemented only in the factories, but software robotics has entered to the back-offices in the service industries to work alongside of humans. The big service sector companies such as telecommunication companies, banks, financial services, and hospitals can be divided into two parts if considering the corporate and operational functions within the organization. These parts are the front office and the back office. The front office handles all customer interactions, contributes sales and marketing. The second part, back office, contains all the business functions related to its operations. This kind of business functions contain, for instance, handling applications and paper work, following and enforcing the regulatory compliance, providing the needed human resources, collecting and generating important data and information, being responsible for accounting, providing ICT services and many more. The main difference between these offices is that the front office is visible to the customers whereas the back office is not. (Investopedia 2016)

The software robots, also known as Robotic Process Automation (RPA), is a technology where process logic is taught to computer. After that the robot automatically works like human on computer. The process logic is the pattern and set of regulatory and company rules, which determine how a business case is handled. An example of a process logic would be the following: how an invoice is written to a customer that has bought monthly billed service. The process logic contains also the information of what applications are used in a particular task, which template to open, what information is written and how to generate and gather it, and how the invoice is delivered to the customer. The logic is taught in a similar way as Microsoft Visio's flowcharts are drawn, with a graphical user interface depicting the process flow. The software robots function like macros in Microsoft Word or Excel, but they can within and throughout the whole operating system via the user interfaces like human would do. The robots are generally developed to do repetitive, high volume back-office tasks. The criteria to robotize something is rather close to the criteria of outsourcing that process. The RPA solutions are not going to replace the traditional information systems and traditional IT automation within the information systems, but RPA is complimenting them. The main benefits of RPA are its cost reduction capabilities, reduction in processing and lead times and reduction of simple and boring tasks that human workers do not necessarily want to do. Even if a robot could not work faster than human, it does not need to sleep, eat or have breaks during the day. Additionally, the RPA solutions are scalable. This leads to an increase in the customer satisfaction and in the value experienced by the customer.

Once there are hundreds of processes automated and there are hundreds of resources (computers or virtual computers in a server) where the robots could be run, controlling the RPA entity will get harder and harder. The robots require people to schedule them and supervise the production runs which refers to a single run of a robot or a process in the production environment. Furthermore, when considering that a group of people, called controllers, must know how to schedule and run these hundreds of process and robots in a sensible manner, the situation gets complicated. As the number of controllers grow, they are not able to optimize the production in companywide perspective, but they are rather optimizing their use of time. As partial optimization seldom leads to overall optimization, the utilization rates are poor. Hence, resources have typically lots of downtime and are not used efficiently. The resources are not free, not to forget that the number of licenses, both for RPA-software and all the other software increase with every new resource. Also, someone must configure and install new computers. Some RPA software have schedulers, but it usually makes things worse, since there is very limited intelligence in the scheduler. Also, trying to decide which process should be run first and how many resources are needed, is something that comes challenging after the number of processes increase. This problem is a very similar to when work orders are assigned to different machines in manufacturing industry. This problem is known as a job shop scheduling problem. Job shop scheduling problem has been implemented in various places. (Latecoere et al. 1976; Kim, Park 2003; Pham, Klinkert 2006; Pan et al. 2009; Masoud et al. 2015). This article looks to solutions of those and what kind of variations are required to fit this solution method in those fields and how it could be applied in this case.

The main research question is: Can robot run other robots and is it possible to produce real-time optimization algorithm to get the most value out of the robots?

Secondary research problems are: How to identify the limitations that robot may possess and how to design processes so that processes has as few limitations as possible?

What are the setup costs and times that are generated by changing process?

How to determine the revenue of single case run by the robot and how the expected revenue changes if the case have to wait?

How to model the customer value for different products simply but effetely in a function of time?

Could same optimizing principles use in software robots' workflow control as in manufacturing industry?

Could optimization program bring extra stability to autonomous processes or will it cause more problems?

How optimization has been approached in other industries and are there similarities with this industry?

To optimize production, coding is required. This could be done by any language but MATLAB was chosen to this research, since it has possibility create simulations easily to test the optimization code. Also, the online usage possibility made it suitable for this purpose. Different parameters and setups will be tested with Simulink simulations. After different codes have been generated, Blue Prism robot is taught to control and read the results from MATLAB. After this has been verified as a working solution it can be tested with simple real-life system to investigate the actual processing times. However, if this was to make a working product, a proper program should be done, that is capable to calculate faster and to make it more user friendly. To this research identifying correct optimization methods and finding the parameters this is more suitable. However, since Blue Prism works on .NET, C# or VB would be easiest to integrate in a robot.

The tools to optimization will be taken from manufacturing industry and the results will be compared against simple scheduling and other basic managements rule, in this case FIFO (first in first out) and more complex manner by using cost functions.

To fully understand RPA, interviews and literature will be used. Interviews are conducted to workers of company y and consulting firm a, who are working with the RPA. Only limited number of interviews can be performed as there are only couple people who have sufficient information and understanding within the practical level how things are done. Interviews are conducted by the principles of Irvine Seidman's book: *Interviewing as qualitative research: a guide for researchers in education and the social sciences, 3rd edition.*

Articles have been found from Scopus and Science Direct. Optimization articles have been taken from different fields and time periods to get understand what kind of limitations can and could be used. Key words that were used are: "Job shop scheduling", "case". For RPA, there weren't that many scientific articles to use. However, the few that were found, found with search word "Robotic Process Automation". Along those Blue Prism's, UI path's, Accenture's, Deloitte's, CGI's and Digital Workforce Nordic's websites were used to gather additional information. This is due to the fact that this area has not been studied a lot in scientific papers, at least not yet, but IT-companies have experience and information available from this area.

2. RESEARCH METHODS

This research can be divided into two parts: first Robotic Process Automations (RPA) and its distinct aspects are introduced. In the first part the theoretical view on optimization is also studied. In the second part the theory of optimization is applied on RPA and furthermore a case study of RPA efficiency optimization is done.

Since Robotic Process Automation is a relatively new technology and field of IT, there exist a relatively small amount of scientific literature. Therefore, RPA is introduced and studied from a more pragmatic and practical point of view, meaning that case study was chosen to be the most suitable research method. On the other hand, optimization has been a subject of great interest and it has been studied for decades. Hence a lot of relevant scientific literature and case studies from the different sectors of industry are available. Whereas this study is otherwise qualitative, the case study of RPA optimization is quantitative.

The scientific literature utilized in this study was collected from scientific databases such as Science Direct and Scopus. Since optimization is a field of study that is based on mathematical facts, the older literature is mainly still valid. Hence it was not considered necessary to limit the sample based on the publication date. Still some conditions were set when collecting the literature: first of all conference papers and other unpublished or otherwise non-white papers were excluded from this study. The reason for this is that only commonly accepted, scientific views were considered to deliver adequate quality for this study. Secondly the amount of quotation was utilized when choosing the pieces of literature.

2.1 Used software

To make this optimization two software were used. First one was MATLAB version R2015b. This version was selected as it was most familiar version to the writer. The version number should not be relevant to this experiment as the fundamentals of calculation have not been changed as specific toolboxes have not been used.

The second one was Blue Prism 4.5. Blue Prism version was selected because it was used by company with who the experiments were conducted. However, there is new version 5 which was more popular at the end of this research and version 6 has been announced, and early access version has been published. Version 5 works similarly as 4.5 but from this research point of view nothing important other than the user interface changed.

2.2 Interviews

To fully understand RPA, interviews are used. Interviews have been selected as there is not much scientific text written from RPA. Interviews are conducted to workers of company y and consulting firm a, who are working with the RPA. Interviews are conducted by the principles of Irvine Seidman's book: *Interviewing as qualitative research: a guide for researchers in education and the social sciences, 3rd edition*

From interviews, it was wanted to know following questions:

1. How are the jobs scheduled now?
2. How scalable is the current setup?
3. Are the process run on time?
4. What types of jobs there are?
5. How time-consuming scheduling is?
6. What kind of problems have been faced?
7. How predictable coming tasks are?
8. What type of processes are easy and hard to schedule?
9. Criteria for success?

The interviews were conducted as solo interviews face to face. Interviews were short so that the focus would keep on topic. Interviews were selected over survey as there are only handful of people available with sufficient skills and knowledge. With interviews there is also possibility to ask for clarification, but questions were laid quite open so that the interviewees would give their own opinions and experiences and not the conductors opinion.

Questions were conducted to three people who have experiences in process controlling, even though some of them also had some experience in development. People were selected from two companies. Company A, the company has plenty of processes automated and has in practice experiences in controlling RPA processes. Company B, is a company focused purely in RPA consulting and development with 20 consultants. Questions are listed in appendix A.

2.3 Literature

Articles have been found from Scopus and Science Direct. Optimization articles have been taken from different fields and time periods to get understand what kind of limitations can and could be used. Key words that were used are: "Job shop scheduling", "case".

For RPA, there weren't that many scientific articles to use. However, the few that were found, found with search word "Robotic Process Automation". Along those Blue

Prism's, UI path's, Accenture's, Deloitte's, CGI's and Digital Workforce Nordic's websites were used to get additional information. This is due to the fact that this area has not been studied a lot in scientific papers, at least not yet, but IT-companies have experience and information available from this area.

3. ROBOTIC PROCESS AUTOMATION

Robotic process automation is a relative new application in the business usage, but the concept has been around from the 1960s. The technology has been used in test automation, to test software for bugs. Between 1960s and nowadays different screen scrapping technologies have been used, but screen scrapping is not robust in large scale. Screen scrapping is typically executed by having screen shots of the buttons that must be pressed or simple coordinates to that button. The idea has been that the program will search for picture within the screen and as it finds it is able to click on its coordinates. As the user interfaces are typically unstable and have quite lot of bugs, this method was unreliable. Also, the calculation power needed to find pictures from screen and with reasonable tolerances is huge. (HfS Research, 2012)

RPA-software does not require own application programming interfaces (API) but it utilizes the same user interfaces as a human would do. The difference is that instead of picture, RPA-software sniff the code and identifies elements from it. The snigging can be done by different methods depending on the application type. API's are ways to interact with software that must typically be implemented to the program as it is made. During this chapter a closer look is taken how software robot is developed and how it works and what can be automated and what cannot to automated and why. This chapter focuses on the ideas behind Blue Prism software, but the points and practices are applicable in most part for majority of the RPA-software on the market.

The rise of RPA -software has started from fear that all jobs will be moved to offshore location. It has been described as very profitable and there are many articles about it. For example, HfS Reasearch states that full time employee (FTE) in an onshore location costs 80 thousand dollars a year, an offshore worker 30 thousand dollars and RPA-solution costs 15 thousand or less and it does not require training and management as on- and offshore workers would. (HfS Reasearch, 2012) Harvard Business Review tells an example from O2 who deployed 160 robots working 400 000-500 000 task monthly and they have yielded 650 % return on their investment with 4-member team. Other large UK based company has 2-member team with 300 robots performing 600 people jobs, generating 200% annual return. (Harvard Business Review 2015)

The main reason for this kind of technology to emerge has been the fact that calculation speed has been increasing according to Moore's law. (Moore, 1998) As computers can process calculations faster, the code does not have to be as effective. This enables technologies that are easy to use and require little it skills to be competitive option for expensive software integrations, even though they are not as effective. Other success factor has been that big companies are very strict with their it polices and governance systems. Compliance and regulations have been tightening and developing new software is slow and very expensive. RPA has been a way to get around those things and development has been done with much lighter governance model.

During this chapter, a closer look is taken on how a software robot is developed, how it works and what can and cannot be automated and why. Hence this chapter focuses on the ideas behind Blue Prism software, but the points and practices are applicable for most RPA software on the market.

3.1 Robotics in general

RPA is a tool that can be used in many places. There are many companies providing different kind of tools that help to identify potential business processes that could be automated. RPA software is good at reading, writing and storing data and doing calculations and parsing of the data. The main technical requirement to automate any given business process is is that all data needed is in digital form. And it will help a lot if that data is in structured and in a form that is easy to read. In other words pictures are not good. There are methods that can be used to handle pictures, for example OCR. OCR is unrelated technology that is quite often packaged inside RPA software as there are open source projects that give away OCR tools. Most known one is the Tesseract, developed by originally HP and after that supporter by Google (GITHUB 2018). OCR is basically neural system that is able to identify letters from picture. Also unstructured data can sometimes be parsed with regular expression or code. However getting good results with this kind of methods is usually more expensive and requires more knowledge and skill.

Automation products have lot of differences. Also they have been updating quite a lot during last couple of years, as their usage has grown. Their growth is a part of bigger low-code trend (Richardson 2014). Low-Coding is as concept of doing typical IT work without coding, but rather drawing graphical process charts and giving them attributes and properties and finally connecting them with lines. When comparing market leaders, Automation Anywhere, UIPath, Blue Prism and WorkFusion. They differ especially what type of automation they are intended. Automation Anywhere is mainly suitable at building small macros that are enabling user to work efficiently. However it is not suitable for large scale server infrastructure as it is not providing case monitoring (queue system or Orchestrator). Blue Prism and UIPath are providing systems to allocate working to server and they are suitable for heavy weight backend automation. UIPath is maybe slightly lighter automation tool but requires more coding and Blue Prism is following more closely the low-coding concept and has easier scheduling system. WorkFusion is quite new player and their product is not done when writing this text. However it is estimated that it could be a product that would be a game changer and would bring lot of new tools to really do enterprise level automation.

The main RPA products are to attach to different kind of applications. Mainly applications that are web-, .NET- or java -based software, but they are working also with terminal or emulator based software but the integration has to be done by coding or by using select all text and copy method. If there is an application that does not support hooking that is provided by the RPA product or the software is running through virtual computer, surface automation can be used. Surface automation is basically saving pictures of the elements that are wanted to be used. When the process is running there are algorithms build inside of the products that are able to seek from the screen if it has corresponding element that the picture has. This however requires that the element is always on top of the screen. This method works with every application but it is slower and not as robust. The quality of picture recognition varies quite a lot between RPA products and many products are working to improving their surface automation algorithms.

In business sense the process can be anything. In general RPA is a suitable tool for processes that are highly regulated or are rule based. The processes can consist of different types of tasks, reading information from various systems, combining data, comparing data, saving data, modifying data and presenting data. As discussed earlier the main business processes that have been automated by various companies have been financial management processes. Also back office processes, which require integrating lots of information from multiple systems, have been popular automation targets. Three reasons for automating business have been presented by many customers. The main reason has been to save work to stop new recruits or to do organizational changes and shift people to do new job or to lay off people. Second reason has been to prevent human errors, and the third one has been to decrease time that has been used to do that task. On addition one of the key benefits is that RPA is that it is scaling quite well. This means that if the company knows that it is able to scale its processes, it might be more willing to expand its business. Automated processes also scale down, so it is not as expensive to stop some segments of business as it might be with human workers. Befits of automation are quite similar as would be with any IT projects.

Before RPA the processes are almost always done manually. There can also be a new process that has not been done prior to automating it. The main measurement of feasibility is that how many people that task requires at the moment and how many people it would take to automate. The main principle is that it would have to be able to pay back in 1-3 years in order to be profitable. As the library of automated software increase, at least in theory the time of automating next process should decrease.

Most companies are using task manager systems to distribute the cases to workers. Some companies or teams might use some simpler was as tagging emails but the principle is same. RPA products are trying to work as ask manager products but at least so far it has been better to use other task manager products and use RPA products to get task from those programs and save them to RPA queue and then fill back to the system.

However at least WorkFusion has been trying to enter that space, but it success will be seen in the future. Hence currently the RPA requires someone to supervise that the robot is taking cases from task management systems and filling them back once completed. This person is commonly called as controller. The controller has to also make sure that cases run through the RPA product so that the cases are done on time.

3.2 Robotics in practice

The RPA software works by designing flowcharts. In Blue Prism there are two basic layers of designing and queuing system. RPA is a tool that can automate human made business processes quickly without need to change the old programs. Developing RPA processes itself require little programming skills, however to make processes as fast and as robust as possible, some coding skill will be required. As RPA-processes are relatively cheap to make as they do not need big IT infrastructure changes or as skilled IT workers. Thus, it very attractive to businesses. (HfS Reasearch, 2012)

3.2.1 Process layer

There is a process layer that is essentially a model of the business logic. Business logic is the actual high-level description of a business process that organization follows to deliver value. In this layer the core logic of RPA is generated. There can and should also be multiple process layers, especially if the automated process is long and multiple software is used. This layer is purely visual programming. However, the decision and the calculation boxes contain wanted arguments. The language used is close but simpler to visual basics.

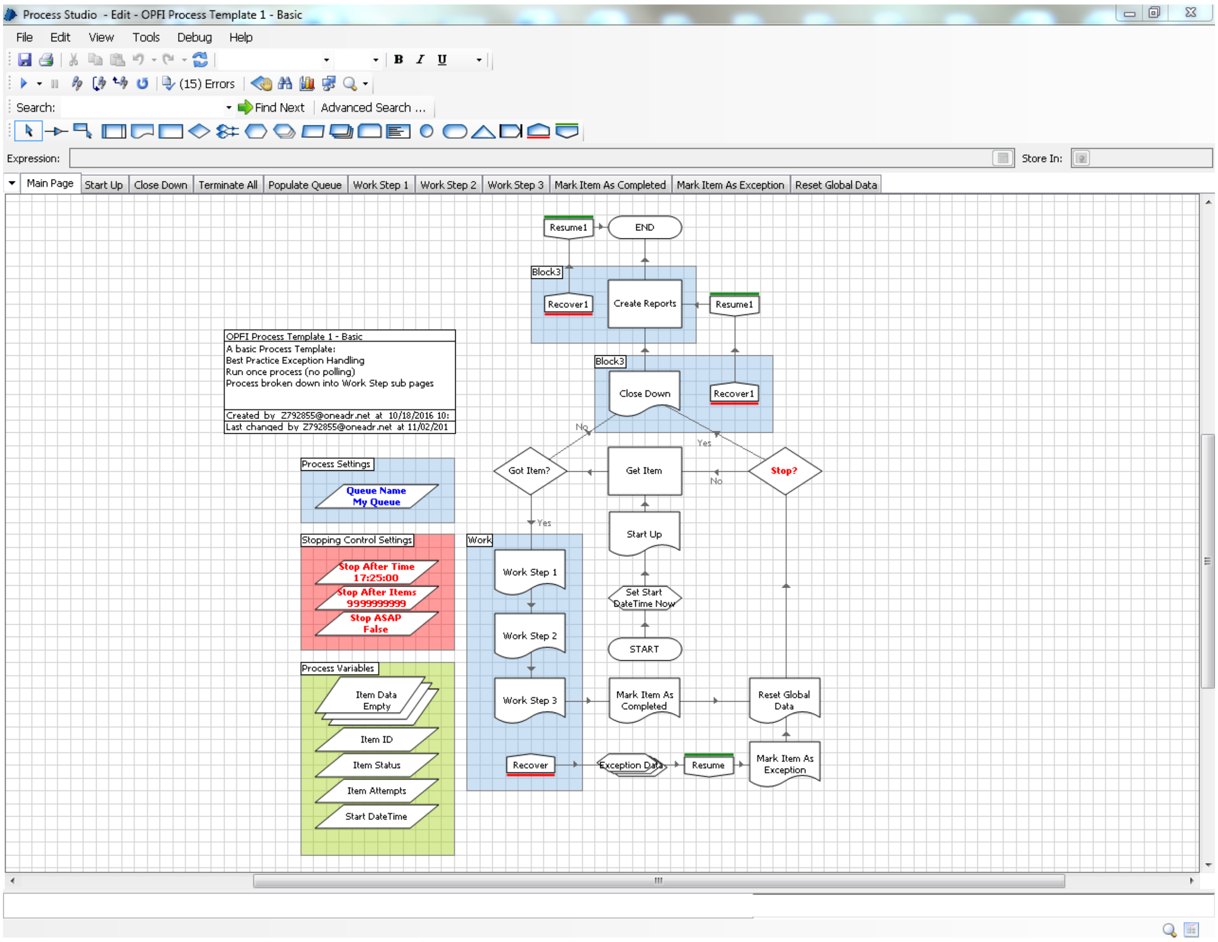

Figure 1. The Main Process Layer

In the figure 1. The Main process logic can be seen. In this example the start is in the middle. From the start the process continues to start up-page, that sub-process stars up required programs. Next is get item that access to the process queue and takes the next pending item. After item is taken the processing parts starts. The item can go to exception for either if system fails or the case is out of business scope, this moves the process to recovery stage. Both marking item as a complete or exception saves item to respected collection. Once the loop ends the process goes to close down, all programs are closed. At the end data is written to excel sheets to be passed to workers.

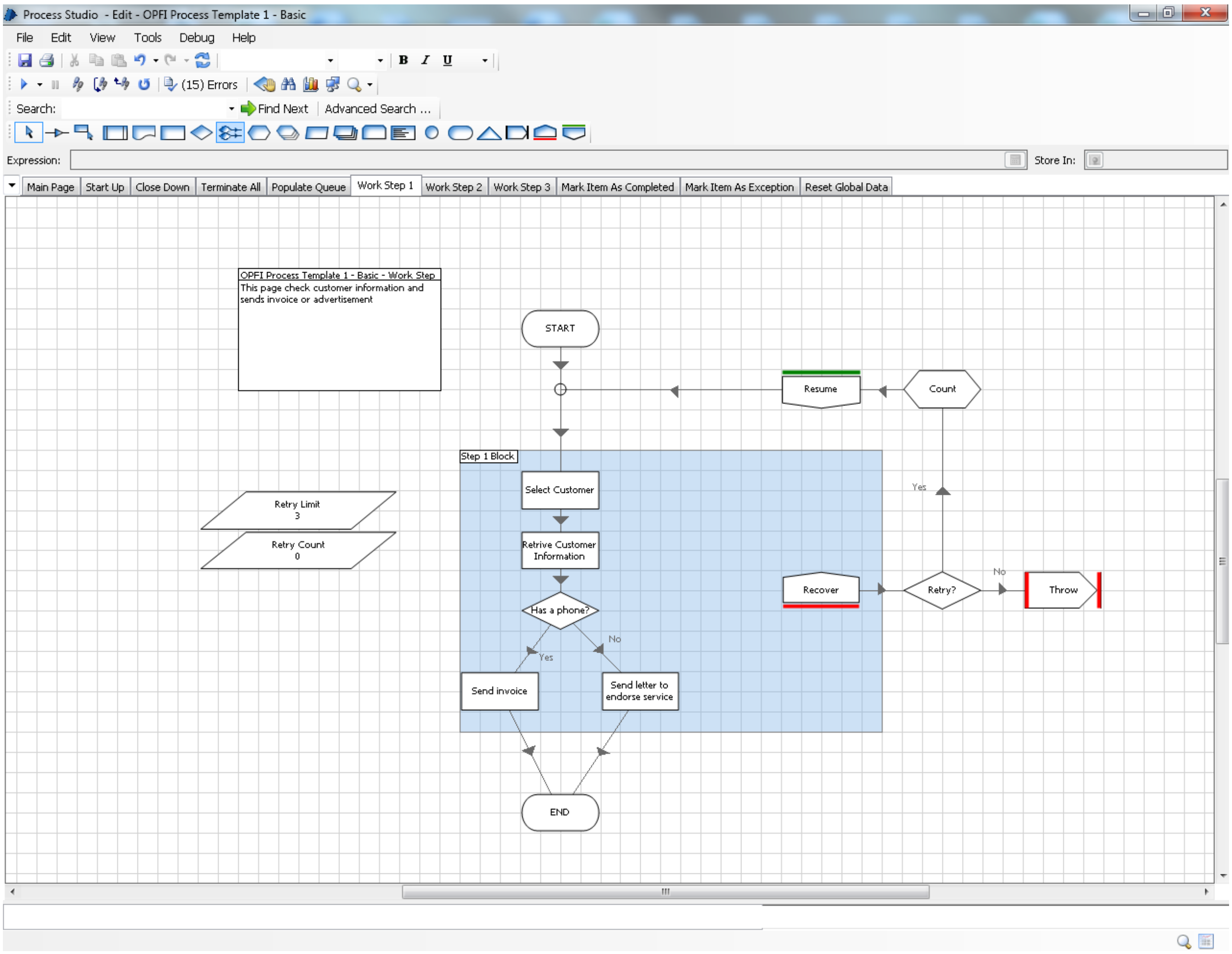

Figure 2. Business Logic Layer

Figure 2 is an example of how business logic could look like. At the begin the process selects customer and retrieves customer information. After that it makes decision if the customer has a phone. And if he has the process sends invoice and if not the process sends advertisements. Business logic also includes exception handling logic.

3.2.2 Object layer

The deeper level object layer contains more detailed commands. The objects are small sub parts of the process. The process level calls these objects, so they are like functions or libraries in programming. Objects can be called from multiple processes, bringing the scalability and coherence to fully scale the RPA to improve efficiency. They contain the detailed information on how a single task should be completed. Every block is basically a function in .NET programming language and they are executed in the order of flowchart. These blocks can be for example .NET commands such as wait for, sendkey, mouse click, press, write and read text from screen to complex text parsing, calculations and finding and gathering date from data structures. Object layer is shown in figure 3. In the figure object attach to process, MS One Note, checks that main window is open. After that clicks screen, sends home to go upper right corner and pastes clipboard. In this layer both visual programming and object oriented programming can be used. One object layer object should access to one computer application. Applications can run on the background or on the foreground, since Blue Prism goes in to the code of the application through the user interface, so the window do not have to be showing, but running the process is enough. In one process, multiple objects can be used and process and objects can even call other processes.

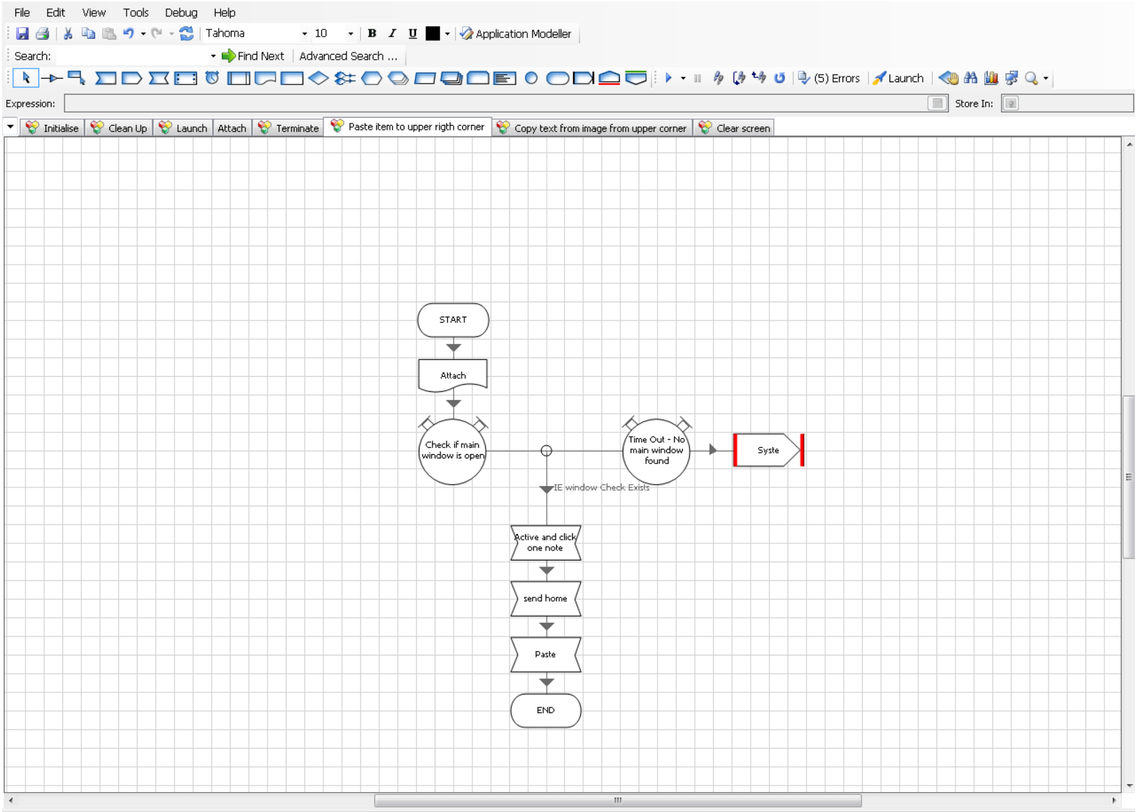

Figure 3. Object Layer

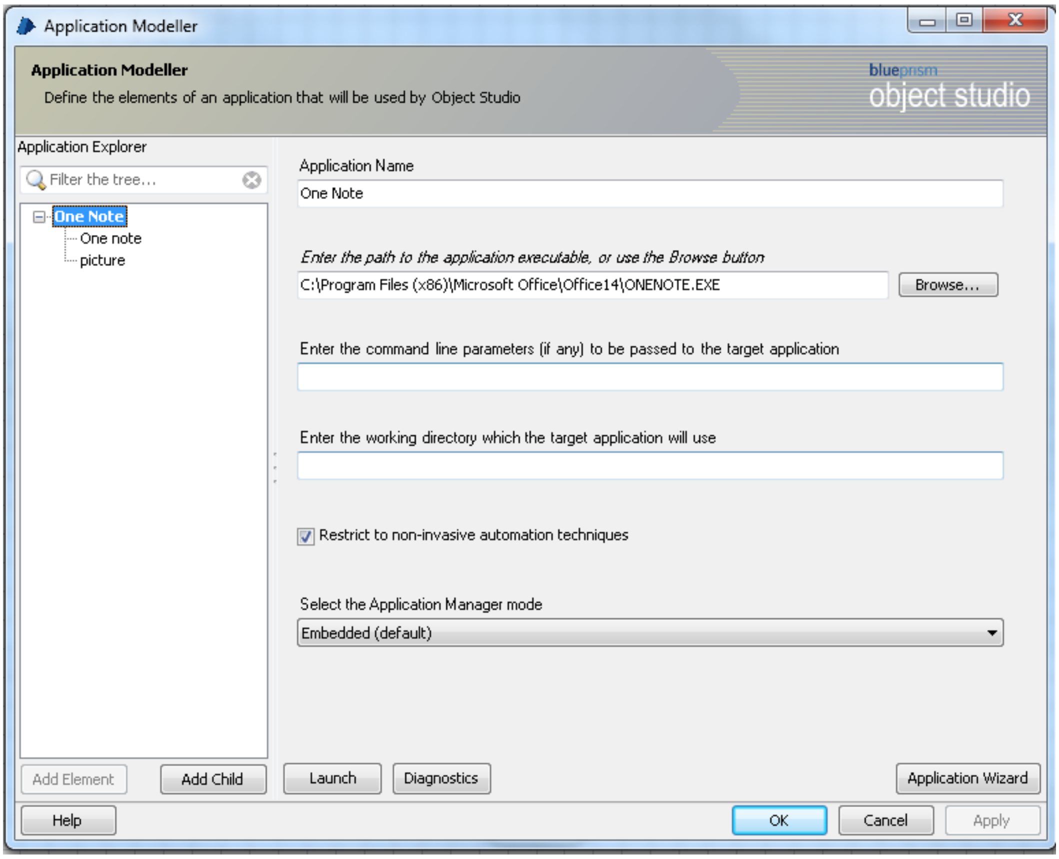

Figure 4. Settings For Identifying Applications

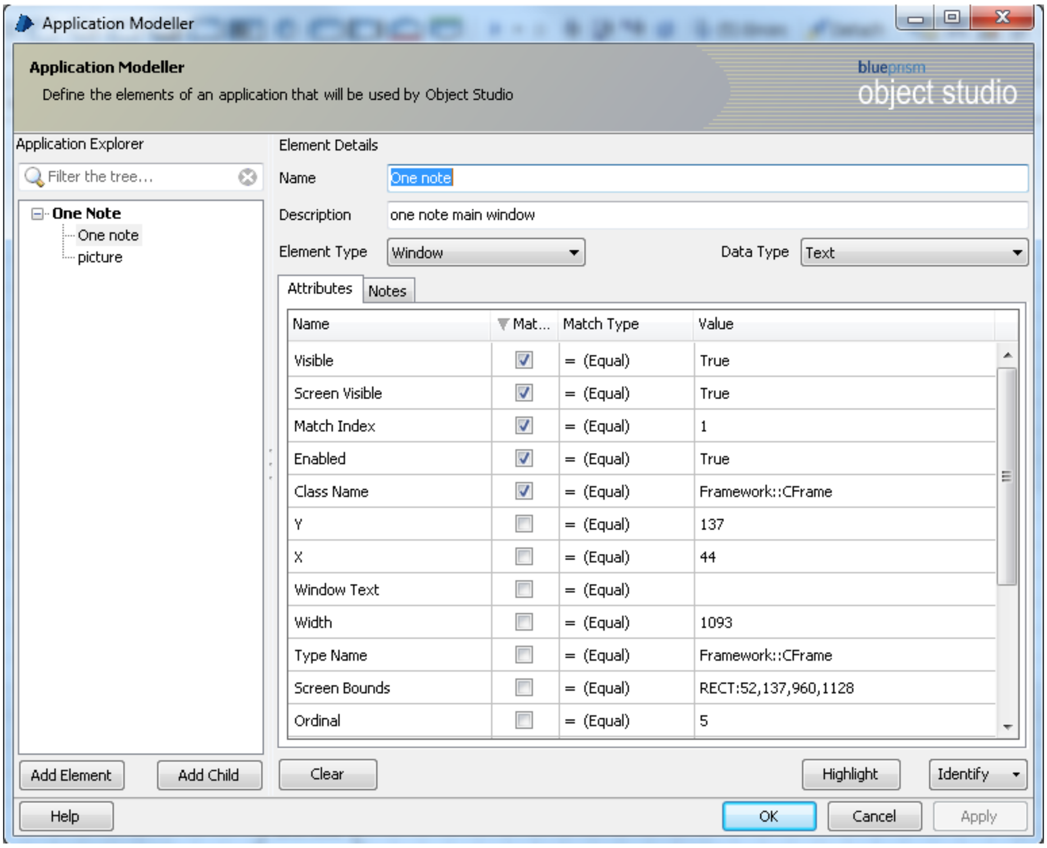

Figure 5. Identified Attributes For Buttons

In figures 4 and 5 the application modeler is shown. This is the tool to identify target applications and different functionalities in them. Selecting correct tags is in important role, for making the process robust and good to scope with changes. Also, some attributes such as match index influences the robots operating speed.

3.2.3 Queue management

RPA-software like Blue Prism work by getting and saving information to the queue. Queues are lists of the cases that should be or have been processed. This information can be used to monitor handled cases and to distribute them if multiple resources are working on same process. From queue, different cases' logs and status can be seen. Items can be in a status of a pending, a completed or as a business or a system exception. The items in the queue can have priorities. Priorities can be used to better manage the order of items to be processed. Queues works as important tool to the business to observe what cases has been running and what cases are still to be run. To follow this information is the main task that the process controller has. The information is used to decide what processes to run next and to calculate how long it takes to complete the process. Queue data can also be used to create reports and statistics about what has been completed.

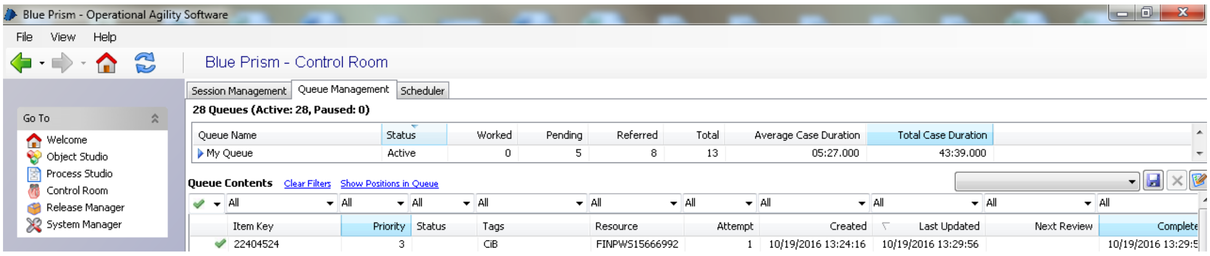

Figure 6. Queue Management

In figure 6 queue management can be seen. We can see that for this queue My Queue, there are 5 pending items. There are also 8 cases that have been failed. Also, average case duration and total working time can be seen. In queue content shows individual items, and their status. Resource, number of attempts and creation and completion times are shown.

To allow multiple computers to run same process queue locks items. Once item has been locked no other process or other resource can retrieve that items data. Queues show what items has been saved to the application server. The basic principle is that every process should have its own queue. Thus, items can be moved from process to process by passing it to different queues.

3.3 Use cases

Dr. Willcocks introduces in his book *Service Automation: Robots and future of work* two diagrams which describe where software robots could be used. (Willcocks, 2016) The first diagram, figure 7, compares cognitive intelligence (CI) against RPA. CI and other artificial intelligences are in the brink of coming to commercial use. They are software that can learn like a human would. For example, IBM's Watson and Google's DeepMind and others have shown promising results. Now when comparing RPA and CI, CI is much more advanced tool. However, using them in commercial usage will be expensive. Thus, it should be used in cases that are complex, for example analyzing patient medical condition. Hence the room for RPA is the cases that are relative easy. Being easy means that the logic is easy to write on paper and there are not many rules how to do something. This kind of operation is for example filling up a monthly invoice to a customer. RPA kicks in when the volumes are high and so that automating it gets worthwhile. CI does not require high volumes to be profitable as the task that are complex. Because the tasks are complex they requiring expensive workers, like doctors, and there is still in a risk to have the wrong conclusion.

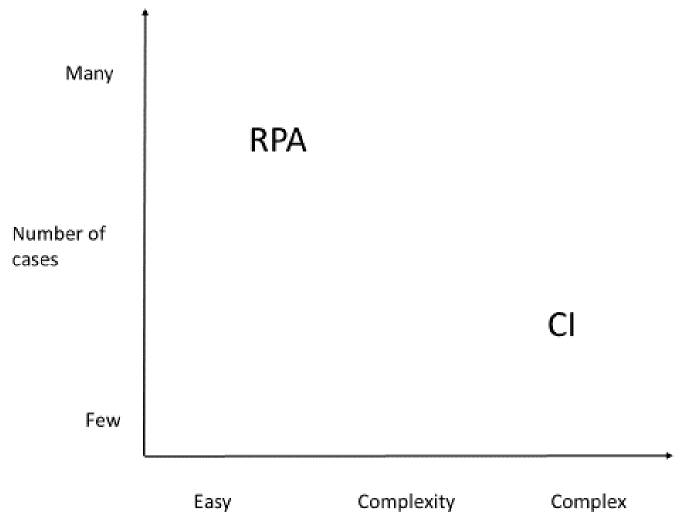

Figure 7. RPA versus Cognitive Intelligence

The second diagram, figure 7, that Dr. Willcocks presents compares RPA to traditional IT. Traditional it should be used on when it investment is big and it the process is it owned. This kind of systems are for example ERP and CMR systems. RPA should be used on the business owned processes that are relative small but automating them with heavy IT would not be profitable and would reserve the IT capacity that could be used on the processes that require IT expertise. RPA processes are smaller in size but the list is long where RPA can and should be used, making it very profitable and giving advantage to companies deploying RPA solutions.

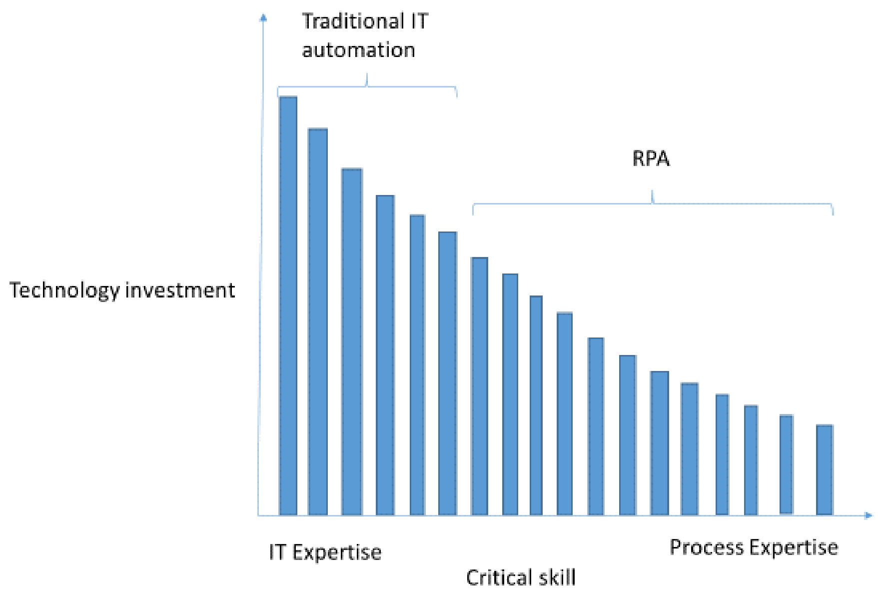

Figure 8. RPA versus Traditional IT

4. PRODUCTION OPTIMIZATION

In this chapter job shop scheduling is explained and some general issues and requirements for optimizing software robots are viewed. Also, some general developing principles are discussed in the subchapter 4.4

4.1 Job Shop Scheduling

Job shop scheduling as an old optimization problem regarding what to make on what machine. The basic rules for the problem is: 1. The next task for a job cannot start before the last one is completed, 2. A machine can only do one task at a time and 3. Task must be finished once stated. (Google Developers 2016) In order to get a result there are couple of different algorithms that have different goals. The most basic approach is to try to make all the jobs as quickly as possible. There can restrictions and one of the most common is that there is a due date for jobs. There can also be set up times involved. The problem is NP-hard, so finding the best answer for many machines with many jobs requires a lot of time. A following notation could be used to represent jobs, Job X = [(M_1,T_1),(M_2,T_2)], where X is the number of job, M_y is the machine number that is needed and T_z is the appropriate time that it takes to complete the job.

A simple example could be to schedule following jobs:

job 1 = [(1, 2), (2, 2)]

job 2 = [(1, 2), (2, 2)]

job 3 = [(3, 3)]

And of the optimal solutions would be:

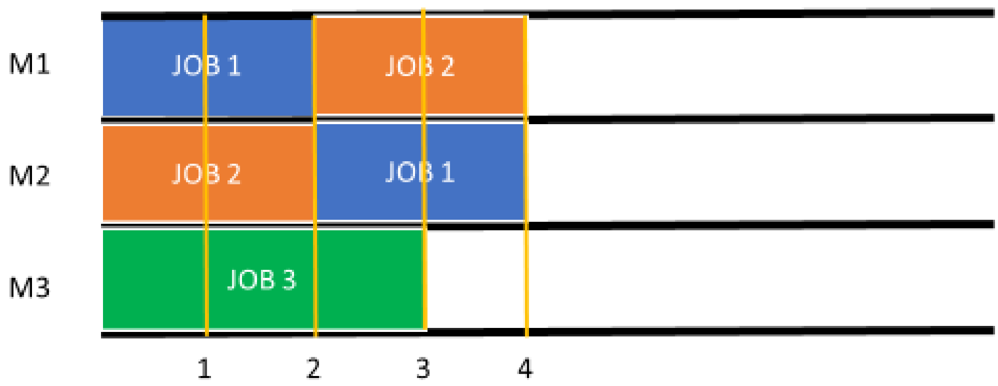

Figure 9. Simple example optimization

4.2 Optimization in other industries

In other fields scheduling is used to improve production. For example, Latecoere describes how in clothing industry there are lot of perturbations in production. Perturbations had three categories. Short term perturbations that only influence one production line for short time. Several period perturbations that affects one line for long time or multiple lines for short time. Waiting stock perturbation to balance items on a storage. By using regulation algorithm allowed quicker and better decision making. (Latecoere et al. 1976)

Kim and Park investigated on how the cranes efficiency could be improved in ports. The problem is hard as the cranes can sometimes block eachothers. Also the task must be conducted in specific order, you cannot fill the ship before it is empty. They came up with two solutions: greedy randomized adaptive search procedure (GRASP) and branch and bound (B&B). B&B was better solution when number of task was smaller and number of cranes was small, but as the number of tasks and cranes increased GRASP was better algorihm. (Kim, Park, 2003)

4.3 Similarities and differences between optimization of software robots and optimization in other industries

There are some differences between software robots and more classical manufacturing scheduling problem. Since software robots can be used for different types of processes the accumulation of jobs can vary. For example, there might be a process to get new passport to customer and the application can be filled at any time in the internet. Thus the order list for this job accumulates with a certain function. Most likely there will be a lot more applications coming at 7 pm than 7 am. However, there are probably a lot of daily variance. The same company could has other process, for example sending monthly bill. In this type job, orders come once in a month in a long list. In company that has multiple automated processes, there are usually mix of this types of order flows and some processes could have both kind of order flows. Hence it could be said that with RPA the diversity of different order flows possibilities is much wider than in traditional industry. In traditional industry there are usually some seasonality or at least economical changes make changes in the quantity of orders. RPA is mostly used in teleoperations, hospitals and banking, however RPA growing else were also, but the process seasonality usually cancels each other's so that the basic order flow is quite stable.

In RPA environment, the whole process can be seen as a job. However, if the process is long and uses multiple software, it can and should be divided to multiple subprocesses and those subprocesses as a task. Designing processes will be discussed further on subchapter 4.4. So the jobs are quite similar, however in software robots case, depending on the process, the processing time might have quite big variance from order to order. Number of jobs can be big in software robots case, since jobs are seldom reusable to other process. The differences can be small and parts of a job can and should be reused on other job.

Machines in RPA environment are most typically identical or at least they should be able to get to be identical. With this in mind, every job should be able to be done on every machine. However, in practice it might be that some computers are slower than others so that processing time might vary from computer to computer. Also some companies can have restrictions with credentials and production rights that might prevent doing some jobs on some machines and some jobs on some machine. For example, there can be a rule that only 2 machines that have rights to handle money and 2 machines have right to check if the person who ordered passport, can have it. From optimization point of view, these kind of company policy roles makes optimization easier, but they can compromise a much more efficient way to handle orders.

In the processes that have been deployed by the companies set up time can be quite long. This is due that the servers are in sleep mode when not run. So, all processes must begin with logging in to windows and running settings and starting software and it typically takes about 5min. Not to forget closing programs and logging of the windows, but if it is correctly done, it is significantly faster, about 30s, thus it can be omitted for this paper. Now, some of the typical processes that have been automated by various companies take usually from 20s to 10min to process one case. So even for the very long 10min process getting 5min set up time every time would be very significant.

4.4 RPA production problems

The basic scheduling problem in RPA is same as the manufacturing industry has had for a long time. There is a bottleneck in the process that slows the handling. Typically, in

RPA the bottleneck comes from that there is a software that only one person can use at the time. For example, some software that have list of possible works can only have one user selecting to take that job. This does not necessary come up with people but as robot is faster, and they cannot talk to each other as easily as people, the risk of two robots taking same task at same time gets high. Also, for example old excel and word documents do not allow two people to edit it simultaneously. Thus, getting those parts to own processes and maximizing efficiency on those processes is critical.

As there might be long processes other kinds of bottlenecks also starts to emerge. For example, there might be subprocess that is pure code stage and take 0.1s per case and other extremely slow web application that takes 15 minutes per customer. It gets even worse if there is limited number of accounts that can do the task and there cannot be simultaneous robots using one account. Especially on processes, where they should be finished relatively soon from the start, taking care that there are not any major bottlenecks some important. Keeping the queues constant size is important to keep the SLA's and to ensure there are not any problems with accounting or other support functions that the company might have.

In manufacturing industry, there are multiple ways to identify bottlenecks. For example, Bert Markgraf has listed four ways in his article (Markgraf, 2016). First one is to check if there are items accumulating on the queue. Second way is to increase capacity of individual parts, if the total output does not change there is bottleneck somewhere else. Third one is to see if some of the processes has high capacity usage. The fourth one is to check wait times. The process after bottleneck must wait for the next item. If a bottleneck is found, the solution can be to redesign the process, split it on parts or to have the dedicated resource to run it one.

4.5 Developing principles from production efficiency point of view

In order to develop robots to be as efficient as possible several notions have to be taken. In this subchapter some guidelines are discussed. However, from other developing point of view these rules might be not as useful. Other developing views could be such as robustness, reusability, memory control or easy to develop for non-technical person. From interviews and experience it came up that these goals are not excluding each other's. In manufacturing industry there is usually decision between flexibility and efficiency but in RPA robustness and speed tend to walk hand on hand.

4.5.1 General problems regarding robot's objects

RPA is often endorsed being easy by just copying the robot to work as a human would work. This is an easy approach to robotics for a person that is not good with technical skills or does not have experience with robots. This approach often leads to slow robots and plenty of unnecessary clicking and writing. From point of optimization this will mean that the lead times will be long.

Robustness of objects is key component for optimization as it brings more predictability for the run times. Robustness in this case refers to the objects capability to handle different scenarios. For example, if the computer lags, is it able to wait dynamically to the response.

4.5.2 General problems regarding robot's process architecture

One of the most important things when doing first process model and draft for blue prism architecture is to understand how many applications are needed in the automated process. The number of applications can be smaller, bigger or equal to the number that humans use when working on the process. For example, the number can be bigger if external reporting applications to pass data to human is needed or text recognition software is needed. The number can also be smaller if some parts, for example calculations can be done within the robot with code.

Once the applications are known, next step is to find how many robots can operate in that application simultaneously and if there are any restrictions on applications operating hours. For example, in banks most of the applications are down during nights and weekends.

If there are parts that do not allow multiple robots to work simultaneously or have smaller operating hours, those parts should be separated to own processes. By taking the bottle necks to own processes a better scalability can be achieved and thus increasing the volume and reducing lead times. Dividing processes to multiple sub-processes also reduces the effect of one part breaking. This also improves maintainability since the error is easier to locate.

4.5.3 Options for process architecture

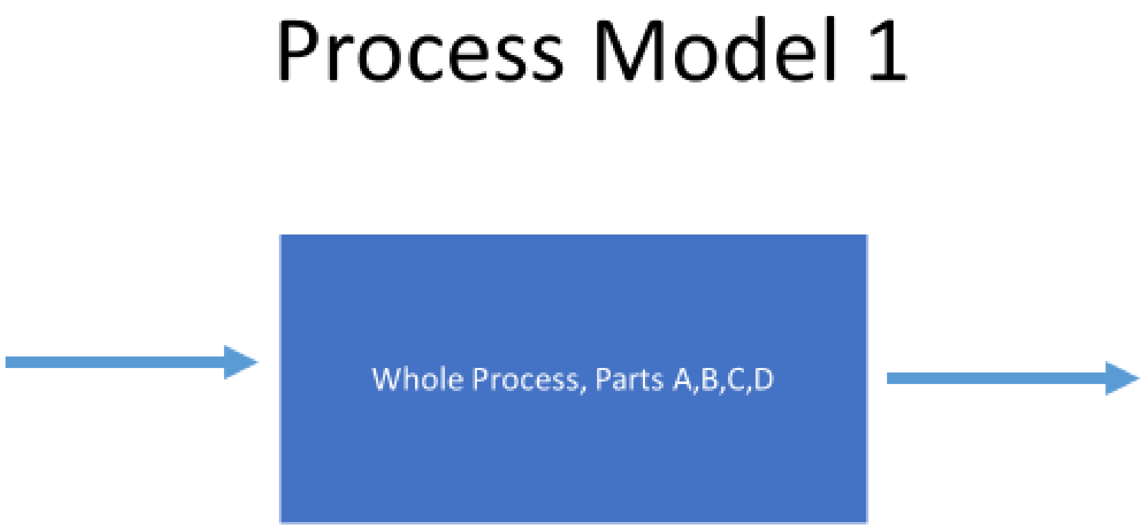

Figure 10. Process Model 1

Figure 10 is the basic process model. In this model, all parts are together and cases are run one by one through the whole process. This model is easy to understand and it is easy to follow which cases have been worked. Also, once the case is started it will be completed immediately, this prevents any confusions between manual workers and robots. Hence there are two statuses that the case can have, completed or pending. If there is even minor change the whole process stops working until it is fixed. The progress of this process type is easy to follow and the process is easy to schedule. If there is a software in use that is not available as much as others, it will limit the maximum uptime. With this model the all process level changes must be committed simultaneously what might harden the maintenance and updates on long processes.

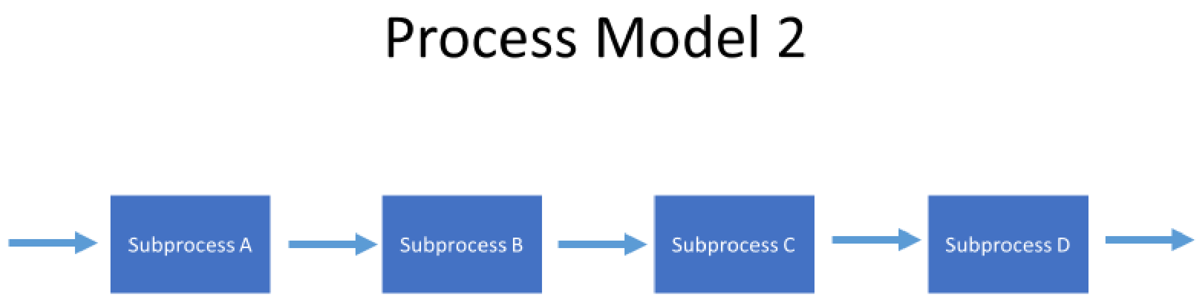

Figure 11. Process Model 2

The Process Model 2 in the figure 11, is process model where process is split to smaller subprocess. All subprocesses are own processes in RPA software and they have own queues. In this model cases go through in batches. Running with one robot this model could be run for example by doing 100 cases on subprocess A. Then working 100 cases on subprocess B and so on. This process model also allows handling for example with one robot doing each subprocess. This model could provide faster handling, since the robot does not have to close and open applications as many times. Dividing process helps on memory conservation. This process can be imported to production on parts and thus making the gains of automation be deployed faster. Since the parts are not dependent on each other, if one part breaks, the other still works. This process model helps with bottleneck handling, since there can be multiple robots doing the slow part and only few robots doing the faster parts. This model's negative side is that knowing what cases have been handled gets harder. It gets even harder on reality, since there are usually exceptions that must be stopped on that part. So just by following the subprocess D's output does not tell about the cases that only went through A and B, unless especially designed to do so. For one item going from subprocess A to D can take some time. Hence the risk of confusion among the manual workers is bigger and can lead to unnecessary work.

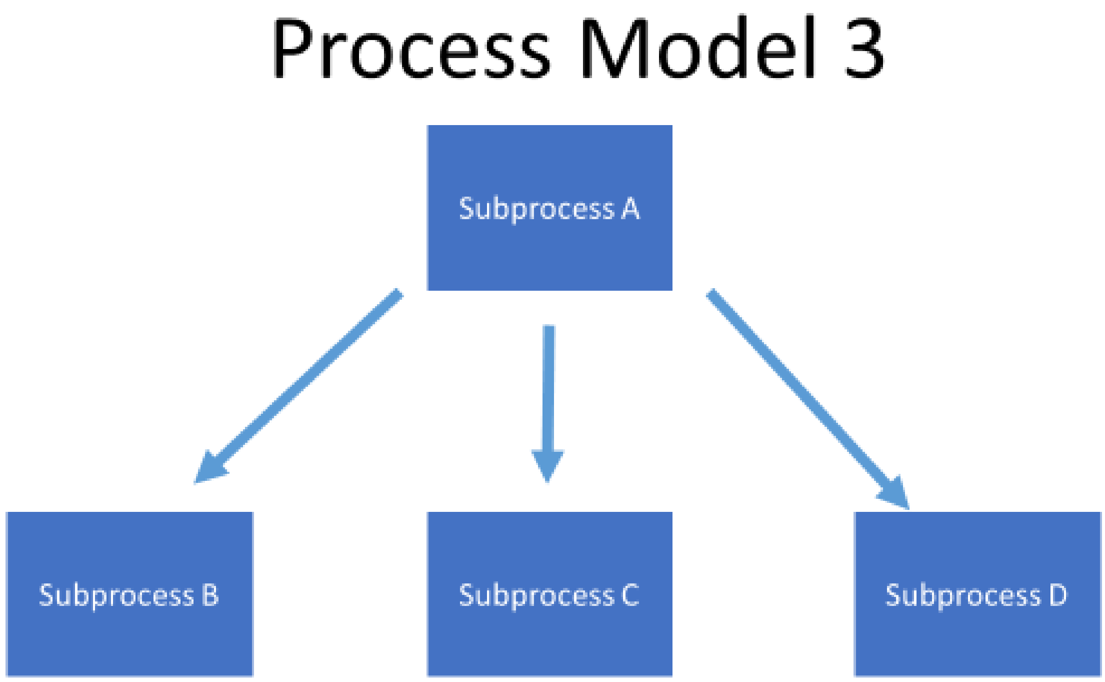

Figure 12. Process Model 3

The Process Model 3 has some of the same qualities than process model 2. However, in this process model the parts B, C, D are not related at all and no information can be shared between them. Thus, following what has been done and what not, requires more work. This process's benefit is that it can cope well with changes and if one subprocess is down it does not matter as much. In this model if the case goes through the subprocess A it will at least start B, C and D.

In companies that utilize RPA, they probably have processes that have different structures and these basic models have their places and usages. In some very long processes mixture of process model 2 and 3 can be used.

If types 2 and 3 have been used, the first subprocess can be called as a populate process. Once there are many processes that have populate subprocess, an active populator can be done. In this model, main process runs different populates on turns. The process can be stopped after any subprocess. As this model gathers the items, there is better understanding what processes should be run and how long it takes to finish them. Thus, helping to keep on track how much robot capacity is needed for the following hours. This model increases the knowledge about what times the items comes. Also as there is one machine actively populating it helps on practice to increase production efficiency.

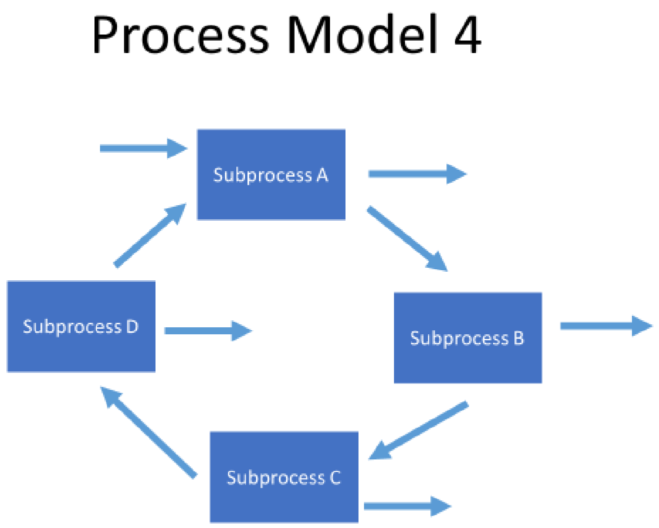

Figure 13. Active Populate

5. CASE OPTIMIZATION

In this chapter, an optimization attempt is produced. This chapter will give heuristic guide and some mathematical formulas and codes on how to produce and build optimization for versatile RPA solution. The purpose is to cover widely different kinds of processes and how they can be fit together to a single scheduling code. However this chapter assumes that the general architecture of processes have been built with chapter 4 guidelines to minimize the issues that can follow from trying to schedule these jobs. These kinds of problems include complexity on task flows and on generally understanding the dynamics on what are the different process complication time requirements.

As the scheduler where being build, it was noticed that new task kept coming all the time. Small request, extra customers and cases had to be run as ad hoc work very often. Also, there are three basic types of processes that have been noticed. These are shown especially in financial services. The first type is continuing processes. These include all kinds of applications and request that for example customers can send. For example, loan applications are this type. They come in quite regular intervals throughout the day. The second type is periodic works. For example, many accounting processes are such that they must be done at specific time of month or quartal. For example, balancing accounts. The third type is processes that come as lists and they do not have as tight time frames. These include internal changes, updating and moving different information in systems, writing bills. Thus, scheduling next week's run was pointless as they had to be changed anyway for many times, also the information that something is done at specific time on next week does not give great value, especially if it ended up being done at other time, as long as everything is done within the time demand that is required. Thus scheduling 1 day forward should be enough. Also, the certainty that things go right is much higher within 1 day than in long period. As the solution does not have to be hard real-time, but from experiments and interviews it was found that ten to fifteen minutes updating period is sufficient. This is due to the fact changing the process takes approximately 5 minutes.

For this research two optimization methods were selected. First one is earliest due date and the second one is using cost functions. Earliest due date was selected to ensure that all cases will be handled on time. As it was found to be the number one criteria for success it is the choice what was wanted to prioritize. FIFO or critical ratio could basically work as well but there is higher risk of failing at the most important criteria. Commonality is that these options are easy to understand and to produce. Cost function on other hand would allow more complex thinking and drive the general development of scheduling the AI and machine learning side, that have been hyped for long time. In this chapter both of these ideas are explored and the ground work is established on how to build such solutions. To compare those options a test bench is generated. That test set up is described in the subchapter below.

5.1 Test setup

For the optimization testing, there are two setups created. Other one is test environment that is simulation in Simulink. The other environment where production tests are made is production environment in real company who is utilizing software robots. The test environment was made to practice and tune the setup, so it was safe to test it in production under supervised circumstances. As the detailed information of the production environment is not wanted to be published and the test environment is a replica of that, only high-level description can be given.

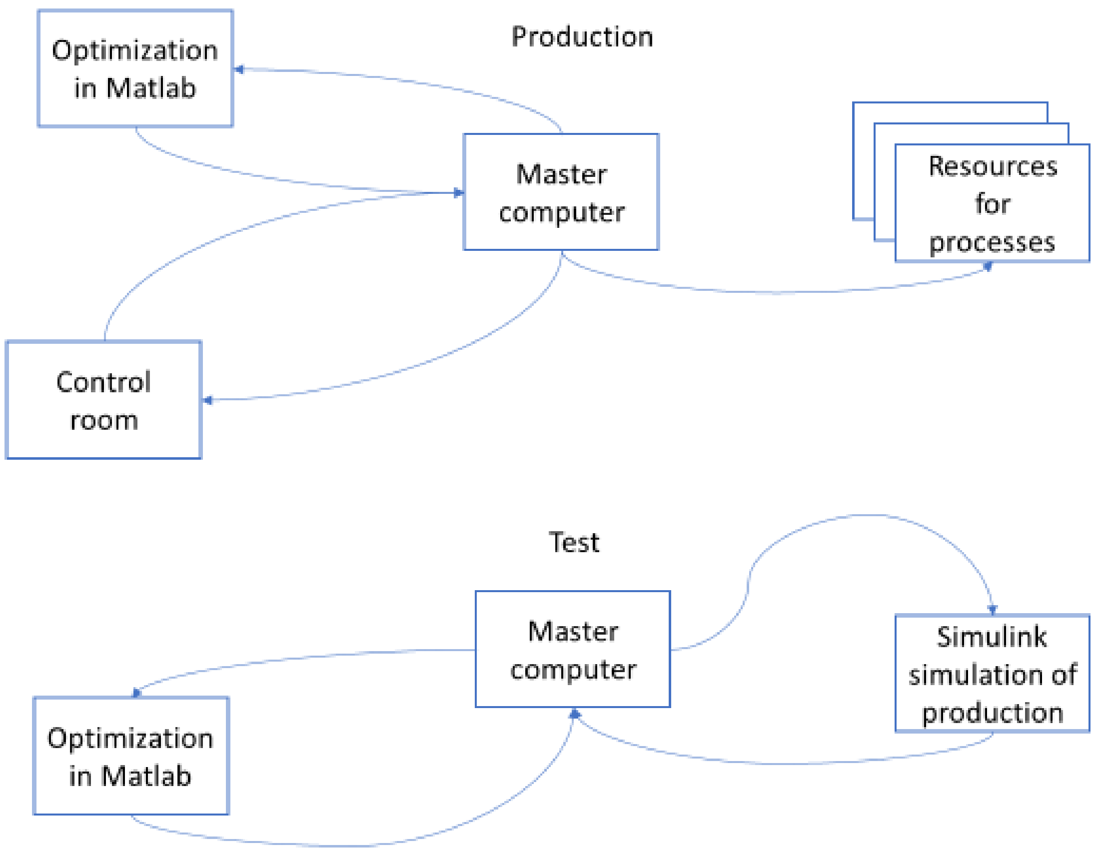

Figure 14. Test and production environment

As seen in figure 14, the test production environment consists of 4 pieces. The master computer that has the automation to get needed data and pass the information. The optimization codes that are in MATLAB. RPA control room where the status of the resources and number of jobs are viewed. And fourth piece is the commanding the resources to working as wanted. The test environment differs a bit from the production environment. As there are no real resources or control room in test, there is a simulation that simulates the acting of these two entities. As the maturity of RPA environments is still quite low, the number of processes for this testing was set to 3 processes. Locking down the environment early meant that the focus could been shifted on improving optimization and not on developing and toning test environment for new processes.

The three processes that were in setup were following. The first process is a continuous process where the tasks are coming on average 5 tasks per hour. The SLA for that process is 1 day. The average processing time is 5minutes per task, but in real life the variance is quite high. The second process is a batch process with a list of 500 cases. The list comes once in a month and has 20 day SLA. The average processing time is 2 minute per task. The third process is similar to the first one. It has on average 7 cases and the average processing time is 2 minutes per case The SLA for the third process is 20h. This data is presented in the table below.

Process	Number of Tasks	Average processing time per task	SLA from getting the task
Process 1	5 per hour	5 min	1 Day
Process 2	500 per month	2 min	20 Days
Process 3	7 per hour	2 min	20 h

5.2 Production restrictions for processes

For most of the processes and environments there are limitations. As discussed earlier for example time limitations can be caused by updating windows or general software and environment downtimes. Also, regulation and policies can prevent certain processes to happen at given times.

The limitations to suitable times can be done in begin of optimization code. If the uptimes for processes are similar every day, hard coded tables can be generated. Also, if there are processes that must be run at specific time or specific date, they should be placed first to the schedule. These processes with specific starting time should be done with simple if, else statements, as they do not have any other options.

Other types of limitations include the number of machines on which the process can run simultaneously. There can be limitations on which machines the process can run at all. Limitation on machines is best solved by adding table to optimization, which maps the jobs to appropriate machines. Even though as a principle machines should be identical but as there can be exceptions those should be identified in a similar fashion.

There can also be jobs that cannot be run simultaneously. This can be caused for example since the robots have same user accounts and that system only allays given user to have one session open. Also there might be need to import and export data from same excel and as only one person can open excel at once it is risky to have to machines on using same excel. This will help on preventing processes to get stuck or fail.

To map these a matrix is a best way of to describe the information. For example if there are four different jobs that can be processed, a 4 by 4 matrix should be created. This matrix should always be symmetrical matrix. As the jobs are connected in symmetric relation do to the fact that the relations are caused purely by technological connections. Hence the matrix must be symmetrical as it is not possible to have situation where it would be possible for example to have job 1 run simultaneously with job 2 but job 2 cannot be run simultaneously with job 1.

1	0	1	0
0	1	0	1
1	0	0	1
0	1	1	1

Figure 15. Job matrix

From this matrix should be read so that first row and column are for job 1. Second row and column are for job 2 and so on. As it is noticeable from row 1, job 1 can run simultaneously with job 1 and job 3 but it cannot be run same time with job 2 or 4. Similarly from row 2 it can be run simultaneously with itself and job 4. Job 3 cannot run simultaneously with itself and job 2, but it can run with jobs 4 and 1. Job 4 can run with itself and jobs 3 and 2.

If there is need to limit the number of simultaneous runs with itself the diagonal line can be used to make those limitations. The as in the following figure there is presented the same matrix but the diagonal line now contains the information on how many simultaneous process can run for that job. If this kind of presentation is used the numbers on diagonal axis must be greater than 1. Basically all number in that diagonal line from previous matrix has to grow with at least 1.

2	0	1	0
0	2	0	1
1	0	1	1
0	1	1	4

Figure 16. Job matrix with limit on simultaneous jobs

Now there is limitation that job 1 can only run on two machines at time and as they are running third machine could run job 3. Similarly with job 2, it can run at maximum of two machines simultaneously and job 4 can run simultaneously on third machine. Job 3 can only run on one machine at the time and job 4 can run on four machines.

5.3 The cost of waiting cases in different processes

One of the possible optimization solutions would be trying to minimize the cost of jobs that they wait. Comparing this solution for example with earliest due date algorithm there are few differences. Firstly the balancing of cost of waiting was to be set properly. This might be hard and requires long and comprehensive analysis on what jobs should be prioritized. The cost is not necessary the monetary cost of potential client losses due to waiting but there might be also some strategical and tactical prioritization on what customers and on what cases should be run first. However using cost function alloys this type of tweaking, also this alloys better way of responding to unexpected events.

As just said the cost that is given to cost function does not have to be the correct monetary cost, but rather relative prioritization on witch process should be run at different scenarios. The monetary perspective is still a good starting point and should be analyzed first. For jobs that are related on selling something new to customer, finding out there is correlation with time in which they receive offer, monetary evaluation should be used. As determining the monitory cost can be quite challenged sometime and is not appropriate at all jobs, some other parameters such customer satisfaction or legislation can be used as starting point. For example to sending bills, customer satisfaction analysis should be used to know what dates or time periods are best to send bills. There might be also some elements which are monetary for example people might pay bills with higher rate if they receive it on their pay day rather than 2 weeks after it. Also they might pay it earlier on some time periods than others. Legislation can be used especially in accounting processes. These are quite straight forward requirements and there is usually quite little room to play in.

After this kind of data has been has been found they should be balanced together to make solid entity. This is the part where strategical analysis should be done on so that other methods do not overpower others or if wanted so then how much. To build the functions that determine the cost as function of time different functions are needed. For this purpose few example will be given so that reader will have idea on what types of solutions could be utilized. Functions should be built on occasion to correspond wanted dynamics for needed purpose.

5.3.1 Customer experience

For customer satisfaction case the cost structure can differ greatly depending on the process and the customer base. As customer satisfaction is wide and complex are of research it will not be reviewed here. From experience through examples, few basic types of customer satisfaction behavior will be discussed here.

Generally companies have SLA is the promised due date for customer or internal customer within the service or product should be delivered to the customer. For example, SLA for handling credit card application is 1 day. As SLA are something that the service provider promises to customer breaking it can lead to very negative customer experiences. For some customers getting the service quickly can be determining factor, instead of the price. Thus, understanding the customer need and purpose of the service can impact greatly how it is prioritized. These kinds of customer needs can be received by surveys and customer feedback questioners. The customer need and requirements can vary greatly for different customer segments. For example, young people would probably expect application filled on internet to be replayed within minutes or hours whereas elder people would be happy to get the answer in couple days.

There are plenty of views on how to determine appropriate SLA’s. As robotics is usually applied to existing processes there are already existing SLA’s. However, robotics enables to handle more cases faster, so SLA’ can and should be tighten. Determining appropriate SLA’s can be challenging. Haynes (2014) talks about one common rule, that SLA’s should be selected, so that 95% of handled tasks are completed within the SLA. Hence getting to understand what can be done in what time and how much resources it requires is in key aspect in determining appropriate SLA. It is good to understand that the SLA’s must be kept 24 hours every day. So, if there is a problem or update in a robot, there should be wiggle room to fix everything. One aspect of determining SLA’s is to look what other companies promise and use them as a baseline.

As customer satisfaction differs for every process and every company on every sector let’s take example on see possible dynamics. The dynamics is presented in figure below. The figure below is a one type of behavior that can happen to customer while they are waiting for the service.

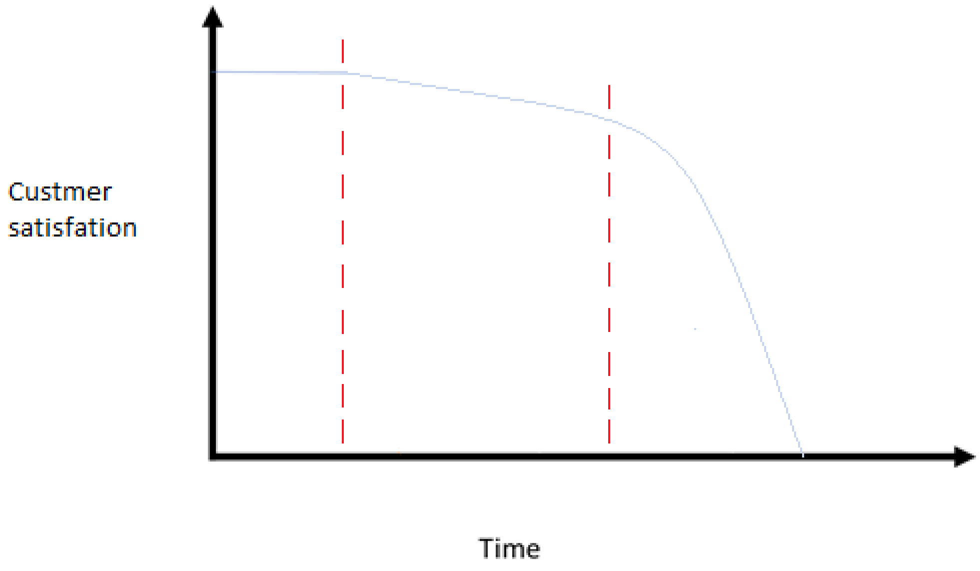

Figure 17. Customer satisfaction over time

On horizontal axel is the time that it takes to handle task from end to end. On vertical axel is the relative customer satisfaction. The first vertical line is the point that customer is waiting happily, this can be the SLA or other expectation of the service speed. After that the customers that are in the most of hurry are starting to drop out and taking other firms offers. After a while comes the next vertical line after the masses with typical expectation of what something can take are starting to take other companies offers. If the time goes over the second vertical line, there will be a big number of resending the order and contacts to customer service numbers and front office employees.

To design a cost element to this job, reverse elements have been found to be good. As the figures begin is very flat the cost multiplier should be constant. So that the cumulative cost is linear. On the second part where the satisfaction drops down as linear the cost factor should also increase linearly. And on the final exponential part the cost factor should increase exponentially.

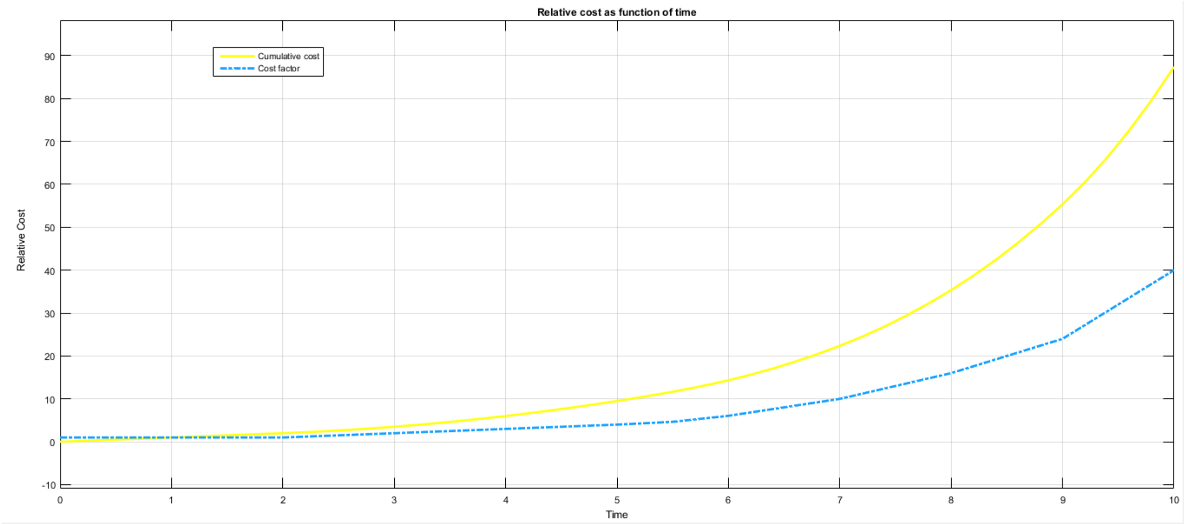

Figure 18. Relative cost caused by customer satisfaction decrease

On the diagram is represented cost function that on shown with blue dashed line. Also the cumulative cost is illustrated with yellow line. When figures16 and 17 are compared it is noticed that the cost of waiting goes up as the satisfaction goes down. This prioritizes the job higher as time goes on and as happiness decreases.

5.3.2 Legal requirement

Legal requirements can be found for example in accounting processes. There are lots of rules when a certain task must be done and business limits from the other side when it can be started. This is especially important for businesses that have been listed in stock markets.

If there are jobs that are made by legal requirements, getting the completion time is much more critical. It was found that the best way to make process to be done on is to have linearly growing cost coefficient until a safety bonder is reached. The safety bonder could be for example one day, but depends on stability of platform, number of machines and jobs. As the safety bonder is reached the coefficient grows with step to infinity to ensure the start of job.

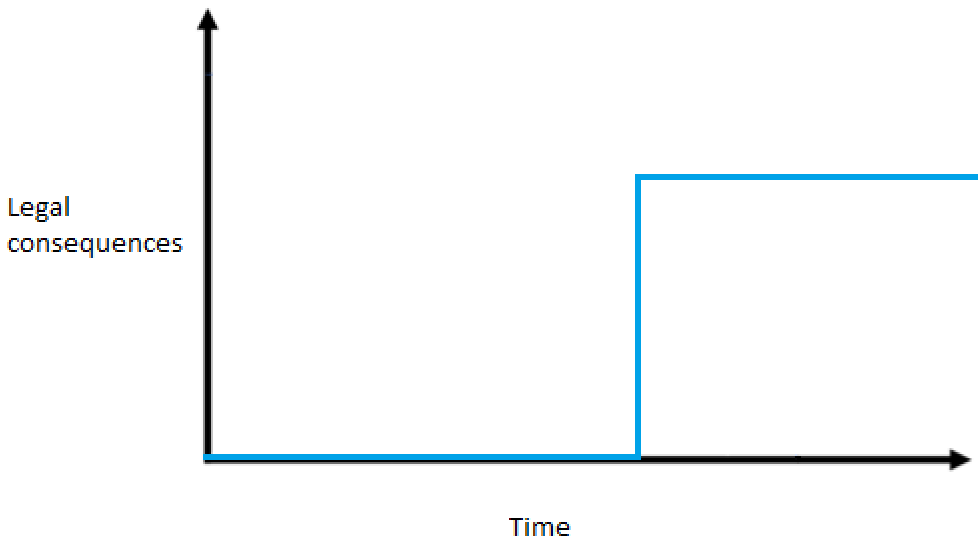

Figure 19. Legal consequences over time

The cost coefficient should have similar shape, but the step should be in advance. The amount of time that should be there is dependent on the environment. Stability of the environment, number of jobs and computers and process that have this kind of requirements. The period of time should be between one day and 6 hours to be safe. The step should be infinite size, to make it number one priority.

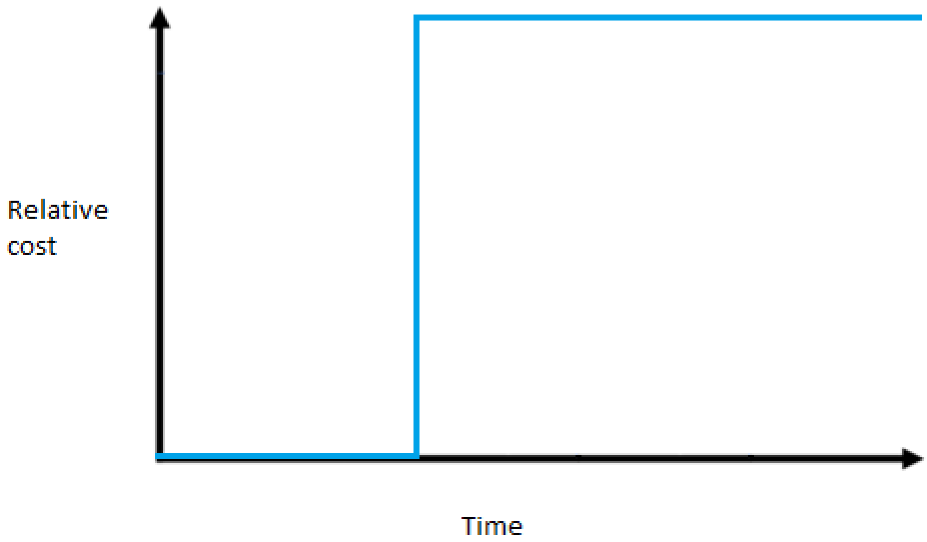

Figure 20. Relative cost caused by legal consequences

5.3.3 Monetary cost

Monetary cost coefficient should be calculated in a similar manner as customer satisfaction. They usually behave in a similar fashion as they are connected. Monetary cost and customer satisfaction have been separated because there might be services that do not make any money for the company as such. However they might give great value to the customer.

Monetary value should be taken for jobs that have immediate monetary cost to the company. For example, if bills are not handled on time, there will be interest running. To receive money, opportunity cost could be used as a measure of monetary cost. The dynamics for these is typically exponential or linear. They seldom have changing dynamics, first flat then linear and the end exponent, as customer satisfactions has. Thus the cost coefficient is easier to adjust by suing the same principles.

5.4 Building optimization algorithms

As discussed earlier, multiple algorithms could be used. While keeping up with SLA's is one of the most important things for back office processes thus earliest due date was selected. In subchapter 5.2 there was general discussion on how to start the earliest due date optimization with the fix-timeframe jobs so that they do not have to be considered. In subchapter 5.3 was discussion about how to make the limitations on what processes could be run simultaneously. There is also need to know what items are pending to be run, and when is their due date.

To get information to algorithm there needs to be system that populates the queues and to constantly follow the number of customers in each queue. For this the chapter four introduced active populating concept. Other possibility is to try model and predict the customer flows and making predictions based on them what is hard and requires some mathematical modeling. With this method, there also needs to be populating methods to get measurements to get data points for the model, but it does not have be done as often.

To get the optimization code and the blue prism control room to share data, a robot is needed. This setup will need one extra license but the savings come with better utilization, less licenses for the actual jobs and on saved man hours. The robot must be able to go blue prism control room, check if active population is on if not run it, reads all the queues and their items, then go back to MATLAB and write all jobs and their due date to a data table. There should be list made to the robot, so that it can calculate the due dates. The MATLAB code will return what jobs will be run on what computers and then the robot will go and check if those processes are running and if not it will start them. This logic is also illustrated in the figure below. Also, if there are special due date rules other than with in x hour, they should be coded to the robot. Producing this kind of robot is straight forward for everyone with knowledge about graphical programming. This robot should be online all time so that the due dates will be as correct as possible and to ensure as high utilization as possible.

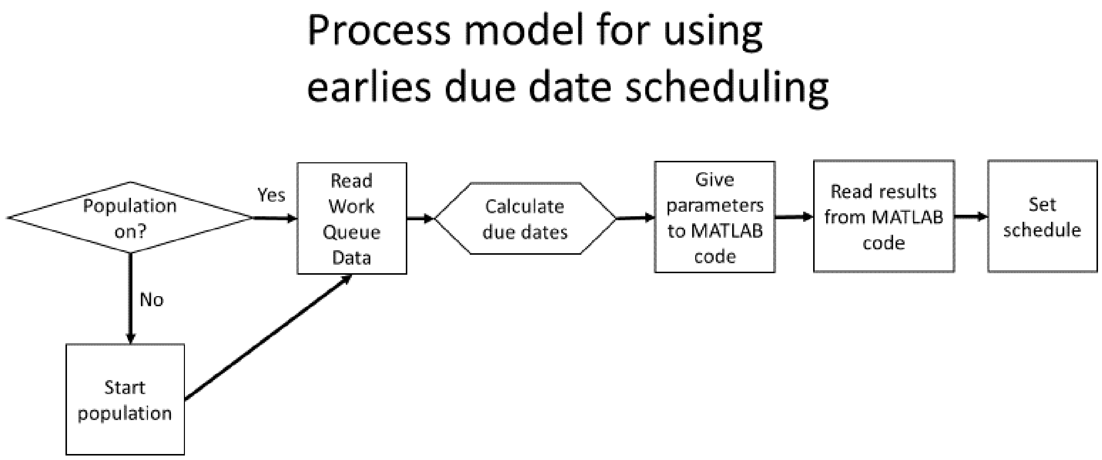

Figure 21. Process model for using earliest due date scheduling

The optimization algorithm for the earliest due date is simple. As said earlier the fixed processes are set to the timeline first. For timeline a data table could be appropriate solution. The main loop will be the one filling jobs for one day forward from the current time. The code will be run every time the robot comes with new parameters. At the begin on loop the code checks that if all machines are empty. If there is a job already scheduled filter out jobs that cannot be run simultaneously unless take the job with earliest due date. Assange task with earliest due date to first available machine. Repeat assigning jobs until all machines are filler or there is no job possible to assign. After that move to next time slot and repeat the loop.

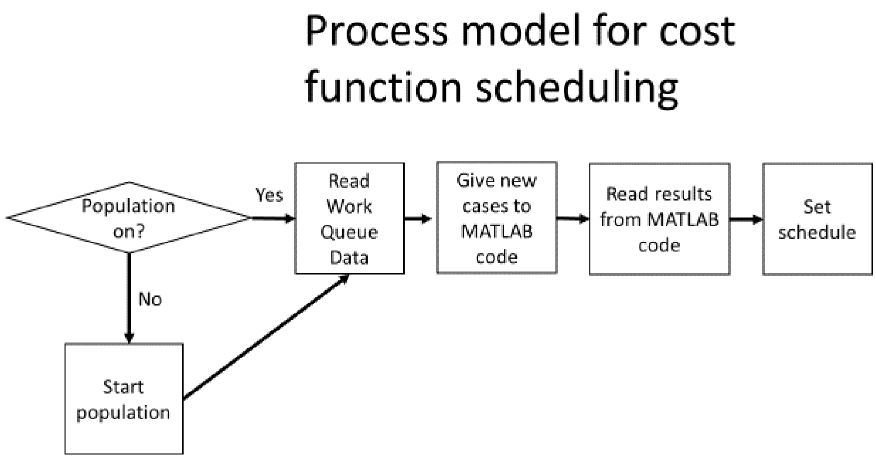

Figure 22. Process model for using cost function scheduling

Cost function have will work quite similarly as earliest due date, except on how to optimization algorithm works. Also for the general process model there is no need to calculate the due dates. For cost function the subchapter 5.3 will guide on how the parameters should be generated, these should be hard coded to the MATLAB code. To cost function it was easier to calculate the setup cost. The setup cost was calculated simply by adding direct license and direct virtual computer cost to find the minute rate of the computer. This cost was then used on cost to change the process and then giving a scaling factor to make it work better. To change process the average time was found to about 7 minutes. Thus the cost of changing the process was easier to implement in a manner that it prevented unwanted jumping from process to process.

6. RESULTS

In this chapter the results for two selected scheduling algorithms will be looked. There is also discussion about the findings and their implications. The research questions and the secondary questions are answered, and further research questions are laid down. The results from interviews are also reviewed.

6.1 Comparison of different results

The earliest due date algorithm showed that it could schedule the processes and determine the following processes. It provided reasonable solution, with reasonable workload. It does not beat human on choosing smartly what jobs to do, but it was able to monitor working almost without any breaks. Hence, it was able to keep utilization rate higher. Either of solution did not remove the need for human controlling as there is still need to monitor, but it was able to decrease the amount of job and to make decisions for human. The main problems were on running processes smartly. As the decision was based on executing task with earliest due date it kept changing the jobs quite often. Setting up limits that it had to handle at least 10 similar jobs the overall efficiency improved, as the setup times is quite high.

Cost function is found to be quite hard to implement so that it would work properly and it requires lots of work to maintenance, the benefits are small if the platform is stable and it only contain processes of certain type. In mixed complex scenario or with multiple processes and resources, according the test it would be better option, however fully testing and setting up such complex scenario is out of scope for this work, and the real-life places are still rare to test it on practice.

The human benchmark result was measured without telling the controllers so that they would work as normally as possible. The benchmark state was wanted so it would be easier to compare results, benefits and problems of as is state and possible to be states. It was found that humans used two ways to run this set up. Either by running all cases for process once in a day or by changing the process at once in an hour.

Run mechanism	Jobs on time	Amount of human hour needed	Process 2 cases completed during 1 day at max
Earliest due date	100%	1h	207
Cost function	100%	1h	247
Manual	100%	8h (3h active 5h passive time)	220

Figure 23. Results

The metrics to compare different approaches was decided to be the percentage of jobs completed on time, number of human hours and maximum number of items processed in a day to have fair comparison of the different run mechanisms. The most important metric was that all jobs had to be done on time. This is something that all methods managed to do successfully. The amount of human work hours is also interesting, as it was the main driver for this experiment. The time for human work hours needed could go down, as the learning of utilizing this kind of approaches would be better and the scale would increase. On manual work it seems to be harder to improve and this a rather getting harder and harder. The performance of the approach was measured by the number of completed cases in the process 2, that was the fill up process to be run when nothing else is available to be run. Overall differences are not that big. Also, something to remember that humans were not working optimally and they could have improved to be the best option by the number of process 2 cases completed during a day.

6.2 Is optimization profitable

As little bit discussed earlier, there were couple of points that were improved but couple things that did suffer a bit. Comparing the monetary effects of these changes is not simple as they differ from the company quite a lot.

The improvements were on the production timespan. Also with minor work it was possible to make the optimization robot to monitor that processes are not stuck, because of software or network instability. Hence, the robot could work with higher success rate during the nights and weekends. Also this reduced human labor but testing was inconclusive on precisely how much it was reduced as it was found to be dependent on the environment and on number of processes and resources. It is good to notice that the working times in the figure 23 are for case when everything goes smoothly. But as things do not go smoothly it was a lot quicker to escalate when people were controlling than, letting the robot notify someone to come check it out. This is also a thing were learning and experiences and internal working processes will effect a lot to result.

On the downside, prioritization by command suffered. If someone would have had quick urgent need, it would have been hard to interfere to the system. This would be fixable by adding manual population process and one extra license. Also with earliest due date the efficiency of day time suffered a little bit and with this setup testing cost function does not give right data on how it would behave in large scale. Also making and maintaining such system takes time and resources.

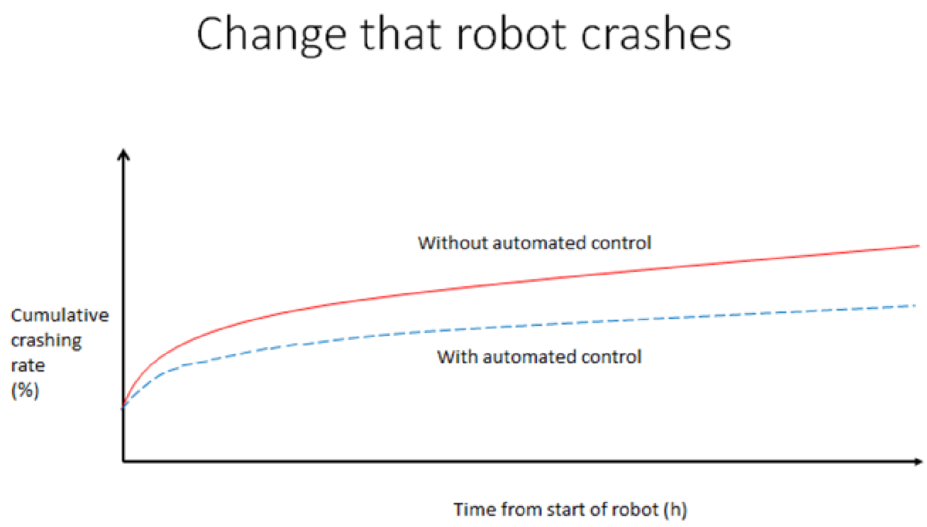

Figure 24. Stability comparison between no supervision and automated supervision

As a conclusion to optimization, it seems like optimization could be profitable, as it could keep total utilization higher especially during nights, holidays and weekends. However, to say how many processes or resources there should be to use this either of optimization methods is not possible with theses experiments.

6.3 Interviews

Interviews gave quite precise direction, what should be fixed and how. Most of the answers corresponded to the findings of this research, however also few different views were presented. The full interviews are presented in the Appendix B. The main points found are discussed on this subchapter. In the interviews, the question can be grouped to four categories. First one is to find out how familiar the person is with scheduling with RPA software. The second group is to find out current setups. Third group on how to classify processes. And fourth on what are the possible pain points on scheduling.

From the answers, it was noticed that people were quite familiar with scheduling. They could describe answers in detail and they used correct terms. It was also noted that the interviewees were familiar with the business side as well.

The interviewees described the current setup to be done manually with utilizing with fixed timetables on scheduling. They all had some ideas that it could be automated, but as talking outside of the question there were very little if non-attempts on automate scheduling. The utilization of this solution was not measured by anyone but everyone thought that it was poor. However interviewees thought that not all times is as valuable as others, so getting weighted or weekday utilization would be better. With this we can see that the production focuses heavily to the timeslots that people are working. Is this since all jobs should be done during the day or that scheduling outside working hours is hard to answer with accuracy.

The third group was bit more mixed and not clear results were not get. One of the approaches of classifying to processes was taken to consideration in subchapter 5.3. As discussed in that chapter there was reason why that approach was selected but other methods could have worked as well. From interviewees it was noticed that depending on their position and hands on experience with RPA affected on their view of classifying processes.

For the main points interviewees were quite consistent with their opinions. The main problems were related on getting the processes run on simultaneously. They reckoned that setting timetables was quite time consuming and it was not scalable.

7. CONCLUSION

In this research, it was tested if optimization could be used to improve software robot production efficiency. With humans, it was found that the production efficiency was quite low and the active hours were quite small where the scheduling is good. As the licenses are expensive and the current setup is not scalable to factory level automation, some changes will be required.

At the chapter 4 there were discussion about what are the best practices on development when trying to produce solutions that are scalable and automatable. During the research phase, multiple automation architects was tried but finding the bottlenecks and splitting processes to multiple processes from the bottle neck parts was the best solution. Also splitting long processes to many smaller processes had multiple benefits as better testing ability, easier debugging and better scheduling.

The approach that was selected in this research was using earliest due date scheduling and cost function. The reason why these were selected was based on the fact that in the interviews it was found that completing everything on time was the success meter that was followed. Especially the earliest due date will follow that. The cost function on other hand was seen as a better, more sophisticated and scalable results which could priorities and further improvements on efficiency as the earliest due date did not provide optimal result.

It was found that earliest due date was able to schedule the jobs successfully. However the result was not robust in a sense that if there were jobs on different processes so that the earliest due date was on one process and then on next process and then back to the first process. The overall setup time percentage of daily operations got quite high. With some extra rules, it was made bit better but as it was not behaving as wanted.

The cost function was found to be working well but the test environment was not complex enough to get really challenge it or to make the fine tunings to it. With the simple environment, it however worked as wanted. However, to automate such simple environment with this result would be waste of time, as the simple timetable option was able to work on that too.

Experimenting with the solution and from these results it would indicate that this kind of solution would be able to work on larger scale and work efficiently. The method is possible solution and nothing that would indicate that it would not scale has not been noticed.

7.1 What could be improved

The complexity of the test environment should have been bigger. As it was selected early, there were quite limited tests that could been perform. As the test environment was quite simple, the research question about the stability of solution was not being tested. Also, there should have been a pure test environment with random data and self-made processes, so that better comparison of different methods could have been fully tested, to get concreate numbers on their performance.

As this research is there are quite little actual scientific results on the efficiency and the suability on this kind of system. However, the main purpose was to demonstrate if this a viable solution for this problem.

7.2 Further questions

As discussed on earlier subchapter the hard numbers are still to get. Also the question about the profitability of this kind of solution is still open. The secondary research problem: Could optimization program bring extra stability to autonomous processes or will it cause more problems? Is something that was not discussed at all and would be very important. Both to calculate the monetary gains or losses and its part on the efficiency side.

During the research number of new questions also rose. For example, what is best method to handle process failures, how to monitor process that get stuck and needs to be restarted. Also, how the internal process should be changed so that they more support maintenance better. As stated previously there are still lots to work on improving this method on more complex cases and designing on how to balance costs of different processes.

REFERENCES

[1] B. Markgraf. 2016. How to Identify Bottlenecks in Manufacturing. [ONLINE] Available at: http://smallbusiness.chron.com/identify-bottlenecks-manufacturing-72465.html. [Accessed 11 December 2016].

[2] C. Richardson. 2014. New Development Platforms Emerge For Customer-Facing Applications. [ONLINE] Available at: https://www.forrester.com/report/New+Development+Platforms+Emerge+For+CustomerFacing+Applications/-/E-RES113411. [Accessed 16 January 2018].

[3] D. Pham, A. Klinkert, 2006. Surgical case scheduling as a generalized job shop scheduling problem. European Journal of Operational Research, 185, 1011–1025.

[4] G. Moore, 1998. Cramming components onto integrated circuits. Proceedings of the IEEE, VOL. 86, NO. 1, 82-85.

[5] GitHub. 2018. GitHub - tesseract-ocr/tesseract: Tesseract Open Source OCR Engine (main repository). [ONLINE] Available at: https://github.com/tesseract-ocr/tesseract. [Accessed 16 January 2018].

[6] Google Developers. 2016. The Job Shop Problem | Optimization | Google Developers. [ONLINE] Available at: https://developers.google.com/optimization/scheduling/job_shop. [Accessed 20 November 2016].

[7] Harvard Business Review. 2015. What Knowledge Workers Stand to Gain from Automation. [ONLINE] Available at: https://hbr.org/2015/06/what-knowledge-workers-stand-to-gain-from-automation. [Accessed 11 December 2016].

[8] HfS Research. 2012. Robotic Automation Emerges as a Threat to Traditional Low-Cost Outsourcing | HfSResearch. [ONLINE] Available at: http://www.hfsresearch.com/pointsofview/robotic-automation-emerges-threat-traditional-low-cost-outsourcing. [Accessed 11 December 2016].

[9] I. Hayes, 2004. Metrics for IT outsourcing service level agreements. Available at www.clarity-consulting.com/metrics article.htm. [Accessed 04 December 2016].

[10] Investopedia. 2016. Back Office Definition | Investopedia. [ONLINE] Available at: http://www.investopedia.com/terms/b/backoffice.asp. [Accessed 06 November 2016].

[11] K.H. Kim and Y. Park, 2003. A crane scheduling method for port container terminals. European Journal of Operational Research, 156, 752–768.

[12] L. Latecoere, J. Boebion, L. Pun, 1976. Model adaptive scheduling for solving perturbation problems in the clothing industry. International Journal of Production Research, Vol 14, 239-250.

[13] L. Willcocks, 2016. Service Automation: Robots and the Future of Work 2016. Edition. Steve Brookes.

[14] M. Lacity, L.Willcocks. 2016. Robotic Process Automation at Telefónica O2. MIS Quarterly Executive, 15/1, 21-35.

[15] M. Masoud, E. Kozan, G. Kent, S.H. Liu, 2015. An integrated approach to optimise sugarcane rail operations. Computers & Industrial Engineering, 98, 211–220.

[16] Q. Pan, M. Fatih Tasgetiren, P.N Suganthan, T.J. Chua, 2009. A discrete artificial bee colony algorithm for the lot-streaming flow shop scheduling problem. Information Sciences, 181, 2455–2468.

APPENDIX A: INVERVIEW QUESTIONS

1. Can you define shortly what is RPA?
2. Can you describe with few sentences your relation to RPA?
    a. How about scheduling? if not answered already
3. Could you categories if there are different types of jobs there are?
4. What type of processes are easy and hard to schedule?
5. How are the jobs scheduled?
6. Are the process run on time?
7. What is the resource utilization rate that has been achieved with this method?
8. How scalable you think the current setup is?
9. How time-consuming scheduling is?
10. What kind of problems have been faced?
11. How predictable coming tasks have been?
12. When is the scheduling done successfully?

APPENDIX B: INVERVIEW ANSWERS

Person 1.

1. RPA is technology utilizing existing software to automate routine processes.
2. Continues Process development, as well as configuration of new processes, and analysis of potential new pipeline processes. I have scheduled processes in some projects. And thinking of timetables and overlapping times for almost one year.
3. There are multiple ways of grouping processes. From process point of view it could be done by the quantity of items, the length of processes or complexity. From organizational perspective by organizational importance, or customer satisfaction. Also the financial gains. Also the quantity is important on priorizating and strategical goals.
4. Easy are processes that humans are not related, and if humans are using same software that can cause problems. Depending on the process structure on what kind of limitations there are and how item exceptions are handled.
5. Manually, setting timetables on when to run. Finding what times are suitable for different organizations and to ensure that there is time between processes. Also by sending messages to controllers to run to, especially while being on soft launch phase.
6. Sometimes there are mistakes, especially if there are multiple processes on same resource, there are overlapping situations. And the it is noticed after a while that other process has not been run.
7. There has been little analysis on what is the utilization rate. And there could be automation checking if there is room to run processes. There is quite a lot of downtime that is not utilized. There are many ways of calculating utilization and filling up weekends and nights is difficult.
8. There are quite a lot of errors and bugs and there is continuous problem-solving mode. So that is hurting the scalability. Also as there are more and more resources and processes utilizing them fully gets harder.
9. The scheduling and running processes takes about, well most of the time goes to the finding out of problems, but maybe 30-40% of time. That might be a little too high. Depending on how the packages are imported and keeping track of versions.
10. Processes put to same resources and also some processes have been running too long. Sometimes the schedule does not work.
11. Quite predictable at least on process level. They are quite well defined on how many tasks there are on day and weekly level and what time they are coming. Sometimes there are peak times, and quieter times. So there is quite a lot of spare time that has to be taken to account while designing timetables.

12. Scheduling has been successful when the resource can run the process successfully without blocking any other process. It did not block any other process.

Person 2.

1. RPA is a software automation technology that utilizes existing user interfaces but has the capability to also include code for more complex decision making.
2. I have been automating processes with RPA for over year now. I have been part of all stages of building such solution. I have been working as a controller of two companies for couple months each.
3. I would group processes so that there are continues processes, that are processed daily. Then are the monthly/weekly/quarter processes or periodic processes that are run on wanted periods. And last type is ad hoc processes that are run on demand.
4. Processes that are constant. They are run on same time and running takes always about as long. Otherwise to say that the process should be predictable.'
5. Jobs are scheduled manually with timetables to the control room. And sometimes just by waiting the correct time and running them one by one manually.
6. Usually but not always. Also sometimes they are run on wrong order. Also if something is wanted to test on production finding the timeslot is hard.
7. Poor, licenses are not used efficiently. Especially as the setup grows. With two computers and licenses and couple process it is still quite easy to get high utilization but as the number of licenses and resources grows to 20 it gets hard. Also depending how the utilization is calculated. Are the weekends included or not as usually there are no processes run on weekends or nights.
8. Not too scalable. Of course it depends on type of processes and on how predictable they are.
9. Making a schedule does not take that long but ensuring that they were properly set and the processes were able to run and all cases were handled properly takes lots of time.
10. Most of the issues are related to processes not being run or it has terminated. Also processes that have been scheduled so that they are running simultaneously will prevent the later process from starting.
11. They are quite predictable, to processing time is not always as predictable. If more time would be spent on analyzing processing times and item quantities.
12. The scheduling is successful when all cases have been completed so that they were completed within the wanted service level.