Lappeenranta University of Technology
School of Engineering Science
Industrial Engineering and Management
Master’s Thesis

21.01.2019

Understanding Business Process Complexity for Robotic Process Automation

Tapani Qvick

1. Supervisor: Petri Niemi

Tiivistelmä

Tekijä: Tapani Qvick **Työn nimi**: Liiketoimintaprosessin Kompleksisuuden Ymmärtäminen Ohjelmistorobotiikan Automaatiolle	
Vuosi: 2019	**Paikka**: Hyvinkää
Diplomityö, Lappeenrannan teknillinen yliopisto, tuotantotalous 86 sivua, 12 taulukkoa, 25 kuvaa, 2 liitettä **Tarkastaja**: Petri Niemi	
Hakusanat: Ohjelmistorobotiikka, liiketoimintaprosessin mallinnus, kompleksisuuden mittarit, liiketoimintaprosessin automaatio	

Useat yhtiöt ovat huomanneet ohjelmistorobotiikan tuomat mahdollisuudet liiketoimintaprosessien automaatiossa. Myös tässä diplomityössä tutkitussa yhtiössä on todettu ohjelmistorobotiikan tuovan nopeita tuloksia liiketoimintaprosessien automaatiossa. Kuitenkin ongelmilta ei ole vältytty, ongelmiin haluttiin ratkaisuja ja automatisoitujen liiketoimintaprosessien laatua haluttiin parantaa.

Tässä diplomityössä painopisteenä olivat neljä aluetta, Business Process Management framework, liiketoimintaprosessien mallintaminen, kompleksisuuden mittaaminen, ja ohjelmistorobotiikka. Teoriaa käytettiin kaikilta neljältä osa-alueelta, joskin painopisteenä oli liiketoiminnan prosessien mallinnus, mallien laadun arviointi, ja kompleksisuuden mittaus prosessimallin avulla.

Diplomityössä löytyi selkeät hyödyt prosessien tiedon siirtämisestä prosessimalliin, kun prosessin mallinnus tehdään standardoidulla prosessinmallinnus kielellä ja mallin tarkastetaan noudattavan prosessien mallinnuksen sääntöjä. Prosessin mallintaminen mahdollistaa sen kompleksisuuden arvioimisen, joka antaa tietoa mitä prosessista tulisi mallintaa automatisoituun prosessiin ohjelmistorobotiikalle. Tämä prosessitieto voidaan kerätä workshopeissa, haastatteluilla, ja erilaisista prosessiin liittyvistä dokumenteista kuten prosessidokumentaatiosta tai käytettyjen systeemien lokitiedoista. Nykyisen prosessin mallintaminen on lähtökohtana aina tulevalle automatisoidulle prosessille. Automatisoidun prosessin mallintaminen vaatii ohjelmistorobotiikan tietämystä, jotta automatisoitu prosessi tuottaa halutun lopputuloksen asiakkaalle.

Abstract

Author: Tapani Qvick **Title**: Understanding Business Process Complexity for Robotic Process Automation	
Year: 2019	**Place**: Hyvinkää
Master's thesis, Lappeenranta University of Technology, Industrial engineering management 86 pages, 12 tables, 25 figures, 2 appendices **Supervisor**: Petri Niemi	
Keywords: Robotic Process Automation, business process modeling, complexity measurement, business process automation	

Companies have noticed benefits of robotic process automation (RPA) in automating business processes. In the case company of this study the benefits of RPA have resulted in quick wins. However, also issues have occurred during the RPA projects and the quality of automated business processes with RPA is desired to be increased.

In this study the focus was in the Business Process Management framework, modeling the business process and measuring its complexity, and RPA. Theory was researched form all these areas. Main focus was in process modeling, evaluating the quality of process model and measuring process complexity from process models.

In the study clear benefits were found from transforming the process knowledge into process models using standardized business process modeling notation language and validating it with process modeling guidelines. This enabled the measurement of process complexity which gave information of what should be modeled in the automated business process for RPA. The knowledge of this business process can be gathered from workshops, interviews, and different types of process documentation such as business process documents and business process supporting systems log files. As-is process model is the starting point for to-be process model for RPA software. Understanding the capabilities of RPA software is a prerequisite for redesigning the process for RPA in a way that ensures the expected process output.

Forewords

During autumn 2017 I had a luck to find my first long term job. At first, I thought that this would postpone my graduating from university where I started 2012. But during spring 2018 I found a good subject for possible master thesis. It was interesting subject, but it was not the one that I have now written. This master thesis subject, which I have now written was derived from the first subject during the first months. Perhaps I will take this first master thesis idea as my next free time project.

The writing had its own obstacles such as warm summer 2018, which was nothing to compare the perfect rainy and cold summer of 2017 in Finland. Certainly, support from my family and good friends helped me to work and keep on writing this thesis. Support from my boss was certainly one of the corner stones to enable me to write this thesis while working.

After many long days and spent years in the Lappeenranta University of Technology, studying abroad and finding new friends and experiencing amazing time in studies. I have to say that I am also really relieved and happy to end my studies and keep on going forward to face new challenges and improving my capabilities. But as LUT has taught me, supporting the team members and others is always important and will result in the best outcomes. I think this Spirit of Skinnarila is something that I will never forget.

Sincirely

Tapani Qvick

20.01.2019 Hyvinkää Finland

Table of Contents

List of Tables

Table of Figures

1. INTRODUCTION

1.1 Background

A Robotic Process Automation (RPA) project was initialized in a large Finnish-based manufacturing company in 2017. This case company is listed in the OMXH Helsinki stock market. First proof-of-concept automation was developed in the autumn of 2017. After the proof-of-concept, it was decided that RPA could be a way to automate business processes and project team was established in the company. Currently the focus in the case company is to discover suitable business processes for automating them with RPA software in the whole organization.

Since RPA is new for the case company and its employees, there is a knowledge gap in understanding the capabilities of RPA and how to find suitable processes for RPA. This understanding gap is realized when business units suggests processes for RPA team of case company, usually these processes are vaguely described and lack good understanding the process flow and rules. Key to correctly model the business process automation, is to understand current as-is business process correctly (Mendling et al. 2012).

1.2 Research Questions and Objectives

Case company's challenge is how find suitable processes for robotic process automation (RPA). Business process complexity would be used in assessing the suitability of RPA to automate the business process. To analyze the business process, it must be modeled and documented during process discovery. After the current as-is process is well understood, its suitability for process automation can be assessed (Dumas et. al., 2013. p. 155 & 316). This leads to our first and second research questions.

R1: How information regarding the business process can be gathered?

R2: What needs to be known of as-is business process before automating it?

RPA is rather newly emerged and rapidly evolving technology of which the case company had no prior experience. Because of lack of experience with RPA technology in the case company, there is a need of increase understanding of the capabilities of RPA as a technology. To be able to determine whether RPA as a technology is suitable and cost efficient to automate a business process, the complexity of the business process must be assessed in the context of RPA capabilities. The third research question is then.

R3: What are the capabilities of RPA?

With this study the case company hopes to achieve better understanding on how business process knowledge can be gathered for RPA project, how to measure business process complexity and how to more comprehensively understand the capabilities of the RPA software tool. The literature review will give the theoretical framework to analyze the empirical data. The main findings and validity of the study are analyzed in chapter 7, where future research topics are also shortly discussed. In the chapter 8 main conclusion of this research are then formed.

1.3 Research Methodology and Structure of Research

Literature was searched from Emerald Journals, Science Direct, RPA vendor documentation, and Google Scholar. The empirical part of the thesis is based data collected during an RPA implementation project in the case company. The literature review has been gathered by using qualitative methodologies.

At first the theory concepts are reviewed and then implemented in the empirical part of the thesis. Theoretical framework is gathered by using qualitative methodologies and literature review is focused on understanding the business process development, process modeling and measuring the business process complexity.

The first chapter of literature review is aimed to gain understanding of business process development. The focus is in the business process management framework, which proposes the methodologies and structure for business process development. In the second chapter of literature review business process modeling is studied in general and in the third chapter, the measures for process complexity are examined to determine which elements of the business process model should be highlighted when estimating process complexity. The fourth chapter of literature review examines the current capabilities of robotic process automation. In the empirical research, the theories from literature review are implemented to understand and solve the found issues from robotic process automation project in the case company.

2. BUSINESS PROCESS MANAGEMENT

In this chapter focus is on Business Process Management (BPM) and what methodologies it introduces for gathering business process information. The three levels of BPM are examined and their purpose in the BPM are told.

2.1 Business Process Management Framework

Business process management focuses on improving business processes by using various methods and tools. The business process is first discovered as it is currently, then this process is analyzed and redesigned. After redesign, the business process is implemented and moved to every day work. During the process execution its performance is monitored, and results of monitoring are compared to expected results. Business process which has gone through the BPM lifecycle should result in lower costs, improved quality and increased efficiency (Dumas et al. 2013, p. 5-6; Mendling et al. 2017; Lindsay, Down & Lunn. 2003).

Business process improvement and changes are done when current business process doesn’t meet its expectations. These expectations may change when business process environment goes through changes (Dumas et al. 2013 p. 21). The BPM framework has three levels to describe the business process management. In figure 1 the chosen business process moves from multi process management level as an input to process model management level, where process goes through discovery of as-is business process and remodeling for to-be business process. To-be business process model is then fed as input to process instance level, where it is implemented in the every-day work in the organization (Mendling et al. (2017).

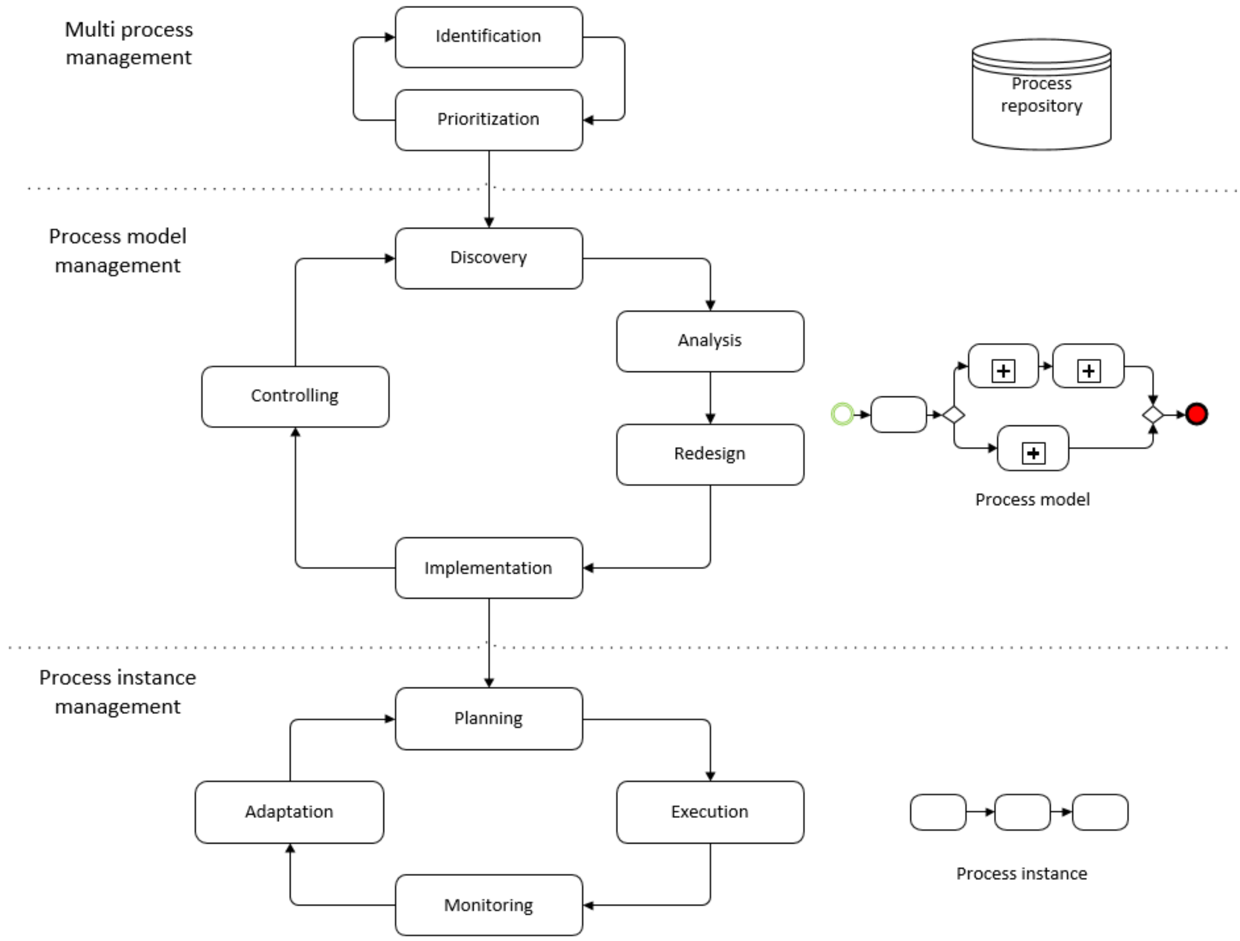


Figure 1. Business Process Management framework (Mendling et al. 2017)

Information technology (IT) is key instrument in BPM to achieve business process improvement needs (Dumas et al. 2013 p. 23; Meidan et al. 2017). When developing and configuring an information system (IS) to support business and business processes, it is important to have a clear understanding of three key elements; data, business process, and business rules (Kovacic 2004).

2.2 Multi Process Management of BPM Framework

The core processes of the organization are managed at the multi process management level. In this level the core process architecture is either updated or created and then stored in the process repository. Process architecture describes the core process at high level and how processes are connected to each other in the organization. The process architecture is documented in the process repository of organization (Mendling et al. 2017). When organization finds an problem in their business process, this found problem and its description serves as an input for process identification. Process or processes that are relevant to this problem are then selected, scope of process improvement is defined and relationship between each process are linked. Prioritization forms the input to process model management, prioritization is formed by measuring the process performance and considering the process importance with the strategy of the organization (Dumas et al. 2013 p. 21-22; Mendling et al. 2017).

2.3 Process Discovery of Process Model Management

When process goes to Process model management level, where target is to, document, analyze, and redesign the as-is process to to-be process model, which then will be implemented in the process instance level. In the controlling state of process model management, the implemented business process performance is measured and observed if it meets the expectations. If expectations are not met, then business process is selected for further improvements (Mendling et al. 2017). At first the as-is process is documented by using various discovery techniques which are; Workshop-Based, Evidence-Based, and Interview-Based discovery (Dumas et al. 2013 p. 161; Mendling et al. 2017).

In Workshop-Based discovery a workshop is facilitated with domain experts who can have various roles in the business process, process owner, and business process analyst. Workshop-Based discovery provides rich information of various aspects of business process. Risk here could be opportunistic behavior to not tell essential information about the process for the analyst. Ideas and perceptions of how process works, might give incorrect information about business process. Characteristics of people and culture of organization might affect what is told during workshop. It is seen that, open culture in organization enables informative discussion about the process (Dumas et al. 2013 p. 164-166).

Since process knowledge is usually fragmented and known by different domain experts, it is useful to use Interview-Based discovery technique, where domain experts are typically interviewed individually. Usually same domain expert is interviewed again after first draft of process model is done. Domain experts tend to explain only the so called “sunny-day” scenario of process, this means that exceptions might not be told. Risk of loss of power or position might affect what is told about process (Dumas et al. 2013 p. 162-164 & 166).

Evidence-Based discovery is highly objective and used methods are observation of work, automatic process discovery and analyzing available documents of business process. Documents might not give up-to-dated information of the process and they might not be trustworthy. Granularity level of information in documents varies from too abstract to fine-granular level. On fine-granular level the process steps are described in micro-step level, such as “clicking save button”. Observation is done by watching when process participant works with the process. Risk is that observation affects peoples' behavior, since people tend to change their behavior when people know that they are being observed. This changed behavior might make them to work in different way as they usually would. For automatic process discovery, also known as process mining, event data from information systems logs are mined. This information tells what activities were performed and the point of time of activity for each case. Automatic process discovery requires tools, that might not be available for analyst to use. The event logs can provide misleading information if they are not stored correctly, which leads to wrong process model generated (Dumas et al. 2013 p. 161-162, 165 &166).

The outcome of process discovery is the as-is process model documentation. The as-is process is used to analyze the process with quantitative and qualitative techniques to find out process complexity, weaknesses, and improvement suggestions for redesign phase. Redesign of business process is done before it can be transformed into workflow model for Business Process Management System (BPMS), in chapter 5 the BPMS is examined in more detail. When workflow model, that serves as executable process automation model, is finished it is then implemented in the every-day work in the organization (Dumas et al. 2013 p. 22, 155 & 316; Mendling et al. 2017; Cardoso 2008).

2.4 Process Instance Management

Process instance management level is where management considers only one of process instances at a time. Planning state is scheduling the activities and which resources should be involved. Execution of process instance is done according to its schedule and process model, if there are no schedule then execution is done without it. Execution of process instance is done according to the defined rules in the process model. Process monitoring checks, that rules are followed, and process performance is measured. Alerts are triggered if rules are not followed and undesired behavior is recognized in the process execution for individual process instance case. Adaptation is reacting for the results of single process instance, such as alert might require reacting for the changed process execution flow (Mendling et al. 2017).

3. BUSINESS PROCESS MODEL

In this chapter at first process models purpose and process model building elements are explained. Then process model complexity metrics and perspectives to measure business process complexity are presented. After reading this chapter, reader should know how to model a well-structured, understandable, and analyzable process model.

3.1 Business Process

Start event triggers one or more activities that involves actors who are performing activities. Actors can be human, also known as process participant, applications, or organizations. Objects can be involved with actors in the business process. Objects have two diverse types, physical and immaterial. Actors sometimes are required to make decisions during the process, these decisions control the executed activities in the process. When the outcome of business process is ready, the business process is at the end event. In every business process there must be at least two events, the start and the end events. Finally the outcome is consumed by a customer. If outcome is positive, then the customer receives value from it. In case the outcome is negative, the expected value is not received by the customer. Figure 2 depicts the core ingredients of business process (Dumas et al. 2013, p. 4-6, 24; Gruhn & Laue, 2009).

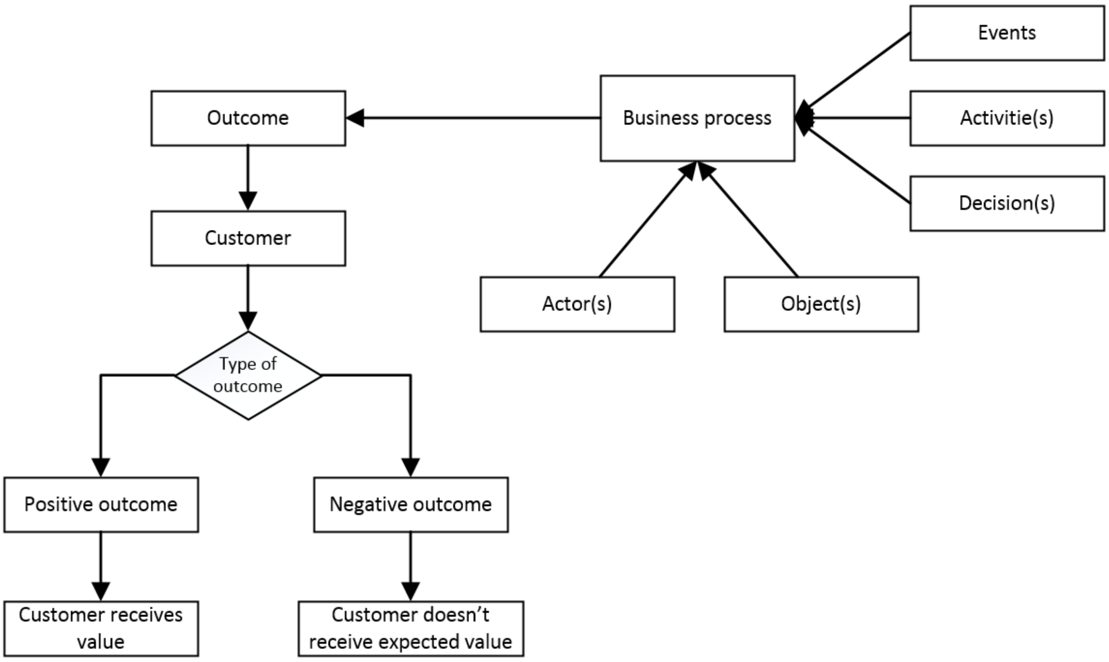


Figure 2 The ingredients of business process (Dumas et al. 2013, p. 6)

3.2 Modeling the Business Process

Process models have many purposes, of which the most important one is the understanding of process, sharing of process knowledge, and identifying and preventing issues. This is also prerequisite for process analysis, redesign and automation of business process (Dumas et al. 2013 p. 62-63; Lindsay et al. 2003). Processes are usually explained in visual process model, but processes can also be explained in structured or plain text format, or with the help of a computer simulation, or computer readable file. As-is model of process explains the real current working process and to-be model of process explains the proposed process (Polančič & Cegnar 2017).

Process modeling has multiple visual notation languages, which are used to form graphical explanation of process flow. These include notation languages such as Business Process Model and Notation (BPMN), Unified Modeling Language Activity Diagram (UML AD), Event Driven Process Chain (EPC), and more. From these languages, two have been standardized by International Organization for Standardization, these notation languages are BPMN and UML AD (Polančič &

Cegnar, 2017). Figure 3 is modeled in BPMN language. In BPMN there are five categories, which are flow objects, connecting objects, swimlanes, artifacts, and data objects (BPMN 2.0, 2011. p. 27). These categories will be examined in chapters 3.2.1 to 3.2.4.

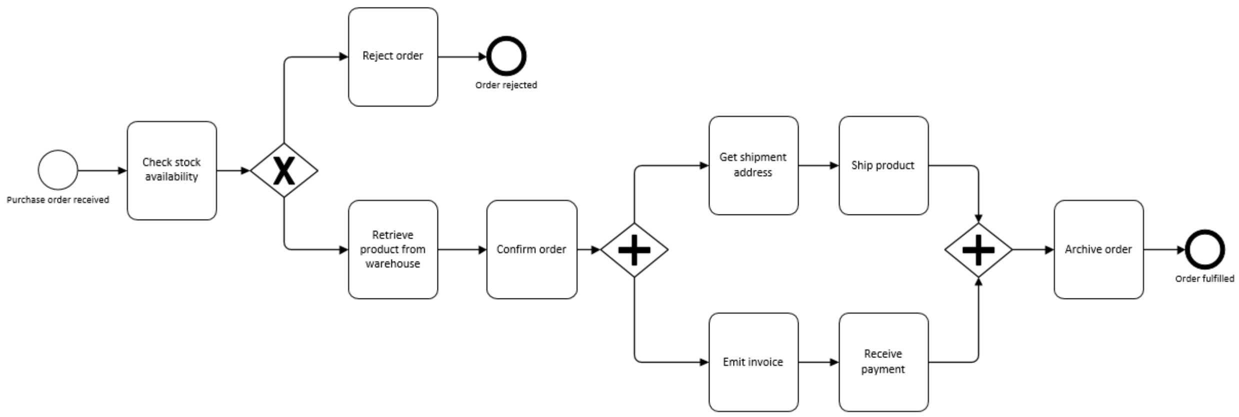


Figure 3 An example of process model in BPMN language (Dumas et al. 2013, p. 71)

A process model has usually two aspects, organizational aspect and IT-aspect, organizational aspect of process model is used for communicating, understanding and improving the business process. IT-aspect of business process is more detailed and has enough information for process automation or system development (Dumas et al. 2013 p. 63-67, and 108). The BPMN modeling language is current standard for modeling the business process from these two aspects (Chinosi & Trombetta, 2011).

3.2.1 Modeling the Flow Objects in BPMN

Flow objects are where all the action in the process is happening. In process model the action is happening in events, activities, and gateways (Chinosi & Trombetta, 2011). Start event of flow objects generates at start the token and in the end event the token is destroyed. Token is used to describe in which state the process is going (Dumas et al. 2013, p. 64). Basic BPMN flow objects are depicted in table 1 with their names and format in which they are in BPMN.

Events are something that happens instantaneously, and thus they do not have a duration. As an example of event could be receiving a message. Start and end event can have also other kind of triggers. For example, a message or a schedule can start the process and exceptions might lead to terminating end event (Dumas et al. 2013 p. 64-65, 110, 115-116).

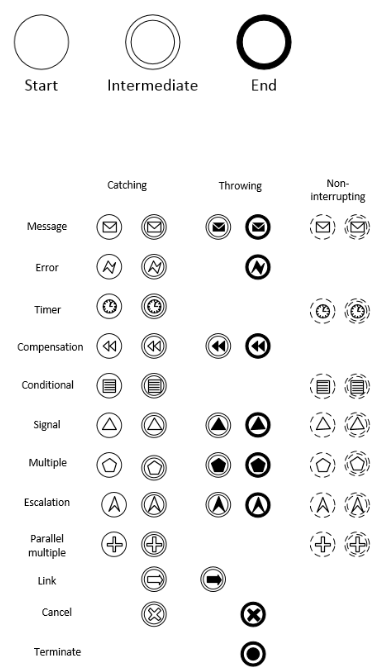
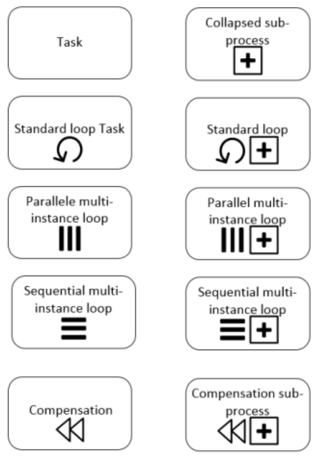
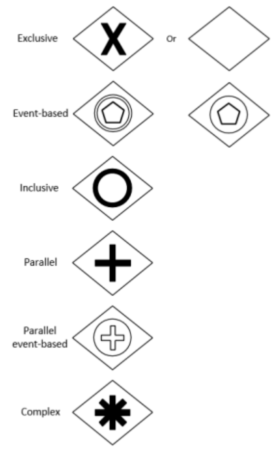
Intermediate event is used to describe an event occurring between the start and end events of a process. Intermediate event can be displayed as catching or throwing event, where white marker displays catch event and black marker displays a throw event. Catch event means that the trigger of event is catch usually from outside of the process and throw event means that the trigger of is sent from a preceding event within the process. Such intermediate event, as message event represents only sending or receiving message and not writing or reading it. Some events are only catching events, as timer event can only catch the specified time unit. Sometimes event can be a non-interrupting event. This event doesn't stop the current activity but may require additional activities to be executed. These types of events have dashed double border in a form of circle. Non-interrupted intermediate events are usable in complex exception handling modeling (Dumas et al 2013 p. 108-110 & 119).

Activity represents a task or sub-process. Activity has a duration and units of work to be completed before token can move forward in the process. Activities are displayed as rectangle squares. Task activity represents the unit of work which can't be decomposed into smaller units of work. When activity element has a boxed "+" it represents a sub-process. A sub-process can be broken into smaller units of work which has same particular goal or produces a specific outcome (Dumas et al. 2013 p. 64 & 97-98).

Modeling a loop block is possible with a gateway and activities or by loop activity. A sequential loop is modeled, when loop activity instance is executed one after another. Task or sub-process activity which has loop marker in it means a sequential loop. When loop activity instances are executed at the same time, it is called a parallel loop. A parallel loop is modeled with three vertical lines in sub-process or task activity (Dumas et al. 2013, p. 103 - 106).

In a recovery process it is required to undo one or more steps. This is depicted as compensation activity in BPMN. Compensation models the activities which should be done to achieve the desired business process state before the sub-process was interrupted by an event that started the compensation branch of process (Dumas et al. 2013, p. 122 – 123).

Table 1 The basic Flow Objects of BPMN 2.0 (BPMN 2.0, 2011. p 29 – 40).

<table>
<tr><th>Events</th><th>Activities</th><th>Gateways</th></tr>
<tr>
<td>
</td>
<td>
</td>
<td>
</td>
</tr>
</table>

Gateways are used to split and join the process in alternative mutually exclusive or parallel concurrent sequences. When gateway is split it has one sequence flow as going in it and multiple sequence flows as outgoing. A join gateway has multiple incoming sequence flows and one outgoing sequence flow. When process token arrives in the gateway it is decided then according to stated condition of gateway which sequence flow or flows of process token takes (Dumas et al. 2013, p. 67). Different types of gateways are displayed in table 1.

Exclusive gateway, also known as XOR split and join represents a split where process token takes only one of the possible process sequences and goes through other activities in the process sequence until it is joined in the XOR-join which allows the process token to continue after one of the possible process sequences has been completed. XOR-split has also event-based split possibility, which is used for example when a racing event is modelled. When two external events are racing against each other, the scenario can be displayed in the process model as a racing event. The conditions in the split which occurs first, determines the execution path of the token in the process. Difference between XOR split and event-based XOR split is, that event-based XOR split decision is based on process environment where as XOR split is internal choice which is based on output data of a decision activity. After an event-based exclusive split, the modeler must use only intermediate catching events or receiving activities. In figure there 4 is an example of racing event modeled in the process model in BPMN 2.0 language. Since there isn't event-based XOR join available, XOR join is used to display join. XOR split is displayed as diamond which has black "X" in it and event-based XOR split as diamond which as white pentagon inside a circle in it. (Dumas et al. 2013 p. 67-69 & 111-112).

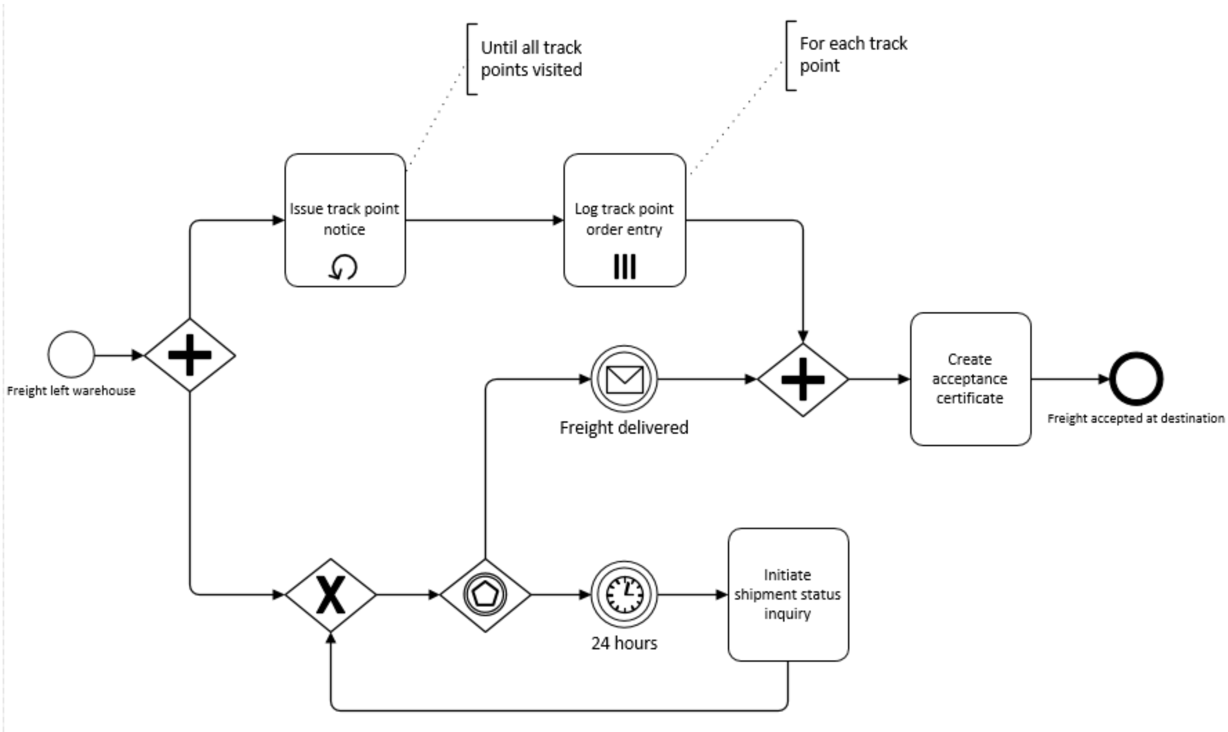


Figure 4 Example use of XOR and event-based XOR splits and XOR join (Dumas et al. 2013, p.112)

When two or more activities does not exclude each other, and these activities occurs at the same time. A parallel gateway split and -join are used to display this type of branching in the process, also known as AND-split and -join. This split and join are displayed in the figure 4 as "+" sign in the middle of a diamond. Both branches of process must be completed before the token can proceed further in the process from AND-join (Dumas et al. 2013 p. 69 – 71).

When two or more activities that may occur at same time and these activities does not exclude each other, in this type of case an inclusive split and join gateway is used. Inclusive gateway split and join are also known as OR-split and -join. When process token comes to OR-split, the OR-split generates as many tokens as there are true conditions for the outgoing branches. Both OR-split and XOR-split haves usually a default sequence flow, which is taken when the results of the other conditions are false. Process token can continue when all tokens generated in the OR split reach the OR-join. OR-split and -join is usually considered as hard to understand and it is recommended to use only when there are no other options. OR-split and -join are displayed in the figure 5 as diamonds which has black circle in it (Dumas et al. 2013 p. 73 – 76).

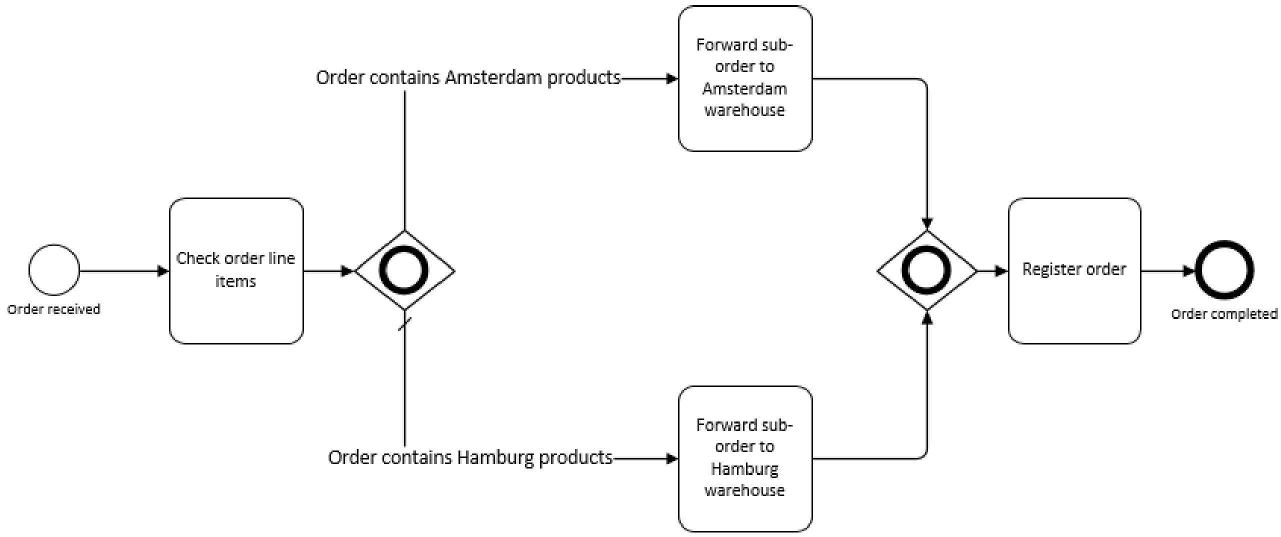


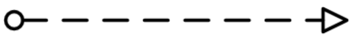
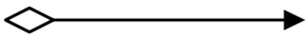
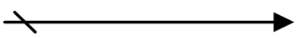
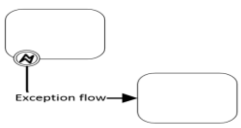
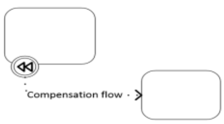
Figure 5 Example of OR-split and -join gateways in BPMN language (Dumas et al. 2013, p. 74)

3.2.2 Modeling the Connecting objects in BPMN

The flow objects can be connected by eight different elements in BPMN. These elements are normal flow, uncontrolled flow, default flow, message flow, association, compensation association, exception flow, and data association. Normal flow is the most typical way to connect two flow objects in the process model, but when connection is to be made between two pools, pools are described in chapter 3.2.4, then message flow must be used. Uncontrolled flow depicts a sequence flow which has no condition to control it. This occurs when multiple flows are outgoing from an activity. A message flow can't be used as a connector between two flow objects inside same pool. Association flow is used to link information and artifacts to elements. Data association is an association flow with an arrow in it. The arrow depicts the data flow direction between flow objects. When sequence flow starts from intermediate error element that is attached to a boundary of an activity, the sequence flow is called as exception flow. Compensation association triggers from catch or throw type intermediate compensation event, and it occurs outside the normal flow also as an exception flow (BMPN 2.0. 2011, p. 27, 34 – 35, 40 & 42 – 43). All these different sequence flow element types are illustrated in table 2.

There are other possible forms for a sequence flow as well. The default sequence flow from an exclusive or inclusive data-based gateway, complex gateway or activity it has a small line over it. The default sequence flow shows the path of the process token, which is taken when other outgoing sequence flows can't be taken. A conditional sequence flow can only be used from activity and not from gateway. It has an expression condition in it that controls that the sequence flow is taken by process token only when the condition is true (BPMN 2.0. 2011, p. 97 – 98).

Table 2 BPMN connecting objects elements (BPMN 2.0. 2011, p. 34 – 35)

Element name	Graphical form of element
Normal flow	
Uncontrolled flow	
Association	
Message flow	
Conditional flow	
Default flow	
Exception flow	
Compensation flow	

3.2.3 Modeling the Data objects and Artifacts in BPMN

A data perspective is needed to understand what kind of data is required and formed when process is executed. This is important to understand when communicating with system developers when target is to develop an application or to automate a process. A data object, as they are called in BPMN, can be in physical or digital form. An example of a physical data object is invoice letter and an example of a digital data object is an Excel workbook. In figure 6 data objects are displayed as documents which have a folded upper-right corner and they are associated to activities with dotted lines which can have arrow heads in them. As data objects model the information flow between activities, sometimes also the state of data object needs to be modeled. A state of data object as seen in the figure 6 is “purchase order” and “confirmed purchase order”, in this model the state of a data object changed, and thus it was needed to model in the process model. When data object has three vertical lines in it, this means collection of data objects (Dumas et al 2013. p. 66- 67 & 79-81).

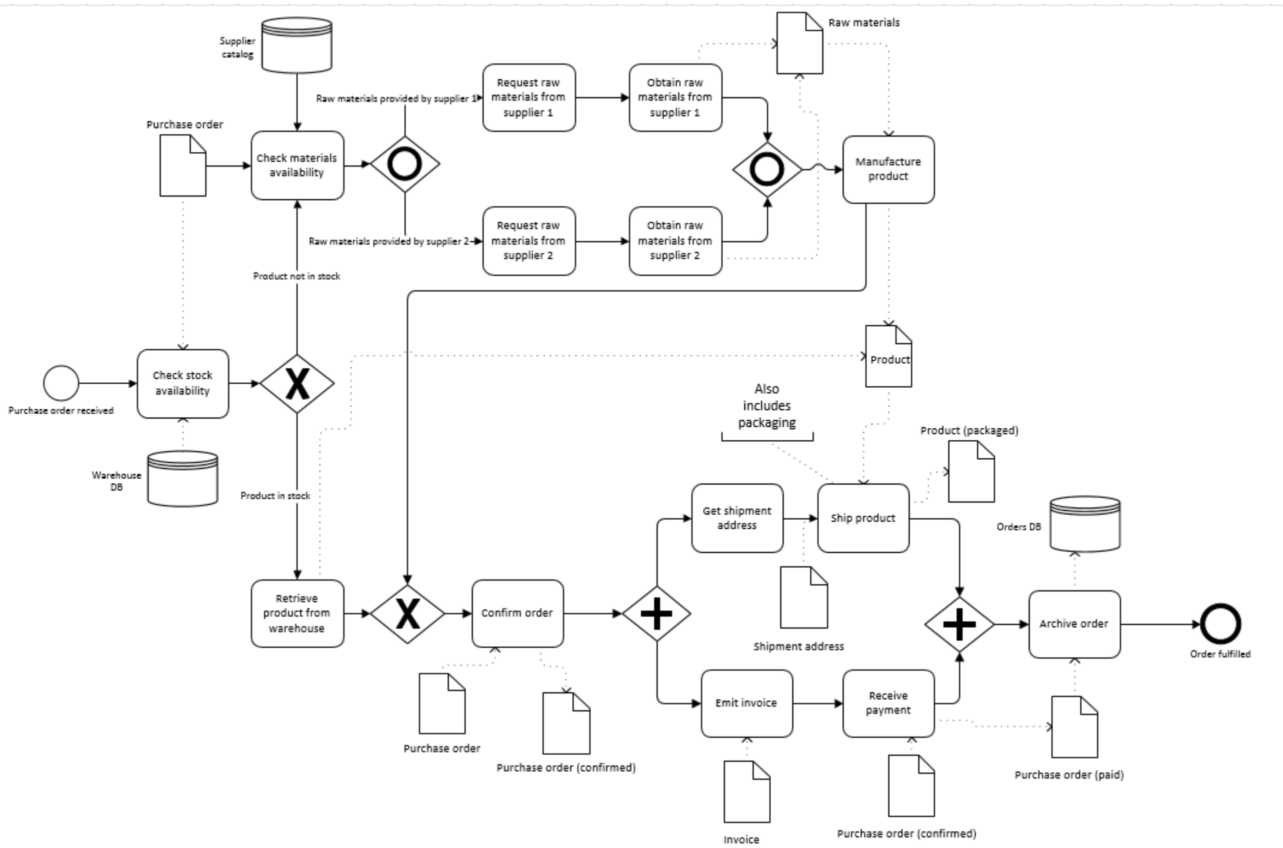
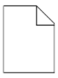
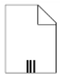
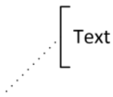


Figure 6 Modeling Data objects and Artifacts in BPMN language (Dumas et al. 2013, p.80)

A data store is used when data objects must be stored beyond the process instance. A data store for digital data objects can be for example a network drive, and a file archive can serve as a data store for physical data objects. Activities in the process can read or write data objects from or to data storages. In BPMN the data store is an empty cylinder and it can be linked to activities with an association line, as seen in figure 6. If a data object is used as an input, it must be generated before the activity can be performed that requires it in order to perform the activity. (Dumas et al. 2013 p. 81).

Artifacts are used to give additional information about business process. There are three artifacts in BPMN. Typical artifacts are group and text annotation. Text annotations are connected to the flow objects with association (BPMN 2.0. 2011, p. 66 – 69). These artifacts can be used for example to describe a business rule of the element in which it is connected. In figure 6 text annotation describes the business rule connected to “Read stock report” task activity. However, artifacts do not influence the process flow. Artifacts are only used to provide additional information for the process model reader (Dumas et al. 2013, p. 82). In the table 3 the data objects and artifacts used in BPMN are depicted.

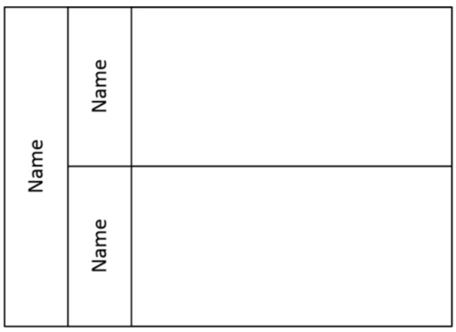
Table 3 Data object and Artifact elements in BPMN language (BPMN 2.0. 2011, p. 30, 36, & 393)

Element name	Graphical form of element
Data object and collection of data objects	 and 
Data store	
Text annotation	
Group	

3.2.4 Modeling the Swimlanes in BPMN

When in the process model who or what performs an activity is needed to model, the resource perspective is then modeled. Another name for this perspective is organizational perspective. Resources have three main activity performers; a process participant, a software system, or an equipment. A process participant is human process performer. A software system is for example an application. An equipment is for example printer or manufacturing plant. Resources can be divided further into two categories, which are active and passive resources. When someone performs an activity, this is called an active resource whereas if a resource is used to perform an activity it is called a passive resource. For example, when process participant prints out a paper from a printer, the process participant is performing printing activity and a printer is used to do this activity (Dumas et al. 2013 p. 82). These resources are modeled as swimlanes in BPMN. Pool and lane elements belong in the swimlane category (BPMN 2.0. 2011, p. 28). In table 4 are the pool and lane elements are separately shown.

Table 4 Swimlane elements in BPMN (BPMN 2.0. 2011, p. 30)

Element name	Graphical form of element
Pool	
Lanes	

Inside the pools and lanes, the activities, gateways, data objects, and events are placed. When activity is placed in the lane, the resource is also the performer of that activity. For example, if we put an activity in a lane which is a system resource, it means that system performs this activity automatically. BPMN does not describe how to put resources in pools and lanes so, it needs to be decided when modeling the business process which type of resources are put in the pools and lanes. A pool usually represents the organizational perspective of business process performer, such as IT organization or sales organization. When there is required communication between two pools, such as customer and sales organization, this communication is displayed with message flows. Lanes usually represent the systems and departments of the business organization that performs the activity, such as ERP system or warehouse. In figure 7 resources are depicted in the BPMN language (Dumas et al. 2013 p. 83-89).

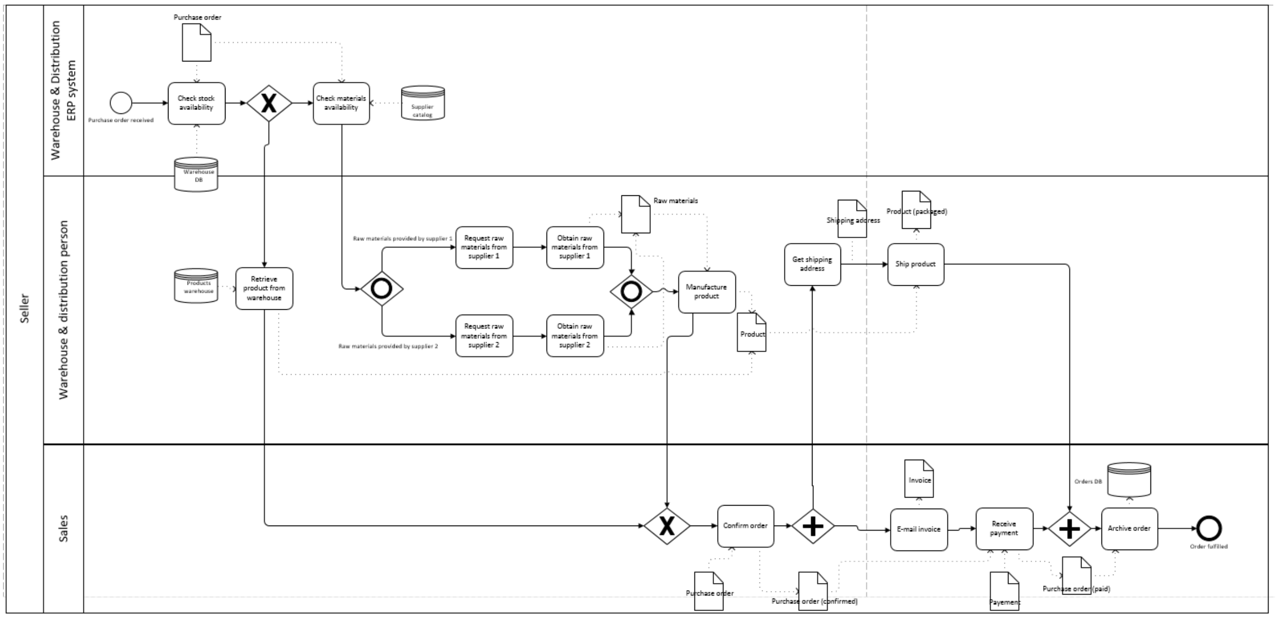


Figure 7 resources modeled in BPMN (Dumas et al. 2013, p. 84)

3.2.5 Modeling the exceptions in BPMN

Events which puts the process flow on a different course than normal “Sunny day” course, are called exceptions. Exceptions which occur frequently, should be modeled in the process model. There are two categories of exceptions, business and technology. Business exceptions occur, when something originating from the business side has gone wrong. An event like this could be for example, that someone has forgotten to order stocked items and company has ran out of stock. Technology exceptions are things like an application crash and network connection lost. These exceptions may interrupt the process and causes delays or even abort the process (Dumas et al. 2013, p. 114 – 115).

The easiest way to handle an exception in business process, is to abort the process. The abortion of process is modeled as terminate event, which destroys all process tokens in the current process instance. The abortion is effective at current process level and sub-processes. Terminate event, also known as abortion event is an end event with black full circle in it as seen in figure 8 (Dumas et al. 2013, p. 115 – 116).

Other ways to handle exceptions are internal and external exception handling. In BPMN the internal exception is caught by intermediate error event element, which has two different semantics, throw and catch. Throw error event interrupts the sub-process at hand and throws an exception which is then caught by a catching intermediate error event. The catching error event is modeled in the boundary of the sub-process and triggers the exception flow of the process. During the exception flow, it is possible that it ends in the abortion of process instance or it returns to normal process if exception could be handled during the process instance. Internal exceptions are modeled as seen in figure 8, where the throwing end error event has a black lighting in it and the catching intermediate error event has a white lighting in it. When intermediate timer event is modeled in the boundary of an activity, this is used to model the timeframe that is allowed for an activity to be processed. If time limit is reached, it throws an exception (Dumas et al. 2013, p. 116 – 118).

External exceptions are caused by an external event which occurs during the activity. External event might be a message from customer who wants to cancel the order, or a signal given from application. In figure 8 these different external exceptions are shown. When external exception is caused by a message, it should be decided where in the process this is possible to occur. As in internal intermediate error event, the black colored symbol means throwing and white colored symbol means catching. External exceptions also trigger the exception flow to recover the process (Dumas et al. 2013, p. 117 – 120).

Figure 8 Exceptions elements in BPMN (Dumas et al. 2013, 117 – 120).

3.3 The Process Modeling Guidelines

A key requirement for a process model is that, it should be understandable. Following the seven process modeling guidelines (7 PMG), as outlined in the table 5, while designing the process model will increase its understandability and error possibility decreases (Mendling et al. 2010). When process model is well structured, understandable, and comparable then it is possible to analyze the process (Dumas et al. 2013 p. 175-176; Mendling et al. 2010). Also, other business process modeling guidelines exist, Corradini et al. 2018 has identified 50 guidelines for BPMN. These 50 guidelines include the mentioned 7PMG guidelines but has also more specific guidelines relating to BPMN process modeling notation language. From Corradini et al. 2018 work, two more guidelines are added in the 7 PMG. These guidelines are “Associate data objects consistently” and “Model loops via loop activities”. In table 5 these guidelines are combined with original 7 PMG and together they form nine process modeling guidelines (9 PMG).

Table 5 Nine process modeling guidelines (Mendling et al. 2010; Dumas et al. 2013 p. 176 – 177 & Corradini et al. 2018).

Guide line (GL) number	Guide line name	Guide line explanation
GL 1	Use as few elements as possible	Large models are difficult to understand. Usually this leads to more syntactical errors, and process related errors. A good way to avoid too many elements is using sub-process activity.
GL 2	Use as few routing paths as possible	Activities should have only one outgoing sequence flow, when this is not possible. The number of outgoing sequence flows should be as few as possible.
GL 3	Use one start and one end event	In the process model there should be only one start event and only one end event. If the number of start and end events increases, this will lead to increased error possibility in process and decreases the understandability of process model.
GL 4	Model as structured as possible	If every gateway split match with same join gateway connector model is structured. If different gateway splits and joins are used together in the process model, this decreases the understandability of process model and leads increased error possibility in process.
GL 5	Avoid OR-gateways	Avoiding OR-gateways, decreases the error rate in the process. Using XOR and AND gateways is preferred since they are easier to understand and model.
GL 6	Use verb-object activity labels	Activities which are labeled as verb-object style are easier to understand and understandability of what activity represents is increased.
GL 7	Decompose a model with more than 30 elements	Decomposing process model which has more than 30 elements will increase its understandability. Decomposed parts of process should still be modeled as sub-processes.
GL 8	Associate data objects consistently	Data objects should always be associated to activities and have a direction in their association flow.
GL 9	Model loops via activities	Loops should be modeled by using loop activity, and not by use of sequence flows and XOR gateway.

3.4 Business Rules for Business Processes

A business rule can be viewed from two aspects. The business aspect and IT aspect. From the business aspect a business rule can be explained as a rule that implements a practice or pattern which instructs process participant to behave correctly while executing the business process (Dumas et al. 2013 p. 124; Graml et al. 2008). From the IT point of view business rule is seen as non-dividable business logic, which is reusable and explanatory (Graml et al. 2008). In the process model business rules are explained with text annotations, gateways and their joins, conditional activities, and as documented text which describes the process (Dumas et al. 2013 p. 82 & 142; Kovacic 2004).

Business rules have five different types how they can be distinguished. These types are derived out from the rule structure and from the purpose of the rule. BPMN offers a good capability to model these business rule types which are related to events, conditions, and transformation. In table 6 these business rule types are listed with short explanation (Zur Muehlen & Induska, 2010).

Since the business process is dynamic, control flow of the business process might need to change during the process instance or decision-making rules might require configuration during the process instance run. By adding business rules in the workflow model of process automation, this model becomes flexible and dynamic (Milanovic et al. 2011; Graml et al. 2008). Business rules gives the more detailed description of how activities are to be performed and decisions to be made in the business process. When business rules are developed in the workflow model, BPMS is then exploiting the knowledge of how activities are related to each other in the process (Kovacic 2004 & Dumas et al. 2013 p. 297-298).

Table 6 Five business rule types (Zur Muehlen & Induska, 2010).

Business rule type	Description
Integrity rule	This rule is expressing constraints in the business process, such as certain data can be only used in this part of process.
Derivation rule	Defines if condition is true or false. These rules tell the decision logic which will result in conclusions in the business process. Such as if customer is certain type, then customer will have discount.
Reaction rule	Defines the event that triggers the evaluation of this event and what activities should be performed after evaluating the condition of occurred event during the business process. As an example, when invoice is received, and its amount exceeds certain amount then manager must handle it.
Production rule	If certain condition is met during the business process given times, then specified action is made. As an example, ten materials have passed the quality check, all produced materials quality is approved.
Transformation rule	Rule that tells the possible changes of states in the business process. When certain condition is true, then process participant can proceed in the process to next state.

4. BUSINESS PROCESS COMPLEXITY

In this chapter the business process complexity measurement and thresholds are discussed. Target of this chapter is to build theory to measure the complexity and understand the complexity level of a process. Thresholds for business process complexity measures can be deducedd from process modeling guidelines. However, in each company the threshold values should be formulated from practical experience (Mendling et al. 2012).

4.1 Process Complexity Measurements

To understand process complexity, it must be measured. There are multiple perspectives that can be taken when analyzing process complexity and many metrics to measure it. Cardoso (2008) has presented four different perspectives for process complexity, which are Activity complexity, Control-flow complexity, Data-flow complexity, and Resource complexity. Polančič and Cegnar (2017) on the other hand are looking at process complexity from multiple aspects that should be measured. The aspects they have discovered are; analyzability, perceptibility, maintainability, explainability, correctness, understandability, required effort for understanding, clarity, testability, readability, and reliability. When measuring the process complexity from the point of view of automating the process with BPMS, then as Dumas et al. (2013) have explained in their book chapter 9.4, process control-flow, data-flow, resources used during process and activities should be known and modeled at a proper granularity level. The perspectives of Cardoso (2008) for measuring the process complexity are meant for measuring process complexity for business process automation.

When measuring business process complexity, its total number of elements should be calculated in each element category. These will form the base metrics of process complexity. In appendix 1 the base metrics are listed. Total number of all elements, connectivity levels of activities and pools, and proportion of pools and/or lanes and activities of process model should be calculated. These will give the structural complexity of business process. With sequence flows, default, conditional, normal and uncontrolled flows are viewed as sequence flow. With events, tasks, collapsed sub-processes, gateways and data objects, each of these should be calculated separately (Rolón et al. 2006).

After calculating the base metrics of elements in the process model the total number of element types and proportions are to be calculated. The derived measurement metrics and how they can be calculated from base metrics are presented in table 7 (Rolón et al. 2006). With the derived values, the business process control flow complexity, activity complexity, data-flow complexity, and resource complexity can be measured (Cardoso 2008). The derived measures are linked in table 7 to their complexity perspective.

Activity complexity measures the total number of activities in the process. Control-flow complexity measures gateway splits and joins, loops, and events. Resource complexity measures the required resources in the process, such as IT resources, human resources, and roles. Data-flow complexity measures data dependencies between activities and data usage complexities (Cardoso 2008).

Table 7 Derived process model measures (Rolón et al. 2006 & Cardoso 2008)

Derived measure	Formula	Complexity perspective
Total number of start events (TNSE)	TNSE = NSNE + NSTE + NSMsE + NSCE + NSLE + NSMuE	Control-flow complexity and Data-flow complexity
Total number of intermediate events (TNIE)	TNIE = NINE + NITE + NIMsE + NIEE + NICaE + NICoE + NILE + NICE + NIMuE	Control-flow complexity and Data-flow complexity
Total number of end events (TNEE)	TNEE = NENE + NEMsE + NEEE + NECaE + NECoE + NELE + NEMuE + NETE	Control-flow complexity and Data-flow complexity
Total number of events (TNE)	TNE = TNSE + TNIE + TNEE	Control-flow complexity and data-flow complexity
Total number of tasks (TNT)	TNT = NT + NTL + NTML + NTC	Activity complexity
Total number of collapsed sub-processes (TNCS)	TNCS = NCS + NCSL + NCSMI + NCSC + NCSA	Activity complexity
Total number of activities	TNA = TNT + TNCS	Activity complexity
Total number of gateways	TNG = NEDDB + NEDEB + NID + NCD + NPF	Control-flow complexity
Total number of Data objects	TNDO = NDOIn + NDOOut	Data-flow complexity
Connectivity level between activities	CLA = TNA / NSFA	Control-flow complexity
Connectivity level between pools	CLP = NME/NP	Resource complexity
Proportion of data objects as incoming product	PDOPIn = NDOIn / TNDO	Data-flow complexity
Proportion of data objects as outgoing product	PDOPOut = NDOOut / TNDO	Data-flow complexity
Proportion of data objects as outgoing product of activities	PDOTOut = NDOOut / TNDO	Data-flow complexity
Proportion of pools and / or lanes of the process and activities	PLT = NL/TNT	Resource Complexity

4.2 Weights of Process Complexity metrics

As the activities, events, gateways and data objects have diverse types, they can also be used to measure the level of complexity, for example in such way that XOR-gateway split and join is less complex to understand and model than OR-gateway split and join. This leads to giving different weighted values for different types of elements. In the table 8 there are presented the weights for each element describing how they increase the cognitive complexity of a business process more than just their given structural complexity which base value is one. The business process model cognitive complexity is the sum of its elements and their total cognitive weight values (Gruhn & Laue, 2006).

The OR gateway split and join has highest cognitive weight 7. It requires first a decision which branches should be executed out of two or more branches. Then these branches are processed in parallel. The next highest weight 6, is for multiple instances. This means that same activity or activities are executed at same time by different process participants. This has two steps in it, choosing which activities are to be executed in parallel and then processing these activities in parallel. Weight is constructed out of parallel AND gateway split and join, and XOR gateway split and join. Difference to OR gateway split and join is that in OR gateway split it is case sensitive and in multiple instances it is simple if – then branching decision. With cancelation activity complexity, depends on if it is only one activity or complete process instance that needs to be cancelled. If only one activity is required to be deactivated, then weighted cognitive value is 2. With terminating the whole process instance the value is 3 for cognitive weighted value as this requires multiple activities to be cancelled and might require more side steps in the process (Gruhn & Laue, 2006).

Table 8 Weighted values for BPMN elements (Gruhn & Laue, 2006)

BPMN Element	Corresponding software structure	Cognitive weighted value
Consecutive task activities	Sequence	1
XOR gateway split with two outgoing branches with corresponding join	If-then	2
XOR gateway split with three or more outgoing branches with corresponding join	Branching with case	3
AND gateway split and with corresponding join	Execution of sequences in parallel	4
OR gateway split with two or more outgoing branches	Branching with case and followed by parallel execution of sequences	7
Sub-process	invoke of a user-defined function	2
Multiple instance activity	branching that is followed by parallel execution of sequences	6
Cancel activity	Deactivating an activity, such as part of a sequence	1
Cancel case	Deactivating all instances in process or sub-process	3

5. TECHNOLOGY OF BPMS AND RPA

In this chapter the characteristics of BPMS examined and RPA is found as one type of BPMS. The UiPath RPA software is examined more closely, as it is the RPA platform of the case company. The target of this chapter is to increase the understanding of RPA and BPMS as technology.

5.1 Business Process Management System

A BPMS which uses process knowledge while executing process instances is called process aware information system (PAIS). In this thesis the BPMS is used to describe PAIS type of BPMS systems. The BPMS knows how activities in the automated business process relates to each other. Knowledge of the process is from a well specified business process model. After the business process model is designed to an executable process model, it is essential to keep it up to date so that the automated business process works as it is intended. With BPMS it is possible to automate anything from a small and simple piece of a business process to a full complex business process (Dumas et al. 2013, p. 297-299).

BPMS is capable to interact with systems without direct data connection. It invokes applications such as web and office applications and interacts with the applications Reichert & Weber, 2012, p. 27). This interaction with the application is performed in the user interface level (Dumas et al. 2013, p. 314).

Most activities that human perform with applications, can be automated with BPMS in electronic format. In figure 9 this interaction between BPMS and invoked application is depicted. The BPMS performs the activity while following the control flow of the process, interacts with application with given input parameters and then fetches data from application as output parameters. This depicts that BPMS is capable to react on the data perspective of the process; it can use the correct data type, read data values from application, knowing the object state and process state at the same time. BPMS has a time perspective when executing the process. Activities can be given deadlines and if activity deadline is reached then required notifications and escalations are executed (Reichert & Weber, 2012,

p. 27-30). A process aware BPMS can operate each work case that is fed to it as a separate case. If there are possible exceptions during any of the cases it won't have an effect on the other cases (Dumas et al. 2013, p. 314).

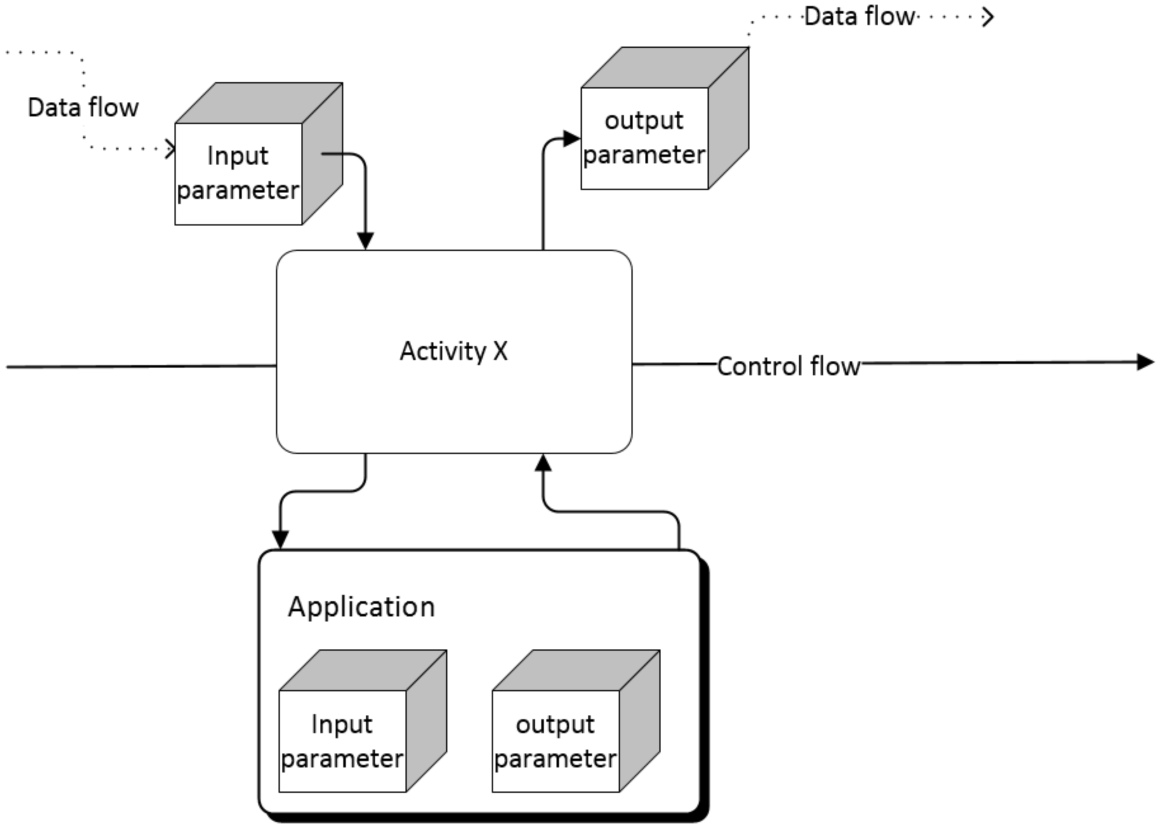


Figure 9 BPMS interacting with Application (Reichert & Weber, 2012, p. 28).

There are three types of data that is involved with BPMS automated processes. Data which is related to specific application is something that BPMS can't control. This is called application data, and it is managed by invoked application. Process states and state transitions are evaluated based on process-relevant data. With process-relevant data, BPMS is capable to decide correct execution paths in process instance (Reichert & Weber, 2012, p. 28). BPMS enforces the business rules, that are modeled in the workflow model which BPMS executes (Dumas et al. 2013, p. 312). Process control data is information about current process state and process instance execution history (Reichert & Weber, 2012, p. 28). BPMS increases the business process compliance while offering the current business process operational information and historical information of the business process case by case. This type of information is usually the foundation for process intelligence (Dumas et al. 2013, p 311 - 312).

When BPMS invokes the application and starts operating the workflow model based on the redesigned to-be process model, the software does not need any human interaction while executing the activities. BPMS executes activities according to its control flow. Control flow includes the sequences of activities, conditional splits, loops and parallel sequences. Control flow enables flexibility in the automated business process, by giving different branches in the workflow model. Data can be used by BPMS to make condition-based decisions in the control flow and execute the activities according to the data gathered during the process. This is also linked to the state transitions in the business process, where data tells the next state to proceed to in the automated business process (Reichert & Weber, 2012, p. 22-24). BPMS way to operate independently without human interaction can reduce mundane workload from employees (Dumas et al. 2013, p. 309).

BPMS does not need integration between the used business process supporting applications. Business logic is easy to model and configure in the BPMS and there is no need to reconfigure the business process supporting application. BPMS enables organizations to become more flexible in business process automation by allowing them to continuously develop their business processes and replace legacy business process supporting applications with newer applications. Applications that used to operate independently, can be integrated together with BPMS. The redundant information storage can be a problem, and it should be well investigated which data BPMS can use while working with the applications (Dumas et al. 2013, p. 310 – 311).

5.2 Architecture of Business Process Management System

BPMS is constructed of 4 main components and data repositories which can store execution logs and process models. In figure 10 there is an example of BPMS architecture. It is possible to connect external applications with BPMS tools. The four main components are process modeling tool, execution engine, worklist handler, and administration and monitoring tool (Dumas et al. 2013, p. 298 - 299).

A process modeling tool is used to design the automated business process model for BPMS which is then stored in the process model repository. These process models can be started in the modeling tool or from process model repository. Worklist handler component handles the process instance cases that are offered to it by process participants. Execution engine creates process instances, distributes cases to process participants, and automatically retrieves data and stores it. Execution engine monitors the progress of process instance cases and coordinates the activities to be taken by work handlers. External applications are interacting with the execution engine. When an external application is needed in the process, the execution engine invokes the application with correct parameters and retrieves a signal from it when the request is completed. Administration and monitoring tool is used to monitor the progress of live processes, review execution logs, analyze historical data and compare it to live data, and distribute jobs for process participants (Dumas et al. 2013, p. 299 – 302).

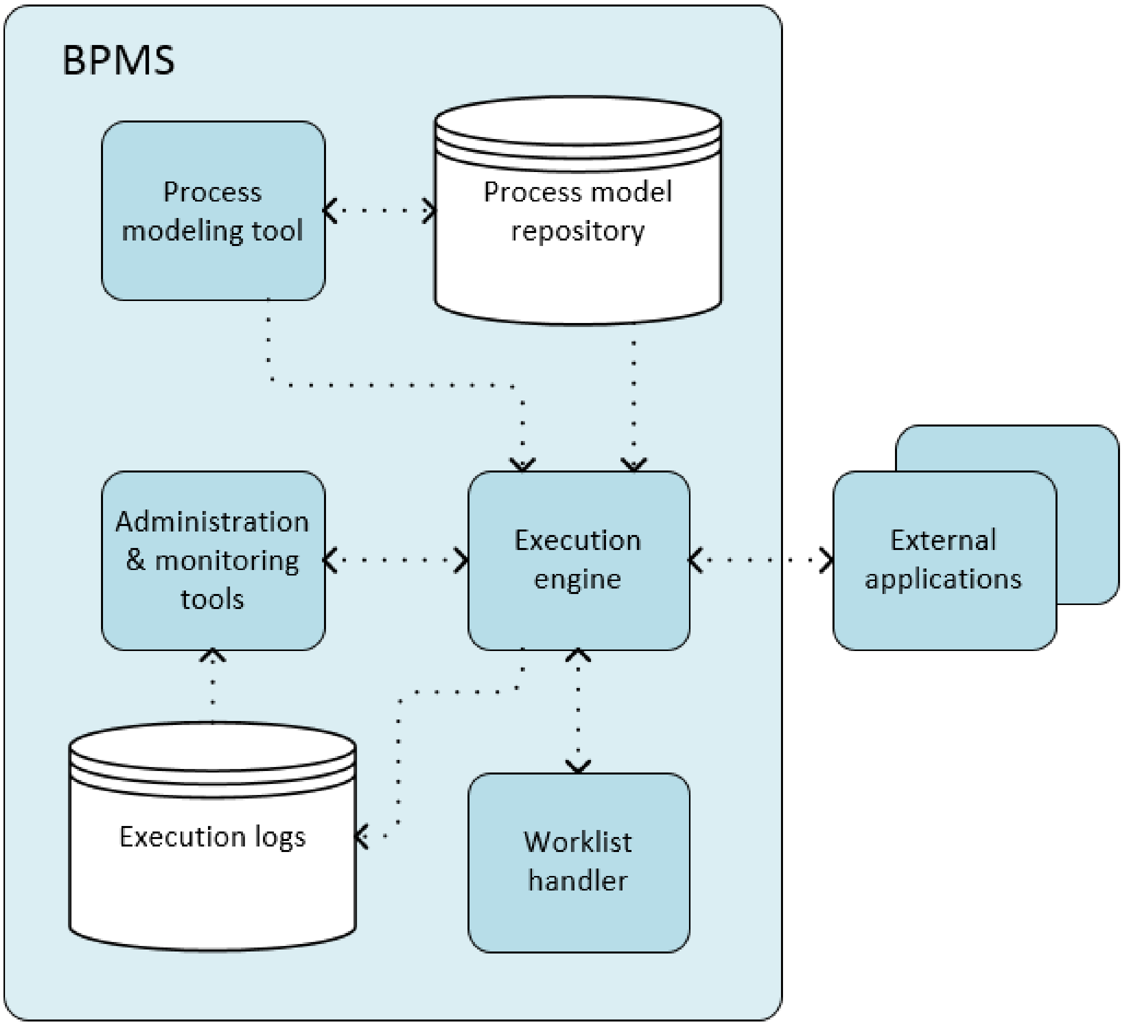


Figure 10 BPMS architecture (Dumas et al. 2013, p. 299)

5.3 Business Process Management Systems Types

Four different types of BPMSs can be identified when parameters used to distinguish the systems are how the process is structured and how the systems differ from each other on process and data orientation. These types are groupware systems, ad-hoc workflow systems, case handling systems, and production workflow systems. In figure 11 these systems are shown and how they are positioned between each other by the degree of support for structured process and if business process is process or data oriented. It is possible, that some BPMS are hybrid, and they can have functionalities from different types of BPMSs (Dumas et al. 2013, p. 306 – 308).

Groupware systems are used for sharing information in either documentation or communication form. Groupware systems do not support explicit business process notion. This lack of support for structured business process notion leads to groupware systems not being able to support business processes in automation of the business processes (Dumas et al. 2013, p. 307).

Ad-hoc workflow systems are highly flexible in process design and in configuring the automated process. It is possible to take a process model template and start modeling the process, even when process instance is being executed. This adds a lot of flexibility to the BPMS. It is possible to insert new steps in the process and modify the control flow. When using ad-hoc workflow systems, process definitions must be well documented and people who know the process should only be allowed to modify it. These people should also be highly capable to model the process and have tools for that (Dumas et al. 2013, p. 307 – 308).

Case handling systems allow deviation in the process model and are aware of the data in the process. With the data used in the process and gathered from the process, they are capable to inform process users of the process state and take steps in the process. It is possible to prohibit deviation during the process, but it must be explicitly modeled in the process model (Dumas et al. 2013, p. 308).

Production workflow systems are used especially when a substantial portion of work is currently performed by humans, but the aim is to automate a large part of that work. In a production workflow system an explicitly modelled business process is followed and any deviation in the business process must be defined and captured in the process model. This type of deviation could be an additional report out of BPMS (Dumas et al. 2013, p. 308).

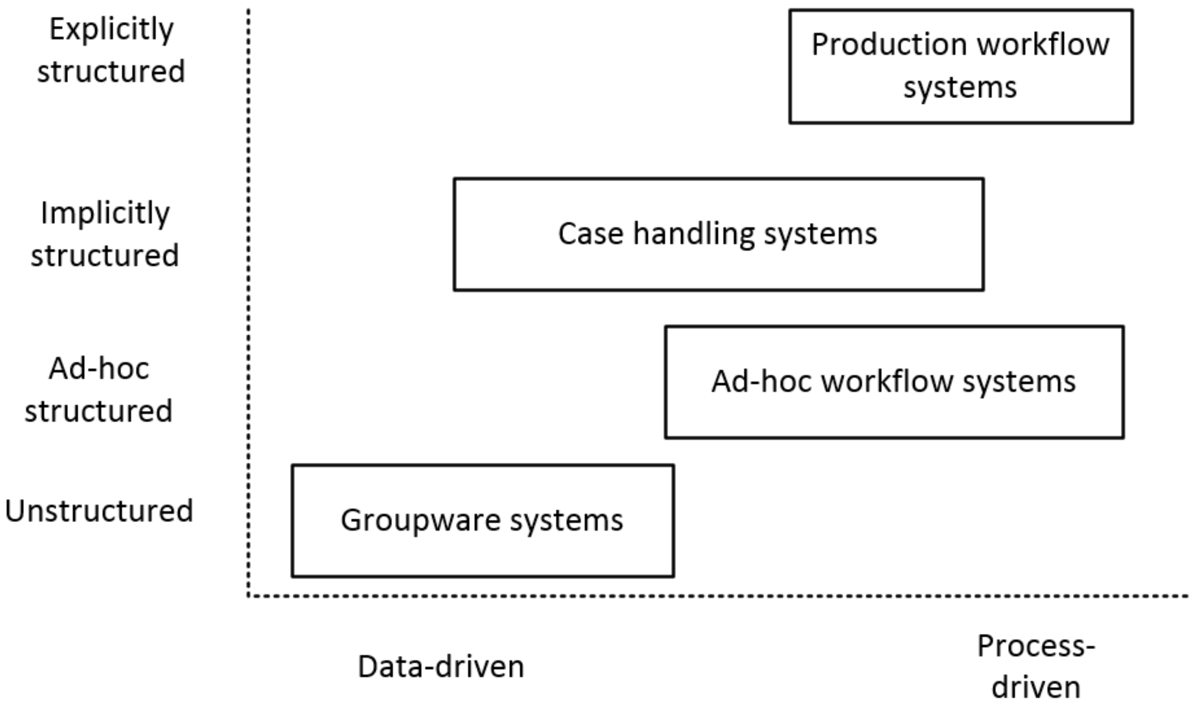


Figure 11 Types of BPMS (Dumas et al. 2013, p. 307)

5.4 Robotic Process Automation Software

RPA interacts with the presentation layer of applications, also known as user interface. It doesn't need the complex application or data layer integration. RPA tools do not require changes to used business systems when automating the business processes, and this makes the difference between classical business process automation methods and RPA software. RPA software has capabilities to use data to form decisions during the process instance. Some RPA vendors offer artificial intelligence capabilities in their products. Process mining tools can be integrated with some RPA software enabling automation of creating process model structures (Van der Aalst et al. 2018).

There are certain vendors in the market that focus purely on RPA, for example AutomationEdge, UiPath, Automation Anywhere, Kyron Systems, Softomotive, and Blue Prism. RPA is used to extend the business process automation to further and keep it economically feasible even when frequency of cases drops down. In figure 12 is depicted the area of business process characteristics where RPA software can be typically utilized. Traditional process automation is feasible when most of the cases follow same business process structure. RPA can be used to business process automation when business process is repetitive, but frequency of repetition is not enough for traditional process automation to be economically feasible. When business process is ad-hoc and has multiple complex exceptions to handle and it is not frequent enough, the process should be performed by human (Van der Aalst et al. 2018).

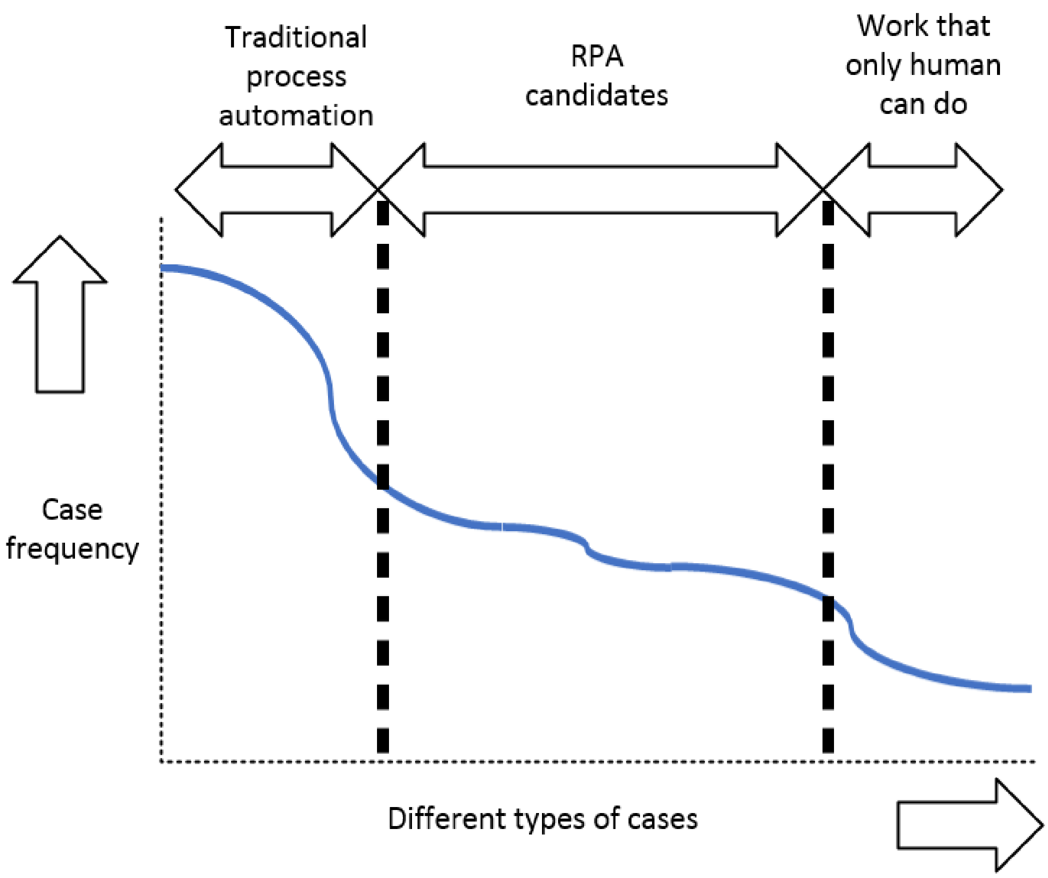


Figure 12 RPA feasibility to automate business process cases (Van der Aalst et al. 2018)

RPA performs the exact steps it has been modeled to execute in the business processes. If a business process is modeled wrong, it may lead to an undesired outcome from automated business process. Wrongly performing automated business processes are threats to organizations' performance especially if they are not been detected on time. This kind of threat increases when business process automation uses more artificial intelligence and it is not well monitored during the execution of the process instance cases because it can learn from hidden disastrous decisions and perform activities which will result in undesired outcome from the business process instance (Van der Aalst et al. 2018).

UiPath, the RPA software used in the case company, offers three software components which together form the full UiPath RPA platform. It is possible to add third party applications to this platform with application programming interface (API) connection as seen in the Figure 13. The three main software components are UiPath studio, UiPath Orcestrator, and UiPath Robot (UiPath, 2018b).

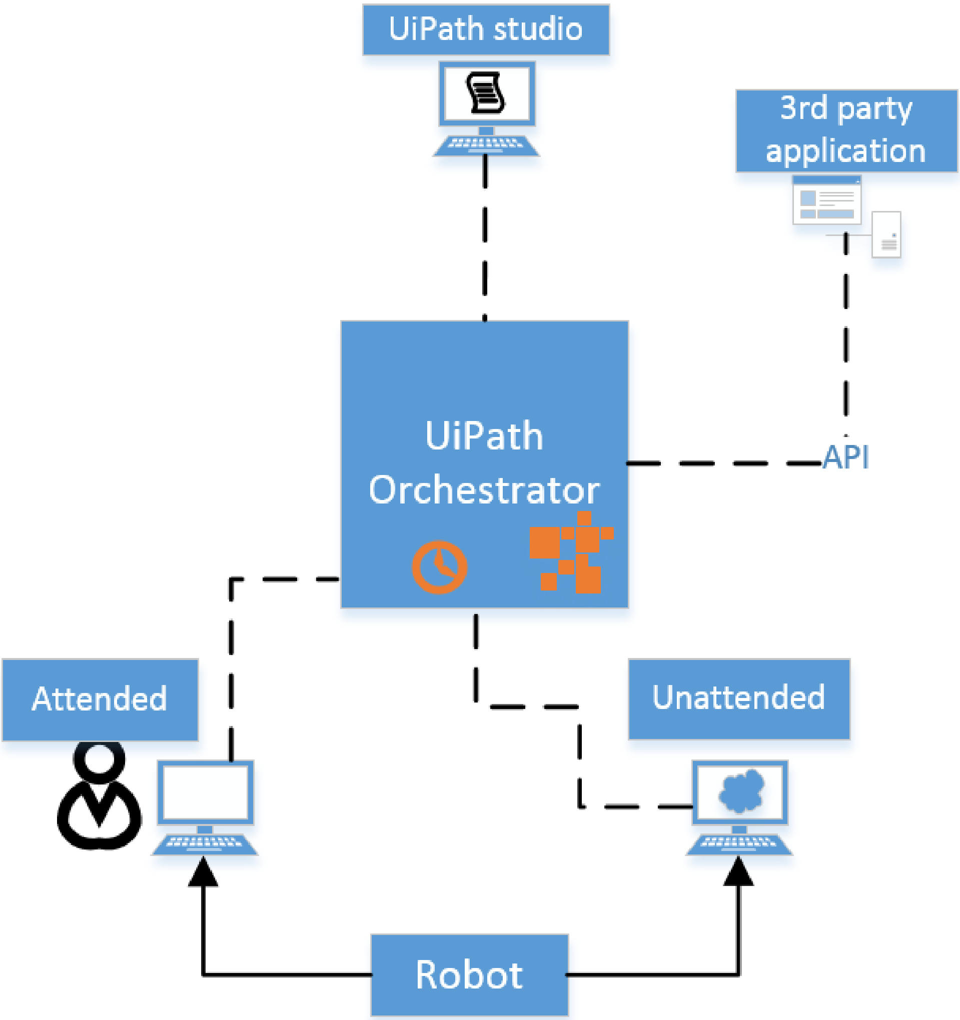


Figure 13 UiPath Platform (UiPath, 2018b)

UiPath Orchestrator is the administration and monitoring tool. Process models which are first built in UiPath Studio are shared through the Orchestrator for all robots enabling controlled updates of process workflow models. Robots are sending work log reports to UiPath Orchestrator and their performance can be monitored from there (UiPath, 2018e).

UiPath Studio is a process modeling tool for RPA developers. Automation code can be designed by using custom workflows or readymade templates. Business process rules can be document in a visual way to the workflow model in UiPath studio. Automation code can be formed from different coding languages such as Java, Python, VB.Net, and from many premade components. Developed components and models can be shared or reused between developers (UiPath, 2018c). Automated process can be coded in three different layers; sequences, flowcharts, and state machines (UiPath, 2018a, p. 5).

UiPath robots are software that run the processes which are modeled in UiPath Studio. There are two types of robots. Attended robot operates on the desktop that is used by a human and unattended robot which operates in a physical or virtual environment without human supervision. Unattended robots can operate behind locked screens while attended robots require open screen and human signed on in the desktop (Uipath, 2018d).

5.5 Capabilities of Robotic Process Automation Software

UiPath Orchestrator has inbuilt capabilities to enable flexibility in automated business processes. This is achieved by using work queues, scheduling jobs, stopping jobs and process related assets. When required, a job can be stopped or terminated while process is running. A “soft stop” is possible for a process, but it requires a “should stop” activity in the workflow model. Terminating the process kills the process instance immediately. Processes can be distributed into different environments in UiPath Orchestrator, such as testing and production environments. Queues in the UiPath Orchestrator can be used for robots to organize the process instance cases and executing them individually in the automated business process (UiPath, 2018a. p. 21, 34 - 38). Assets which are stored in the UiPath Orchestrator such as credentials and variables, enable flexible process instance configuration also when the process is being executed by different robots (UiPath, 2018e).

The robots execute the processes for example by using mouse clicks, keystrokes or by invoking code to interact with the application. An example is shown in figure 14 where simple typing and clicking activities are modeled. These activities are coded activity blocks in UiPath. There are three different ways for UiPath to interact with an application. These are SendWindowMessage, SimulateClick or SimulateType, and hardware events. With simulate types of interaction the robot interacts with the application by triggering the event handler of a given user interface element. For SendWidnowMessage, event details are posted in the message loop of the application and these messages are then dispatched to the targeted user interface element internally. Hardware events work the same way as a human would interact with the application and the user interface screen must be visible (UiPath, 2018a. p. 11 – 12).

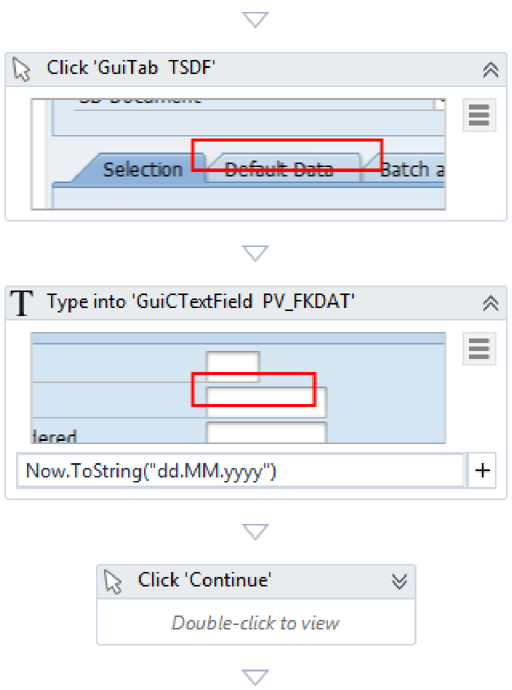


Figure 14 Example of user interface interaction activities and sequence in UiPath Studio

The full list of activity types is available in appendix 2. There are 18 main activity packages in UiPath Studio to model the automated business process. These activities have various purposes for the automation, and they range from interacting with applications to manipulating data and building the control flow in the automated business process model. It is possible to create own custom activities, if needed activity type is not found from offered activities (UiPath, 2018f).

In figure 15 a control flow is displayed inside a sequence. This can be used to make the robot to loop through given data and model a data manipulation logic into a workflow model of automated business process. The conditions in if/else decision points set correct sequences for robot to perform correct activities. There are four different types of looping activities, “For each” (UiPath, 2018g), “For each row” (UiPath, 2018f), “Do while”, and “While” loops. With “For each” looping activity an array, data table, or list is looped for each row in it and given activities are then performed within the loop. “While” looping activity is performed if condition is true. When condition is set to false the loop is then ended (UiPath, 2018g). When complex business logic needs to be modeled, it should be modeled as a flowchart or a state machine. Interacting with user interface and data manipulation should stay in the sequences (UiPath, 2018a. p. 18 – 20).

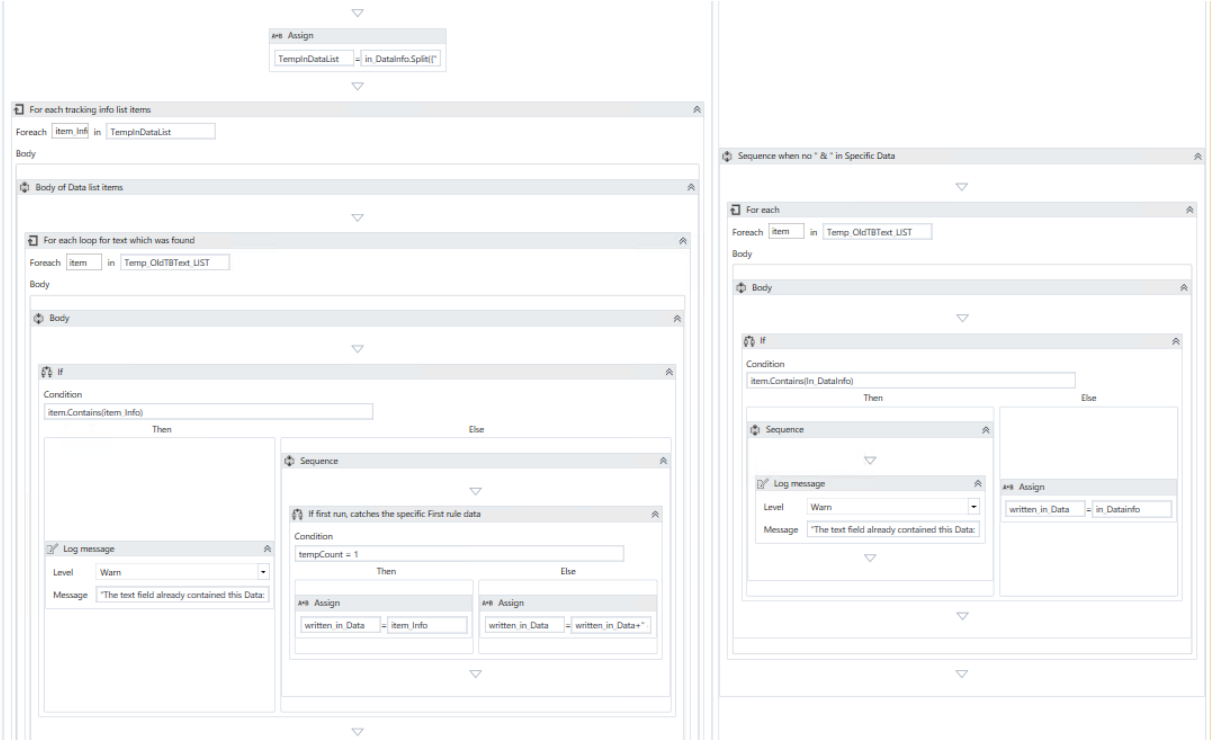


Figure 15 Example of Data manipulation in sequence with UiPath Studio

Flowcharts as seen in figure 16 are useful when automated business process has a complex control flow and there is a decision logic required for the robot to choose the correct branch to perform activities for each process instance case (UiPath, 2018g). In the flowchart design, only the business logic of process control flow should be implemented (UiPath, 2018a. p. 19 – 20).

Figure 16 Example of Flowchart in UiPath Studio

A state machine as depicted in the figure 17 allows process modelers in UiPath studio to model complex transitions between business process states. These states are triggered by activities and data (Uipath, 2018g). State machine is used to catch the high-level process control flow such as exception handling, retrying and recovery of automated process. Business processes which have multiple cases to transact during the business process, requires a state machine type of approach in workflow model to model the automated business process (UiPath, 2018a. p. 18 – 19).

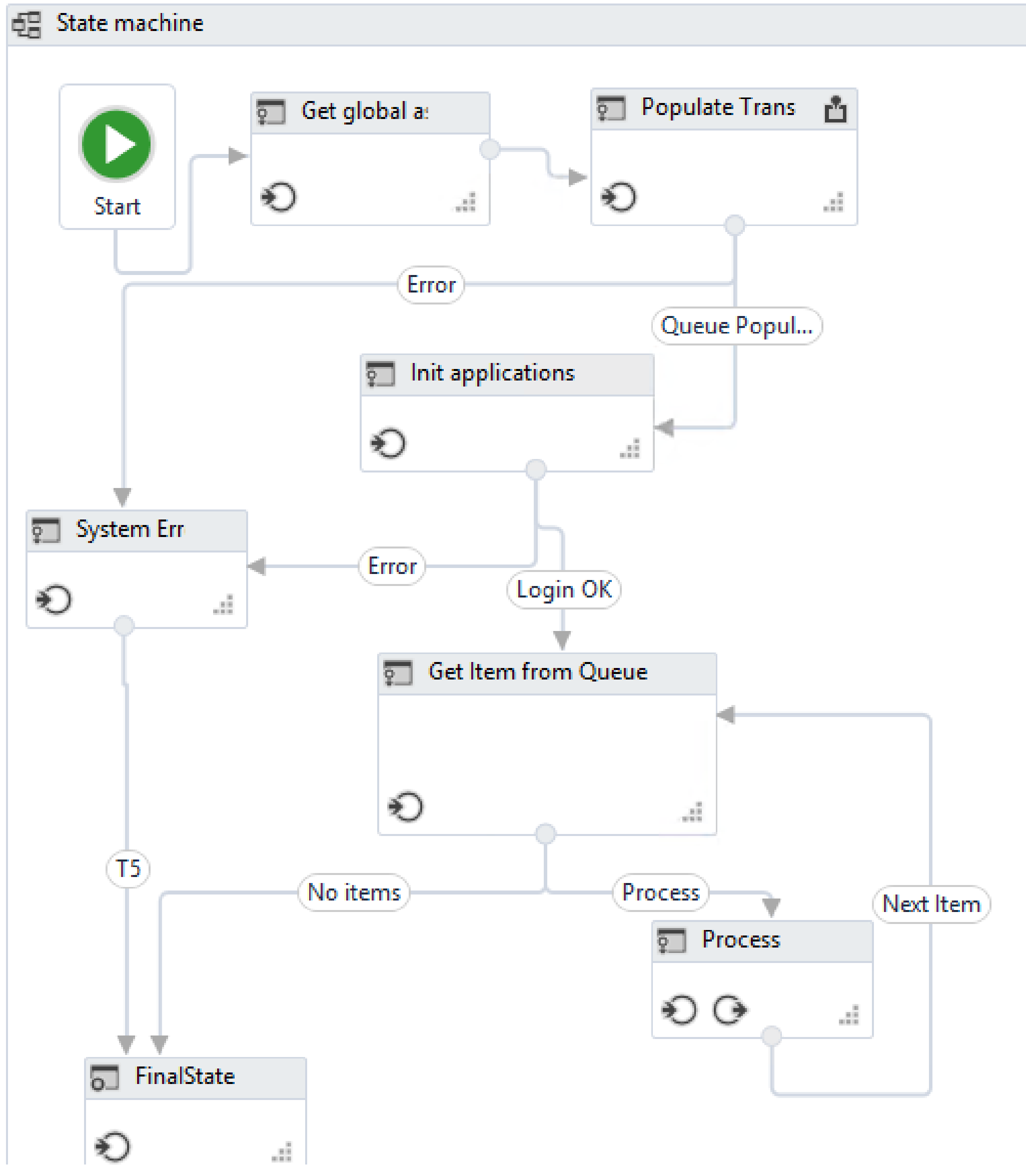


Figure 17 example of state machine in UiPath Studio

An example of computer vision-based selector is shown in the figure 18. These selectors are graphical user interface XML fragments. Robots are capable to navigate to correct application element by using the selector as a guide. The first node of the selector represents the root and parent window of the application. Last node represents the graphical user interface element which is the target of the activity. Other nodes between the first and last node are ancestors of the last element. Selectors enable a reliable way to interact with almost any target application (UiPath, 2018g). If a selector changes in the application in the current versions of UiPath RPA software it must be manually corrected. Other option is to try to create a dynamic selector (UiPath, 2018a. p.12). In case the selector is not reliable, then using image recognition is an alternative way to recognize elements in the application. With image recognition robot searches for image that matches the reference image with given accuracy and robot will execute activities for the recognized element. Selector based on image recognized activity is presented in figure 19 (UiPath, 2018g).

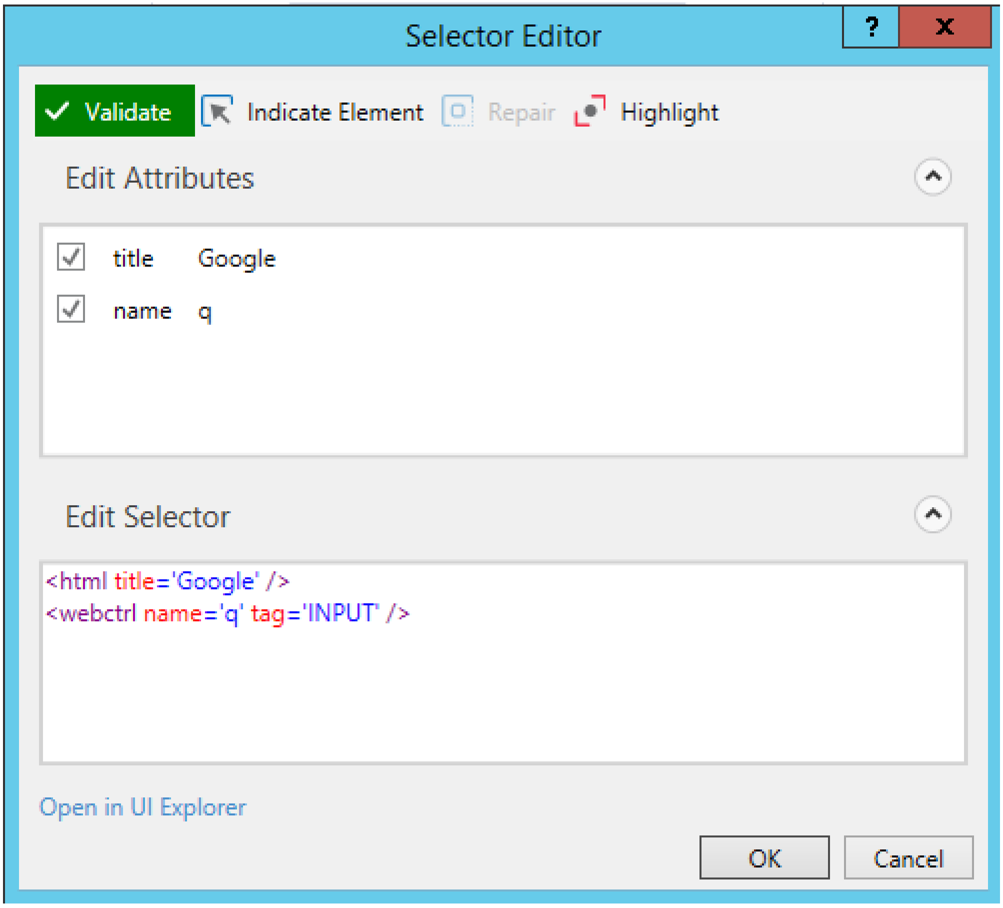


Figure 18 Example of Graphical User Interface Selector

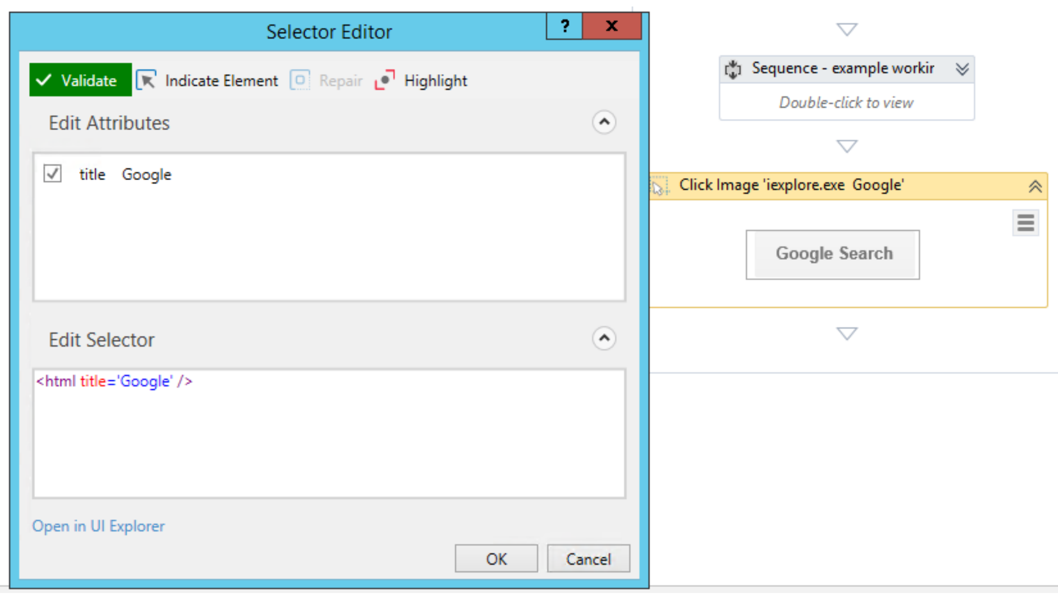


Figure 19 Example of Image recognized selection of graphical user interface element

6. EMPIRICAL RESEARCH

In this chapter the methodologies how, the research data was gathered are explained and analysis of gathered data and results of analysis are formed to answer the research questions. The research is based on one project which was conducted in the case company during summer 2018 for a branch in United States of America (USA). The project started at the end of May 2018 and ended at end of July 2018. Further development for the automated process was required in August 2018, and the original planned time for project did not held. Target was to finish the project by mid July 2018.

6.1 The Methodology of Empirical Research

There are two different methods to gather data for a research, qualitative and quantitative. It is possible to mix these together to make a hybrid version of research methodology. Qualitative research is designed to use non-standardized data collection, such as semi-structured interviewing. Characteristic for a quantitative research is structured data collections, such as numerical statistics and structured questionnaires. It is possible to use mixed method, where both quantitative and qualitative methodologies are used. (Saunders et al. 2016, p. 165 – 172). In this research both methodologies were used, data was gathered with quantitative and qualitative methods.

The used methodologies to gather information about the business process and its complexity during the project were, Workshop-Based, Evidence-Based, and Interview-Based methodologies. The Interview-Based methodology was conducted as mostly freely spoken e-mail conversations. E-mail conversations were conducted during the project between chosen project participants. Only questions regarding the business exceptions were formed before they were discussed. Questions regarding the business process exceptions were:

Questions of e-mail conversations with business organization domain expert:

1. What are the business exceptions of the process?
2. What are the correct steps for each business exception when it occurs?

Figure 20 shows the states of RPA project as they are defined in the case company. These project states follow the BPM framework process modeling and process instance levels. Run, maintain and support state corresponds to process instance level in BPM framework. Pre-analysis, process analysis, process development, robot development, and robot deployment and project end represent process modeling level of BPM framework. Multi process management level is not part of RPA framework in the case company. It is kept in the business function organizations where the input for RPA projects is given. It is possible for the RPA team in the case company to help the business organization in identifying state of the multi process management level to find good business processes to be automated with RPA.

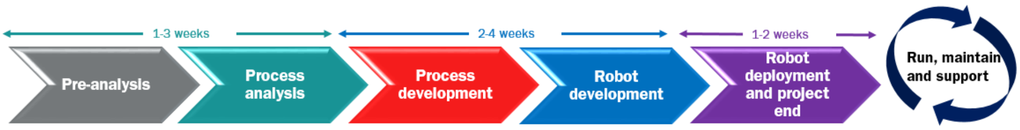


Figure 20 RPA project states in the case company

RPA team in the case company participated in the process identification state to identify suitable processes for RPA amongst the processes carried out in the business organization. The prioritization of processes was then done within the business organization. When the RPA project started, three key persons were chosen to participate in the project. One from RPA team, one from business organization and one from external partner's organization. The person who participated from the RPA team of the case company, is also the person who has written this thesis.

The outputs of the RPA project are three documents and executable workflow model of automated business process. Process design document (PDD) is the document that has all information related to as-is and to-be business process. The granularity level of the documentation is at the micro steps of the business process. In the documentation, also business rules are document and linked to the business process model. The solution design document (SDD) holds the technical aspects of the automated business process and it tells how the workflow model has been built. Complexity and value of business process are assessed in a separate document, which uses information from the PDD. The PDD and complexity and value documents are the main data sources of this master thesis empirical data.

In this project workshops were held two times, one at the beginning of the project and another during the robot development when automated business process model was ready for testing. Then evidence-based methodology was used when as-is process document was done by the business organization after the workshop and sent to case company's RPA team. After analyzing the as-is process document, questions regarding the process exceptions were sent to the business organization. Business organization added the exceptions in the documentation. Then process model was modeled and documented in the PDD. When a problem was detected, the documentation was done by business organization and business logic was described to the RPA team member who then documented this in the PDD.

Four issues did rise during the project that slowed the project. These issues were raised during Robot development, Robot deployment, and Run, maintain and support state of the project. New business rules and logic had to be implemented in the process and this caused delays and problems when running the robot. All detected issues were documented in the process documentation.

Identified issues which caused delays were:

1. A conditional task to check validity of output was missing.
2. A process task to manipulate text data was missing.
3. Exception handling in input data type was missing.
4. One exception and its handling was missing in the document post task.

Process knowledge came from the business organization. Case company RPA team member helped with the documentation, process analysis, and process development. The external consultant was used during robot development to design the automated business process model. Case company RPA team member participated with external consultant in the robot deployment and run, maintain and support state in code configurations to fix the detected issues and documenting them.

6.2 Empirical Research Data gathering

The project started in May 2018 with a workshop between the RPA team and business organization in USA. There were 12 people in total who participated in this workshop. Two from RPA team and 10 from USA business organization. During this workshop, the RPA capabilities were discussed and one business process from the USA business organization was presented and discussed in detail. During the workshop, a decision was made for business organization to document their current as-is process and send it to RPA team for further analysis. There was not any existing documentation available for this process.

Figure 21 presents the as-is business process. The business process was modeled without any specific modeling language. In the as-is process model only the sunny-day flow of the process was modeled. The as-is business process documentation, PDD, was validated by the USA business organization.

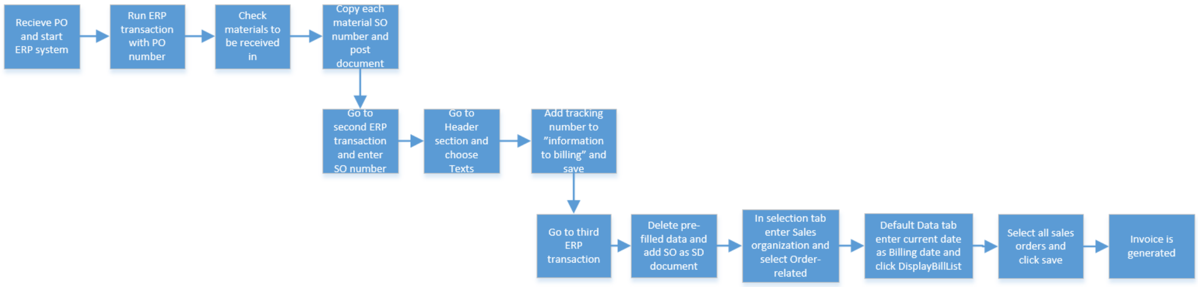


Figure 21 As-is business process model of invoicing process

The as-is business process contains two roles, a buyer and an invoice handler. The process starts when a buyer receives purchase order (PO). Then the buyer logs into the ERP system and posts material document there. Then the buyer sends an e-mail with PO information, sales order (SO) number, posted materials, and tracking information to an accountant. This information comes from e-mails and ERP system reports. Invoice handler will then start adding information to the invoicing document. Then invoice handler will proceed to print the invoice in paper or send an automated e-mail from ERP system with invoice as an attachment to the customer.

In the table 9 there are the complexity measurements and their descriptions. These complexity measurements were defined by the RPA team when RPA team was established in the case company and first RPA project started. There are currently eight process complexity measures in use in the case company. Values can range from 0 to 3 and the total amount is multiplied with 1,25 to get a weighted value from complexity that can be compared with the business value score. It should be noted that without the weight, complexity value of the process in question is 11 and the weight is used only to make the complexity value comparable with business value. When external consultant assessed the business process complexity, the result was a complexity level of a low-high which is not yet medium complexity level in their own process complexity scale. In case company’s RPA team, the scale maximum is 30, where 0 to 10 is low complexity, 10 to 20 is medium complexity and 20 to 30 is high complexity. This process complexity was valued in case company’s RPA team scale as 14 which tells it is at medium complexity level.

Table 9 Complexity measurement of case company and project process

Complexity measurement	Value (0 to 3)	Description of complexity measurement
Business rules	2	Score 0 when less than 5 business rules and score 3 when over 20 business rules. Business rule should always resolve to true or false statement
End-to-end variations	1	Score 0 when no variations and score 3 when over 5 variations. Variations are parallel sequences and process branches.
Validations	0	Score when less than 5 validations and score when over 20 validations. Validations are validating that earlier step was correct or information is correct.
Exception handling	2	Score 0 when 1 exception and score 3 when over 5 exceptions. A business exception is deviation from normal process flow.
Business Applications	1	Score 0 when 1 application and score 3 when 5 or more applications. All applications which are used during the business process, such as ERP, office and web applications
Modules	1	Score 0 when less than 5 modules and score 3 when over 15 modules. A module is transaction module in ERP system or other application.
Screens	2	Score 0 when less than 10 screens and 3 when over 40 screens, a screen is application user interface screens that is required to work with during process.
Steps	2	Score 0 when less than 10 steps and score 3 when over 40 steps. These steps are the detailed micro steps of process.
Total score	11	Sum of complexity metrics
Total weighted score	14	Total score multiplied with 1,25

During robot deployment state robot work was monitored through UiPath Orchestrator and also by logging in to the servers to observe the robot working. The business organization gave feedback that robot was not working as it was expected from the business logic point of view. The RPA team monitored if any technical errors occurred while process instance cases were running.

Issue two was detected from the business organization feedback. The robot was putting same tracking information multiple times in to a text field, even when the original text contained the correct tracking information that the robot should add to the field. This was a quality issue, since customers would see same information even 20 times in the invoice.

Issue three was detected by RPA team. This problem caused robot failure and it stopped working completely. It was quickly found out, from robot logs that people could feed the input data in multiple data types and this had not been taken into account.

Last problem was detected after robot had been running for couple weeks in the run, maintain and support state of project. The business organization had noticed, that this exception occurred frequently in the robot exception handling documentation. They decided that it should be implemented in the automated business process model so robot could handle the exception.

6.3 Empirical Data Analysis

In this chapter the four issues which occurred during the process are analyzed using the theories and empirical data to describe the root of the issues. After analysis of issues, the capability of RPA software to solve this is assessed.

6.3.1 Analysis of First Issue

The first problem shows, that business logic and rules were not implemented well in the process documentation and as-is process model. Used process modeling language was simple boxes and sequence arrows to depict the successful process instance. Process model of the as-is business process doesn’t use XOR gateway split to check if the output data is valid or not. As shown in figure 22 this rule is implemented in the business process model of Generate invoice sub-process. A person must check if the invoice was generated by print, or by e-mail, or not at all due to an unknown error in the business process. This check is done by event-based XOR gateway split in the workflow model of automated business process. If pop-up appears, then customer is using a printed invoice. However, if print pop-up does not open in 5 seconds a manual check must be performed. If system log is empty, then the invoice was not generated due to an unknown error. If system log contains information, then the customer type must be checked. If customer type defines that the customer is using e-mail invoice, then the e-mail log is checked in the system. If customer type defines that the customer is using a printed invoice, then printing is initiated manually. After this the work list is updated with information about the printed invoice. In the process model, the pools and lanes aren’t used due to decomposing the process model. The process context is human performing activities with ERP system, which has multiple connections to other resources, such as data store and printer machines. In this process-model ERP system uses data stores and printers, and human uses the ERP system.

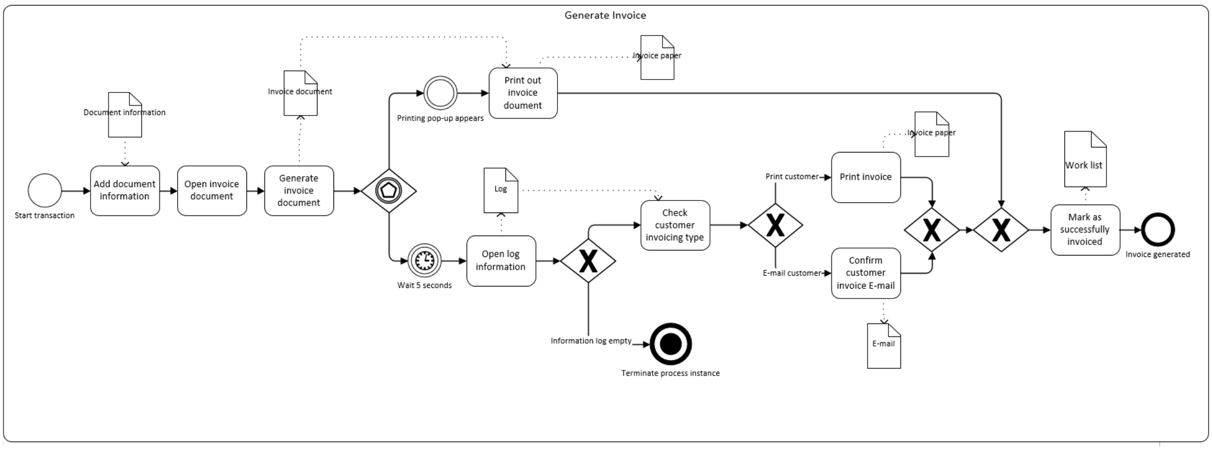


Figure 22 Generate invoice sub-process

The process model has three XOR gateway splits, two intermediate events, one cancel case where sub-process instance is terminated, three input data-objects, five output data-objects, one start event, one end-event, nine task activities, 20 normal flows, and nine association flows. In total without flow-elements there are 27 elements in the process model. With sequence flow-elements there are 34 elements in the process model. According to 9 PMG process model should be decomposed. There are two end events used, so process model does not follow one start and one end event guide line. seven out of nine process modeling guidelines are followed as they should.

Validation of invoice generation was missing from the original process model, which led to wrongly designed automated business process model for the robot. In the original automated business process model, the robot only tried to print out the invoice from pop-up and if the pop-up was not found an exception was thrown. The first issue occurred because process model did not have business rules concerning checking the output of the process in the control-flow and data-flow. Thus, the activities to do the check were missing from the original as-is and to-be process model.

The business process complexity was not well understood before automated business process was designed. The original process model was not informative enough to depict how this check of output validity should be performed. Lack of as-is process documentation before the RPA project started might be also a reason behind this. People usually tend to tell the sunny-day flow of business process and possible exceptions are not explained well-enough. If as-is process model would have been modeled in the BPMN language, the business process complexity could have been understood and measured properly in full end-to-end process level and in its sub-process process levels. The complexity measurements of this sub-process are listed in table 10 with weighted values.

As seen from the complexity measurements, the sub-process has also data input and it generates output data to log and to the invoice. In the process model, resources were not modeled, due to decomposing reasons. In total 3 resources were used, human, ERP system, and printer machine.

What increased the weighted complexity of this sub-process, were event-based XOR-gateway, XOR-gateway, and cancel case activity, which is in BPMN terminate end-event. The gateways and terminate event increases the control-flow complexity of a sub-process. Value of CLA was 1. This means that from each task activity, only one sequence flow goes out and to each task activity only one sequence flow goes in. The whole sub-process model control-flow complexity weighted value was 20. CLA multiplying value was 1 for control-flow complexity.

Table 10 Generate invoice derived complexity as weighted value

Derived complexity measurement	Weighted value
Total number of start events (TNSE)	1
Total number of intermediate events (TNIE)	2
Total number of end events (TNEE)	1 + 3 = 4
Total number of activities (TNA)	9
Total number of gateways (TNG)	6
Total number of Data objects (TNDO)	7
Sequence flows between activities (NSFA)	9
Sequence flows from gateways (NSFG)	6
Sequence flows from events (NSFE)	3
Total weighted value	47
Connectivity level between activities (CLA)	1
Proportion of data objects as incoming product (PDOPIn)	3/7 = 0,42
Proportion of data objects as outgoing product (PDOPOut)	4/7 = 0,57
Proportion of data objects as outgoing product of activities (PDOTOut)	5/7 = 0,71

Activity complexity got value 9 from TNA. The business process does not require many activities, which means that its activity complexity is low. When compared to control-flow complexity it is remarkably smaller. However, performing the activities could be harder for robot than for human since activity complexity does not tell how selectors are formed in the application.

When comparing PDOPIn and PDOPOut, the sub-process produces more outgoing data than is used in the sub-process. When PDOTOut is compared to PDOPIn, it is noticed that activities in the sub-process generates more data out than data is used in the sub-process. When PDOTOut is compared to PDOPOut, the PDOTOut gets higher value. This implies that when task activities generate the data, one of these data objects is consumed during the business process. There are two XOR-gateways which are data-based gateways, these requires input data from earlier activities to make the decision. The data-flow complexity consists of activities that uses the data or generates it, splits that require data and events which generate data or consume it. The total data-flow complexity is 16, when also sequence flows from gateway elements are calculated.

It was possible to add this control-flow in the automated business process. Implemented logic for the robot was that, it waits for the printing pop-up to appear for 5 seconds, if pop-up does not appear during this waiting time. Then the robot will check if the invoice document was generated and if not, then the robot sends a notification as business exception to a defined email address. If invoice was generated, then the robot checks if customer is print or e-mail type of customer. If customer is print type, robot will initiate the print again. When customer is e-mail type of customer, then robot will verify this from the log information and continues to next work case. Robot can perform same activities in the business application as a human can and is capable to understand the context of the process information and of the data it receives from the business application.

6.3.2 Analysis of Second Issue

The second issue occurred, when robot was meant to add tracking information in the invoice document. In this issue the robot was able to add the tracking information when it was already found in the document or add it multiple times when it was found multiple times in the input data. This caused a quality issue, since it should have been only added once in the invoice text, as it was done in the as-is process. In the as-is process model, that was documented in the PDD, the task activity mentioned only “add Tracking number to “information to billing” and save”. There were no previous tasks or control flow activities to read the text from the “information to billing” field and filter the input data. Reason why same tracking information could had been told multiple times, was that different materials in purchase order had same tracking information.

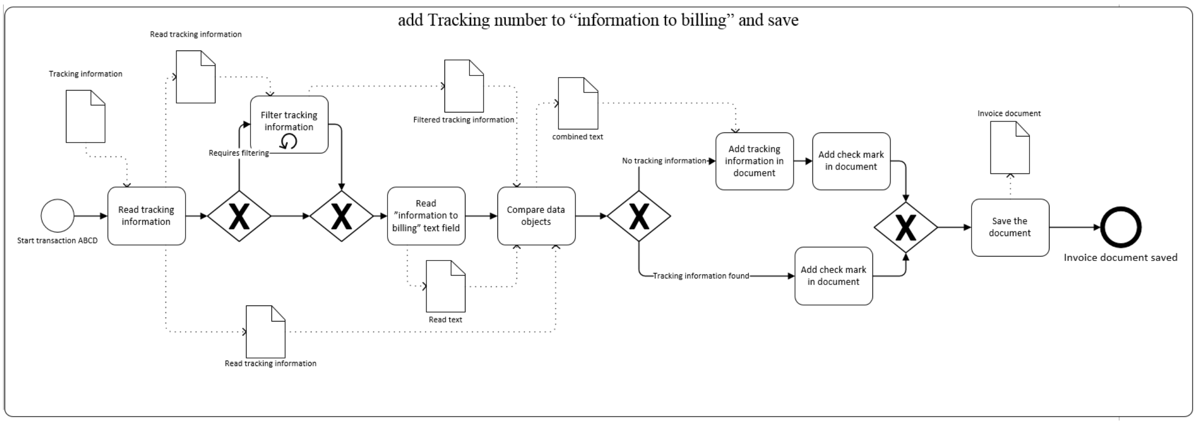


Figure 23 add Tracking number to ”information to billing” and save sub-process in BPMN

In figure 23 the “add Tracking number to “information to billing” and save” sub-process is modeled with BPMN process modeling language. When checking the process model by using 9 PMG:s, it is valid in all guide lines. In total there are 27 elements in the process model, which is below 31. Activity labeling is used in verb – object format. As few elements as possible are used, and loop is modeled by using looping task activity. All data objects are associated with task activities, and only one start and end event are used. Gateways are matched with respective joins, and OR gateway splits are not used.

Data manipulation is performed first to the tracking information data. If it contains the same tracking information more than once, then the text from “information to billing” field is read and checked if it contains the same tracking information. If same tracking information is not found, then it is appended to the text, marked as checked and finally the invoice document is saved. The data flow complexity was not well modeled in the as-is process model. There are in total six input data objects to task activities and output data objects from task activities in the process model.

The lack of understanding the data-flow complexity in the as-is process caused the quality issue of generating the same tracking information multiple times as the output from the automated business process cases. When human process participant was executing the as-is process, filtering the tracking information and checking if “information to billing” field text had already the tracking information was not mentioned because it was obvious for human. When modeling the data-flow in the business process model, it is possible to understand in which activities which data objects should be read or used, and how the data objects should be manipulated before using them. This has an increasing process complexity effect, since it adds activity complexity, control flow complexity, and data-flow complexity. If other resources are required to handle the data, it also adds resource complexity in the business process.

Table 11 Complexity values for add Tracking number to “information to billing” and save sub-process

Derived complexity measurement	Weighted value
Total number of start events (TNSE)	1
Total number of end events (TNEE)	1
Total number of activities (TNA)	8
Total number of gateways (TNG)	6
Total number of Data objects (TNDO)	7
Sequence flows between activities (NSFA)	8
Sequence flows from gateways (NSFG)	4
Sequence flows from events (NSFE)	1
Total weighted value	28
Connectivity level between activities (CLA)	1
Proportion of data objects as incoming product (PDOPIn)	6/7 = 0,86
Proportion of data objects as outgoing product (PDOPOut)	1/7 = 0,14
Proportion of data objects as outgoing product of activities (PDOTOut)	6/8 = 0,75

Table 11 contains the derived complexity measurements of this sub-process. Reason why TNG got value 6, even when process model has only two XOR-splits and -joins in it is the loop task activity. To form a loop, it requires an XOR-split and -join. The process model had in total seven data objects and out of these seven data objects one was input for the sub-process and one was output of the sub-process. Other data-objects were generated during the process model and consumed in it. This process model does not contain resources in pools and lanes element, the resource complexity can't be evaluated.

The control-flow complexity is 13 and multiplying value for control-flow is 1 from CLA value. This sub-process does not have any intermediate events and cancel case events. This lowers the control flow complexity of the sub-process.

Activity complexity of the sub-process is 8. Activity complexity got a low value, because in this process the amount of activities stays low, and the activity complexity type does not have cognitive weight adding activities. This does not mean that performing the activity is easy for robot because. In this sub-process many activities require data as input and manipulating or understanding the data. This means that before performing an activity to application, robot needs to read data and perform activities to data.

PDOPIn is quite high, which means that this sub-process handles most of data-objects as input to its activities, and these activities generate quite a lot of output data which is visible from PDOTOut. In complete sub-process, PDOPOut stays quite small, which means that this sub-process does not generate much output data from the sub-process. Most of the data which is generated in the process is also consumed in it. This indicates that, this process requires quite much data handling. The total data-flow complexity of this sub-process is 16. When comparing this to the first issue sub-process, this sub-process has bigger data-flow complexity than control-flow complexity which also means that this sub-process is mainly data manipulation.

Implementing the data manipulation for the workflow model of automated business process was possible since robot can use different types of coding languages and it has pre-made activity blocks for data manipulation. This issue was solved by using capabilities that the RPA software offered as premade activities and with coding implemented in the automated business process model.

6.3.3 Analysis of Third Issue

In the as-is process the sub-process “Receive PO and start ERP system” the PO data is received by e-mail and then other required information to form the worklist is exported from ERP system. Human is capable to understand quite easily written data, for example what data refers to purchase order, item line number, and tracking number information. This was not felt as necessary to put in the as-is business process model or tell the used data types. When reading Excel file, the purchase order could be stored as text when it should be stored as number data type. Because robot requires the data type is defined explicitly. When a purchase order number was given to robot as text, this caused the failure in the automated business process because the robot was expecting to read only number data type. In figure 24 the “Receive PO and start ERP system” sub-process is modeled in BPMN language.

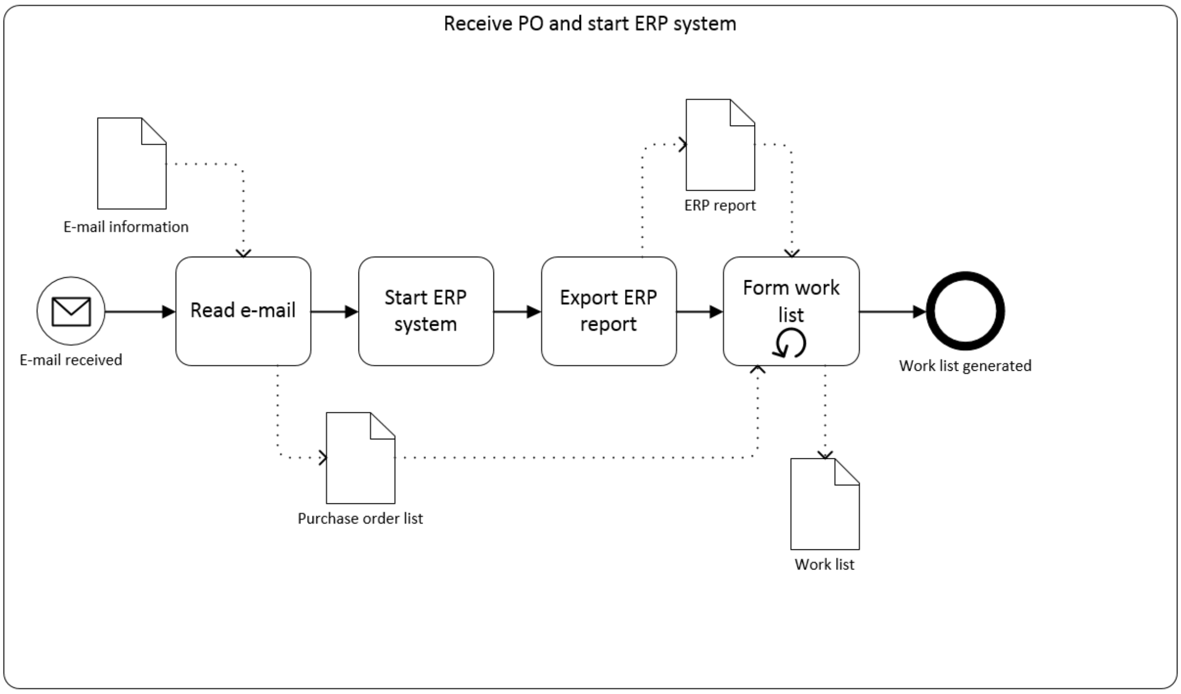


Figure 24 Receive PO and start ERP system

The sub-process process model in figure 24 has less than 31 elements in it, all data objects are associated to task activities, loop activity is used to form the loop, there is only one start and end event, activities are labeled with verb object labeling, usage of elements is minimized and there are no OR-gateways are used. The process model is therefore valid and follows all 9 PMG guidelines.

In the process model the changing datatypes in the work list are not modeled, since human process participant does not require that. Business organization was not expecting this kind of behavior from robot and was unable to identify the need for the data type change. Using complexity measurements to understand the complexity of as-is sub-process won't give the desired outcome and nor the reason for robot failure.

This issue is related to RPA software capabilities. Robot needs to change the read data type to a correct one before using the data in the activities. This is not a business exception but, a technical exception. To solve this issue, understanding the capabilities of the used RPA software is a key requirement and general know-how to use the software in a correct way.

Since RPA team was still quite inexperienced with RPA software in summer 2018, an external consultant was used to build the workflow model of the automated business process. This task was not requested to be implement in the automated business process model due to RPA team's lack of understanding the capabilities of the RPA. After the technical exception was identified, the required activity was implemented in the workflow model of the automated business process and robot was able to solve the technical exception.

6.3.4 Analysis of Fourth Issue

The fourth issue was a business exception, that was not recognized to occur during the sub-process "copy each material SO number and post document". In this sub-process human checks each material line number, also known as sales order (SO) line number, then copies each material line number specific sales order number and posts the material document. In some cases, there was an exceptional pop-up shown on the ERP system regarding the serial numbers. This required a new task to maintain serial numbers during posting the material document.

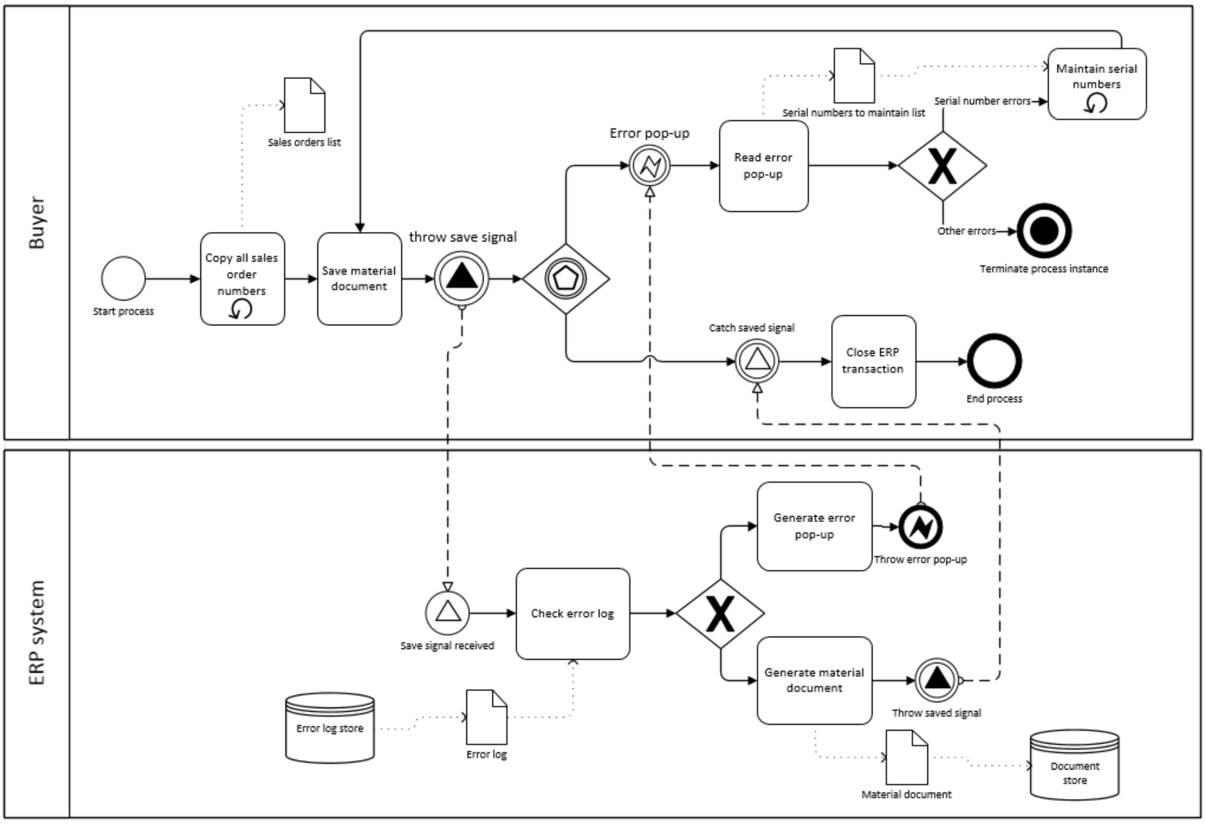


Figure 25 Business sub-process ”copy each material SO and post document” in BPMN

In figure 25 the business exception handling for maintaining serial numbers is modeled in BPMN language in the “copy each material SO number and post document” sub-process. The process model contains in total over 31 elements, when sequence flows and message flows are counted. All data objects are associated to task activities and one loop has been modeled using sequence flows. Two start events are used, and two end events are used. All activities have only one outgoing sequence flow, and no OR-gateways are used. As few elements as possible have been used in the process model. In total seven out of nine 9 PMG are followed, and thus the process model can be validated as understandable process model.

In table 12 is the total weighted value for issue four business sub-process calculated. The value was higher than with other issue sub-processes. This can be explained with multiple loops, work done by two resources, and exceptions. CLA value was over 1, which indicates that there are more activities in the process model than there are connections between them. The explanation for this is found by looking at terminate process, error pop-up and signal intermediate events. Half of all data objects are consumed and used in this sub-process, and half of data objects are generated as output of the business process. During this sub-process three out of four data-objects are generated as output from the activities.

Control-flow complexity is 20,12 and data-flow complexity is 23. Even though, the number of data-objects remained low in the business sub-process, its data-flow complexity was increased due to multiple start events, end events, intermediate events, and message flows between process participants. The activity complexity is 8 and resource complexity is 2. This score on resource complexity means that two process participants were recognized to participate in the business process.

When people described the business process in the workshop, they told the sunny-day process flow. When exceptions were questioned from the business organization domain experts, there were no previous documented data available of business process and its exceptions. Only those exceptions were identified that people remembered. This exception was recognized during the time when robot was already running the automated business process and producing process relevant exception data out of process instance cases. Robot had run already more than 400 process instance cases before this business exception was discovered.

This exception could have been identified at an earlier stage, if the business process would have been documented and kept up-to date before the RPA project or if process mining tools would have been available to use. There is also a possibility, that this exception could have been identified if member from RPA team of case company would have executed the business process with large amount of process instance cases. Since the initialization for exception came from ERP system and its error log information, a process mining tool would be the best way to detect this type of business exception. In addition, a process document that is kept up-to date could provide the same information.

The exception handling steps and business logic was implemented in the automated business process model. It required dynamic selector usage, since robot had to be able to read all possible cases for maintaining the serial numbers. A business logic was implemented to the workflow model of the automated business process allowing the robot to understand when error pop-up contains error information that robot is not allowed to handle. In these cases, robot will send information to human process participant to take control of the business process instance case.

Table 12 Weighted derived complexity values for copy each SO number and save document sub-process

Derived complexity measurement	Weighted value
Total number of start events (TNSE)	2
Total number of intermediate events (TNIE)	4
Total number of end events (TNEE)	2 + 3 = 5
Total number of activities (TNA)	8
Total number of gateways (TNG)	8
Total number of Data objects (TNDO)	3
Sequence flows between activities (NSFA)	7
Sequence flows from gateways (NSFG)	6
Sequence flows from events (NSFE)	3
Looping sequence flows (NSFL)	1
Message flow (NMF)	3
Pool (NP)	2
Total weighted value	52
Connectivity level between activities (CLA)	8/7 = 1,14
Proportion of data objects as incoming product (PDOPIn)	2/4 = 0,50
Proportion of data objects as outgoing product (PDOPOut)	2/4 = 0,50
Proportion of data objects as outgoing product of activities (PDOTOut)	3/4 = 0,75

7. DISCUSSION

In this chapter the theories and empirical data are linked to each other and it is assessed how well the theories provide answers to the issues identified in the empirical research. Also, the validity of this research from both academic and business organizational perspective is discussed. At the end limitations and future research topics are presented.

7.1 How Theories Support Solving the Issues

The first research question was "How information regarding the business process can be gathered". In the case company, it has been noticed that current as-is business processes have been poorly documented. There is no up-to-date process documentation available and during process discovery, the main knowledge of business process is gained directly from domain experts. This leads into the kind of issues which were realized during this RPA project. For example, exception handling was not well understood, and business rules of process were hazy. As it was noted in the Interview-Based and Workshop-Based process discovery methodologies, these will in most cases result to a "sunny-day" flow of the business processes and exceptions are vaguely explained or not told at all. The third issue which was discovered in the project was also unknown at first for business. This exception was noticed after robot had run for quite some time in the production and produced a log file containing information regarding the exception. Since business supporting application, the ERP system, was producing this exception from its log information that contained information regarding the posted materials, process mining could have been used to notice the exception during process discovery state of the RPA project.

In some cases, process mining tools might not provide the correct answer. These situations might occur when log information is not correct, or it is wrongly stored. During interviews and workshops, some essential information might not be told because of the situation. When context is automating the business process, this might be a reason for people not to tell everything about the business process which they are currently working with.

The Evident-Based discovery methodologies requires process documentation, which is created and kept up-to-date, however there is a possibility that business process has no documentation at all. In these situations, process mining tools might provide answers for gaining detailed business process information. Deeper discussion around the business process should be conducted with domain experts in interviews and workshops. Taking video or observing while the process participant works with the business process could be used if there are no process documentation available and process mining tools are not available to be used.

It was confirmed, that as-is process model and business process documentation were validated during this RPA project in summer 2018. Despite this some issues occurred regarding the business logic. The second research question was: “What needs to be known of as-is business process before automating it?”. Complexity of as-is business process was not well understood during the RPA project, and this caused the first, the second and the fourth issue. The process was not modeled with standard business process modeling notation language such as BPMN is and used complexity measures that were used did not measure the different perspectives of process complexity well enough.

Recommendation would be to utilize a standardized business process modeling language such as BPMN in the next RPA projects when as-is and to-be business processes are modeled. Both business process models should be validated by using the 9 PMG for ensuring process model understandability.

Then derived complexity measurements with cognitive weighted values should be used to measure the business process complexity. The complexity measurements will tell the most complex parts of the current as-is business process, whether it will require data manipulation, and what kind of process automation tasks might be required in the automated business process model. As derived and cognitive weighted complexity measurements used in this thesis gave an understanding what is required to model in the workflow model of automated business process. When control-flow complexity gained higher value, it meant that decision points are required to model in the workflow model. If data-flow complexity was measured with high value, it told that data manipulation must be built in the workflow model so that robot can produce correct output from automated business process. However, derived complexity measurements with cognitive weighted values could not be used for technical exception which was related to RPA software capabilities and not to the business process.

Since there are no usable thresholds for complexity measurements, this might cause some difficulty in estimating the process complexity level and effort to automate it. Utilizing the process modeling language won't solve this issue due to lack of current as-is process documentation and because there is a risk of untold business process knowledge residing somewhere in the organization. Using process modeling languages would also require training and practice to fully understand them and benefit from them.

The third issue was caused due to lack of understanding the capabilities of the RPA software in use, UiPath. The third research question is "What are the capabilities of RPA?". Better understanding of RPA software's capabilities will also support to understand what should be documented and what aspects should be modeled in the to-be business process model.

As RPA software is new for the case company and its RPA team, the knowledge of RPA software and its capabilities has not yet matured in the team. RPA team members have participated in UiPath RPA academy courses and they have also built some workflow models of automated business processes with UiPath software. Solution for the issue two was data manipulating, which indicates that training in coding would be required to utilize the RPA software's capabilities even further.

As three out of four issues revolved around the understanding the complexity of the business process, and only one issue was because of lack of knowledge of the RPA software. It gives a good reason to improve the capabilities in understanding the complexity of the as-is and to-be business processes models since they are used to build workflow model of the automated business process. Know-how in using the RPA software should be increased with more training and coding training. This should help the RPA team to fully utilize the capabilities of the RPA software in automation of business processes. It has been proven that RPA contains multiple capabilities from different types of BPMS and thus, it could be well established that RPA is a hybrid type of BPMS and a process aware information system.

During autumn 2018, another RPA project was conducted where the business process was assessed as a medium complexity process both according to external consultant and the case company's RPA team. This project was successfully completed in timeframe it was given and business organization gave good feedback of automated business process performance. In this project BPMN was utilized in process modeling and as-is process documentation existed before the RPA project started but process mining was not used. Workshops and interviews were held during process discovery to gather information and gain knowledge of the business process. However, there were still some issues, that should have been detected during the process discovery. These issues were regarding the business exceptions in business process. Derived complexity measurements with cognitive weighted values were not yet implemented in this RPA project. Estimating the process complexity with old measurements failed, because time required for completing project was over estimated and that resulted to increased costs and time waste. By applying the complexity measurements that was researched during this thesis, business process would be better measured from all perspectives of business process complexity and estimated time and required know-how for RPA project could be more accurate.

7.2 Validity and Reliability of Research

Reliability of research is assessed by possible errors and misunderstandings of researched data and literature review. If there is a possibility that there are reliability errors in the data used in the research, it will increase the possibility for false assumptions which again will affect the quality of the research. The validity perspective whether someone else would be able to do the similar research and access the data that is used in the research (Saunders et al. 2016, p. 202 – 207).

The literature review of RPA software contains web-based documentation from the RPA software vendor and may pose a reliability risk. As they are published by the software vendor and not likely reviewed by any other party, it is possible that this documentation includes some false information or errors, but since there were no other valid theories regarding the RPA software, this was used as one of the sources. Other used theories, BPM, BPMS, business process complexity and modeling languages are cited from reliable research sites and errors in these are highly unlikely except for perhaps a potential misunderstanding in interpreting the theories.

The empirical data that was used in the empirical research, is gathered from the case company's project documentation. The process documents are always validated by at least two people, which increases the reliability of empirical data. Discussion between project participants was mainly done with e-mails. These documents were automatically stored, and they were in a readable format. The robot log information is also stored in the data store of the case company and each log line includes a time stamp, a job stamp, and process name. All issues that were found during the project were also documented immediately after solution was found. This formed a good and reliable empirical data source for the thesis.

The used literature is accessible for everyone who has access to used article either in paper format or to data store where it is stored in digital format. The only potentially volatile data used in this thesis is the information collected from the vendor of the RPA software. This makes it possible, to reach different results if this type of research is conducted again. The reason for different results could be the changed capabilities in the RPA software or using changed information from the RPA software vendor.

As for the empirical data, the process documentation, automated business process model, and robot log information are only accessible by the case company RPA team and they have control over it. For any researcher to access these, they would need to request a permission the access in advance. Regarding the e-mail conversations, only the persons who were participating in the RPA project have access to these e-mails. But the core information from the e-mail conversations are incorporated to the process documentation and automated business process model. This means that even, if similar research is conducted again there would be the same empirical data available that was used in this thesis.

7.3 Limitations and Future Research

Only one project case was used in the empirical part, so the sample size is quite small. This means that empirical research doesn't have comparison between two different projects. If another project would have been analyzed thoroughly in the thesis, this might have given better understanding for the reasons behind the issues and what needs to be improved in the case company for not to repeat the issues.

RPA is quite new technology, which has not been studied thoroughly in the academic's fields. RPA capabilities and how to measure business process complexity for this type of business process automation software are not yet comprehensively researched. This study has given a direction where to find the relevant measurements. However, these measurements need more research and perhaps even a new perspective of measurements could be introduced. As it was noted in the issue number three, technology capability aspect might require new types of measurements. Another direction for future research would be forming thresholds to determine whether business process complexity is low, medium or high for RPA business process automation.

8. CONCLUSIONS

The complexity measurements and understanding the RPA capabilities were not well known at the time of this study in the case company. Issues in RPA project delivery were noted, and this raised a need for better understanding in business process development and forming measurements in business process complexity. The lack of understanding the causes behind business process complexity increased costs and wasted time during the RPA projects.

RPA is quickly developing new technology with capabilities from all types of theoretically identified BPMS types. To keep up with this rapid development, a continuous training program should be implemented to fully utilize the features of the RPA software. It might also make some of the currently used process automation tools redundant and improve the automation of business processes between the applications used in the organization.

Process knowledge should be well gathered before the modeling of the as-is and to-be process model can start. This requires different kinds of methodologies and recognizing their risks when context is business process automation. The best way would be use all methodologies which are; Workshop-Based, Interview-Based, and Evidence-Based methodologies, to get a full and structured knowledge of the as-is business process.

To be able to utilize the RPA, business process modeling should be at the center of it. Only when a business process is well defined and modeled, automating the business process won't cause any unwanted results. Utilizing a business process modeling language, checking the validity of the process models with 9PMG, and then measuring process model complexity with weighted measurements will result in deeper understanding of the business process. Understanding this complexity will then result in costs and time savings, and the desired outcome of the automated business process can be achieved with the selected RPA software.

9. REFERENCES

Cardoso, J. 2008. Business Process Control-Flow Complexity: Metric, Evaluation, and Validation. International Journal of Web Services Research. Vol. 5, issue 2, pp. 49 – 76.

Corradini, F., Ferrari, A., Fornari, F., Gnesi, S., Polini, A., Re, B., Spagnolo, G.O. 2018. A Guidelines framework for understandable BPMN models. Data & Knowledge Engineering. Vol. 113, pp. 129 – 154.

Dumas, M., La Rosa, M., Mendling, J. & Reijers, A. H. 2013. Fundamentals of Business Process Management. Heidelberg, New York, Dordrecht, London: Springer. pp. 399

Fresht, F. & Slaby, J.R. 2012. Robotic Automation Emerges as a Threat to Traditional Low-Cost Outsourcing. Horses for Sources.

Graml, T., Bracht, R. & Spies, M. 2008. Patterns of business rules to enable agile business processes. Enterprise Information Systems. Vol. 2, No. 4, pp. 385-402

Gruhn, V. & Laue, R. 2006. Adopting the Cognitive Complexity Measure for Business Process Models. Cognitive Informatics, 2006. ICCI 2006. 5th IEEE International conference on Cognitive Informatics. Vol. 1, pp. 236-241.

Gruhn. V. & Laue, R. 2009. Reducing the Cognitive Complexity of Business Process Models. 2009 8th IEEE International Conference on Cognitive Informatics. pp. 339-345.

Macedo de Morais, R., Kazan, S., Dallavalle de Pádua, S.I. & Lucitron Costa, A. 2014. An analysis of BPM lifecycles: from a literature review to framework proposal. Business Process Management Journal. Vol. 20, issue 3, pp. 412-432.

Margherita, A. 2014. Business Process Management System and Activities: Two integrative definitons to build an operational body of knowledge. Business Process Management Journal. Vol. 20, issue 5, pp. 642-662.

Mendling, J., Baesens, B., Bernstein, A. & Fellmann, M. 2017. Challenges of smart business process management: An introduction to the special issue. Decision Support Systems. Vol. 100, pp. 1-5.

Mendling, J., Reijers, H.A., van der Aalst, W.M.P. 2010. Seven process modeling guidelines (7PMG). Information and Software Technology. Vol 52, pp. 127-136.

Mendling, J., Strembeck, M. & Recker, J. 2012. Factors of process model comprehension – Findings from a serios of experiments. Decision Support Systems. Vol. 53, pp. 195-206.

Milanovic, M., Gasevic, D. & Rocha, L. 2011. Modeling Flexible Business Processes with Business Rule Patterns. Enterprise Distributed Object Computing Conference (EDOC), 2011 15th IEEE International. Aug. 2011, pp. 65-74.

Lindsay, A., Downs, D. & Lunn, K. 2003. Business processes – attempts to find a definition. Information and Software Technology. Vol 45, pp. 1015-1019.

Polančič, G. & Cegnar, B. 2017. Complexity metrics for process models – A systematic literature review. Computer Standards & Interfaces. Vol 51, pp. 104-117.

Rolón, E., Ruiz, F., García, F. & Piattini, M. 2006. Applying Software Metrics to evaluate Business Process Models. [WWW-dcocument] [referred 20.10.2018]. Available in: https://pdfs.semanticscholar.org/60d7/c6814a50a6993c0f70316532bba4bbb36873.pdf

Reichert, M. & Weber, B. 2012. Enabling Flexibility in Process-Aware Information Systems Challenges, Methods, Technologies. Heidelberg, New York, Dordrecht, London. Springer. pp. 515.

Saunders, M., Lewis, P. & Thornhill, A. 2016. Research methods for business students. Harlow, Essex. Pearson Education. pp. 741.

Steinke, G. & Nickolette, C. 2003. Business rules as the basis of an organization's information systems. Industrial Management & Data Systems. Vol 103(1), pp. 52-63.

Tarhan, A., Turetken, O. & Reijers, H. A. 2016. Business process matuirty models: A systematic literature review. Information and Software Technology. Vol 75, pp. 122-134.

UiPath. 2018a. UiPath Best Practice Guide version 0.5. [WWW-document]. [Referred 2.8.2018]. Available: https://forum.uipath.com/t/uipath-documentation/16123

UiPath. 2018b. UiPath Enterprise RPA Platform. [WWW-document]. [referred 17.11.2018]. Available: https://www.uipath.com/platform

UiPath 2018c. UiPath Studio. [WWW-document]. [referred 12.11.2018]. Available: https://www.uipath.com/product/studio

UiPath. 2018d. UiPath Robot. [WWW-document]. [referred 2.8.2018]. Available: https://www.uipath.com/robot

UiPath. 2018e. UiPath Orchestrator. [WWW-document]. [referred 20.10.2018]. Available: https://www.uipath.com/product/orchestrator

UiPath. 2018f. UiPath Activities. [WWW-document]. [referred 20.11.2018]. Available: https://activities.uipath.com/docs

UiPath. 2018g. UiPath Studio Docs. [WWW-document]. [referred 20.11.2018]. Available: https://studio.uipath.com/docs

Van der Aalst, W. M. P., Bichler, M. & Heinzl, A. 2018. Robotic Process Automation. Business & Information System Engineering. Vol. 60(4). pp. 269-272.

Vom Brocke, J. & Rosemann, M. (Eds.). 2015. Handbook on Business Process Management 1: Introduction, Methods, and Information Systems, 2nd ed. Heidelberg, New York, Dordrecht, London: Springer. pp. 727

Vom Borcke, J., Schmiedel, T., Recker, J., Trkman, P., Mertens, W., Viaene, S. 2014. Ten principles of good business process management. Business Process Management Journal. Vol. 20, issue 4, pp. 530 – 548.

Zur Muehlen, M. & Indulska, M. 2010. Modeling languages for business rules: A representational analysis. Information Systems. Vol. 35, pp- 379-390.

APPENDICES

Appendix 1: Base metrics for BPMN process models (Rolón et al. 2006)

Element	Metric name	Base measure
Start none event	NSNE	Total number of start none events in the process model
Start timer event	NSTE	Total number of start timer events in the process model
Start message event	NSMsE	Total number of start message events in the process model
Start condition event	NSCE	Total number of start condition events in the process model
Start link event	NSLE	Total number of start link events in the process model
Start multiple event	NSMuE	Total number of start multiple events in the process model
Intermediate none event	NINE	Total number of intermediate none events in the process model
Intermediate timer event	NITE	Total number of intermediate timer events in the process model
Intermediate message event	NIMsE	Total number of intermediate message events in the process model
Intermediate error event	NIEE	Total number of intermediate error events in the process model
Intermediate cancel event	NICaE	Total number of intermediate cancel events in the process model

Intermediate compensation event	NICoE	Total number of intermediate compensation events in the process model
Intermediate condition event	NICE	Total number of intermediate condition events in the process model
Intermediate link event	NILE	Total number of intermediate link events in the process model
Intermediate multiple event	NIMuE	Total number of intermediate multiple events in the process model
End none event	NENE	Total number of end none events in the process model
End message event	NEMsE	Total number of end message events in the process model
End error event	NEEE	Total number of end error events in the process model
End cancel event	NECaE	Total number of end cancel events in the process model
End compensation event	NECoE	Total number of end compensation events in the process model
End link event	NELE	Total number of end link events in the process model
End multiple event	NEMuE	Total number of end multiple events in the process model
End terminate event	NETE	Total number of end terminate events in the process model
Task	NT	Total number of tasks in the process model
Looping task	NTL	Total number of looping tasks in the process model

Multiple instance task	NTML	Total number of multiple instance tasks in the process model
Compensation task	NTC	Total number of compensation tasks in the process model
Collapsed sub-process	NCS	Total number of collapsed sub-processes in the process model
Looping collapsed sub-process	NCSL	Total number of looping collapsed sub-processes in the process model
Multiple instance collapsed sub-process	NCSMI	Total number of multiple instance collapsed sub-processes in the process model
Collapsed compensation sub-process	NCSC	Total number of collapsed compensation sub-processes in the process model
Ad-hoc collapsed sub-process	NCSA	Total number of ad-hoc collapsed sub-processes in the process model
Exclusive data-based decision gateway (XOR)	NEDDB	Total number of XOR splits and joins in the process model
Exclusive event-based decision gateway	NEDEB	Total number of event-based exclusive gateways in the process model
Inclusive gateway (OR)	NID	Total number of OR splits and joins in the process model
Complex gateway	NCD	Total number of complex splits and joins in the process model
Parallel gateway (AND)	NPF	Total number of AND splits and joins in the process model
Sequence flows between activities	NSFA	Total number of flows between activities in the process model
Sequence flows from events	NSFE	Total number of sequence flows from events in the process model

Sequence flows from gateways	NSFG	Total number of sequence flows from gateways in the process model
Looping sequence flows	NSFL	Total number of looping sequence flows in the process model
Message flow	NMF	Total number of message flows between process participants in the process model
Pool	NP	Total number of pools in the process model
Lanes	NL	Total number of lanes in the process model
Data objects (input)	NDOIn	Total number of input data objects in the process model
Data objects (output)	NDOOut	Total number of output data objects in the process model

Appendix 2: UiPath Process Modeling Activities (UiPath, 2018f)

Activity packages	Description
UI Automation activities	Basic activities to interact with applications like humans would, such as clicking, typing and keyboard commands. Within these there are other than selector based application element recognition possibilities, such as OCR, image and text based element recognition.
System activities	Data manipulating activities, interacting activities with directories and files, and activities that interact with UiPath Orchestrator.
Cognitive activities	Integration activities with cognitive engines, text analysis, text sentiment, and text translating activities.
Credentials activities	Activities that works with Windows Credential Manager, with these activities robots can retrieve, update or delete credentials.

Cryptography activities	Activities used for creating security enhancing for sensitive data with encrypting it and then decrypting it for further usage.
Database activities	Database connection activities for sending queries and non-queries.
Excel activities	Used to automate the whole Excel and CSV working activities. There are activities for reading data from Excel or CSV and manipulating data with or without macros.
FTP activities	The File Transferring Protocol (FTP) is used to work with files. These actions are deleting, creating, uploading, and downloading files.
Intelligent OCR activities	Activities in this package are used to analyze and read unstructured documents with intelligent Optical Character Recognition (OCR).
Java activities	This package is used for Java coding in the automated business process. These activities can be invoking Java methods and converting or creating Java objects.
Mail activities	Automated way to robots sending, reading, moving, or deleting e-mails while performing the automated business process by utilizing different kind of mailing protocols.
PDF activities	Reading PDF and XPS files, read text is then stored as string format. If file is scanned, then OCR can be implemented in reading the document to extract the text from it.
Python activities	Enables the capabilities to invoke Python coding and methods in the automated business process.
Terminal activities	These activities are used to get data from the user interface, operate with user interface, and reacting on different events that might occur during process instance.
Web activities	Allows robot to connect directly with some web-based applications and fetch files from web.

Word activities	Enables robot to manipulate Microsoft Word files in ways as human could manipulate the Microsoft Word file and reading the Microsoft Word file. Also saving word file in PDF document is possible.
Workflow foundation activities	These activities are used give the business logic functionality, in terms of control flow, conditions, events, and process states. Communication with applications and services is included in this package.