2018 | Faculty of Business Economics

UHASSELT

KNOWLEDGE IN ACTION

Doctoral dissertation submitted to obtain the degree of
Doctor of Business Economics, to be defended by

Marijke Swennen

DOCTORAL DISSERTATION

Using event log knowledge to support operational excellence techniques

Promoter: Prof. Dr Koenraad Vanhoof

Co-promoters: Prof. Dr Benoît Depaire
Prof. Dr Mieke Jans

D/2018/2451/6

Using event log knowledge to support operational excellence techniques

Acknowledgements

After a journey of almost six and a half years, the end of this thesis has finally become reality. In the end, finalising this dissertation was therefore not only a goal to reach, but also a struggle to overcome. However, I survived this struggle because I knew I would get the chance to thank everyone who was involved throughout this journey. Not only for their support during the work, but especially for being there for me. With this I am certainly not saying that I am not proud of the achievement that I obtained, yet only that it is about much more than only obtaining a PhD degree.

I would like to start by thanking my supervisor, Koen, for the opportunity to work in this great place. I was definitely not the best or easiest PhD student, but Koen gave me the possibility to do what I liked the most, even when this was at the expense of my PhD. So thank you very much, Koen, that I received all these chances throughout the past years to spend the biggest part of my time in this nice environment.

Mieke, my co-supervisor since a couple of years, thank you for helping me getting back on track. Thanks to you we found a new way to manage this PhD, and thanks to you this PhD became reality. Thank you for learning me that trying is not always enough, but above all, thank you for being there for me.

Benoît, the one and only. Thank you, from the bottom of my heart, for seeing always the positive in everything! Thank you for believing in me and supporting me in everything I do. And not only for me personally, but thank you for infecting each and every one of our group with your eternal enthusiasm, at any moment of the day. The way you manage, support, coach, supervise, and connect our group is undeniably unique. Without you, our group would have never been where we are now.

I also want to thank the other members of my doctoral jury for reading this dissertation and for providing valuable feedback. Moreover, I would like to express my gratitude to Jeroen Vergauwe and all other people at Deloitte and at the case study company who participated in this research, for their input and feedback to the different parts of this dissertation.

Clearly, I have been very lucky during the past years at Hasselt University. Doing a PhD is a challenging experience that should not be underestimated. But besides that, I have been able to learn and accomplish so much more than I could have dreamed of, and that is only a small part for what I would like to thank the following people. Because, besides having a great job in which I can fully unfold myself, I also met the most wonderful people during the last six and a half years with whom I spent much more than only the office. I am therefore truly happy that I have the chance to thank these people for what they mean to me.

First, special thanks to Hanne, for everything that she did for me since the moment she started working at the university. Hanne, thank you for being there for me, at any time of the day, for any reason, even at the times you were working on your own PhD. Without a doubt it can be said that 'together we can handle anything' and 'everything will always be ok'. You are not only the best colleague, but also the best friend that I could ever imagine.

Gert, thank you very much for your contribution to my PhD. Without you, this PhD would not be what it is today. But besides this, you became much more than a colleague on such a short time span. We both had our struggles throughout the past years, and your presence is truly invaluable to me. Thank you for being so great!

Niels, your contribution to this PhD is equally invaluable. We collaborated on many different projects throughout the years. And if it concerns research, education, or something completely different, working with you is always a pleasure. But of course we have become much more than only colleagues who can work together. I am therefore indescribably happy that we have become the best friends and that I can always count on you.

Also many thanks to Jeroen, for providing distraction at any time of the day with the most diverse topics and the most useless discussions, and for answering questions that have hardly anything to do with work. If there is anyone who can make me smile, both during and after office hours, it is you. And Stef, also thanks to you, for always being there for me, at any time of the day.

Together with these people, I already enjoyed the most beautiful and crazy moments during my spare time, at amusement parks, city trips, kayaks, long walks, and even at simple dinners in a restaurant. Thank you all!

Furthermore, I would like to thank Toon for always cheering up our office with his lame jokes and impressions, which provide distraction from all other things, even early in the morning. And Lotte, thank you for making sure I felt good from the first day in this new environment and for being there for all big and small issues. Frank, Mathijs, Teun, Hilde, Gonzalo, and all other colleagues of the research groups

Business Informatics and Logistics, it is ab-so-lute-ly a pleasure working with all of you. Thank you so much for creating such a unique atmosphere within our group.

Also many thanks to all other colleagues that I worked with, at IMOB and the main building, all colleagues that support our group every day with educational and administrative work, and all colleagues that I just pass in the hallway for a nice chat.

Next to my colleagues, I also want to thank the people outside this work environment, for being there for me and for all the nice moments that we spent together during the past years. Special thanks go to Bram, Brecht, Anneleen and Fréderic, Carmen and Niels, Kristof, and all other people that bring me joy.

Last but definitely not least, I would like to thank my family for their unrelenting support and understanding. I know it has been a big question mark what I was working on during the past years, and if it would result in anything good. I hope this dissertation can answer these questions. Together we have been through a lot the last months, which can make something as a job suddenly really unimportant. Therefore, I would like to thank you all for taking such good care of me.

And a final word of gratitude is for the most important person in my life. Saying thank you is absolutely not enough to show how happy I am with the best mom in the world. Not only did she provide me the last years with a weekly supply of waffles, cupcakes, and other kinds of sweets, which I often had to hand out to get it all processed. Not only did she help me with household tasks, cook for me as if it was for a whole army, and make sure my life did not stop when I was living alone. Mom, the way you are always out and about helping me, dad, Jo, Marlies, and Juul, and so many other people around you, is indescribably beautiful. Thank you for everything!

Marijke Swennen
20 February 2018

Contents

List of Tables

List of Figures

Chapter 1

Introduction and research methodology

1.1 The use of process mining in operational excellence

Companies nowadays are comprised of a large set of business processes that are sometimes deeply intertwined with each other, making it hard to get a decent overview of the different flows of data, activities and resources. Besides, as new technologies and customer expectations grow faster than ever, companies strive to modify and improve their processes and ways of working continuously. To keep up with changing environments, business processes should be monitored at a constant rate and companies should implement process analysis methods and improvement teams. The concept of continuous improvement has been found by Bigelow [18] and Drohomeretski et al. [46] to be related to concepts such as lean manufacturing, Six Sigma, Business Process Improvement (BPI) and business re-engineering in the Total Quality movement. Because these individual programmes are sometimes insufficient for companies, according to van Assen [154], operational excellence can be reached by the combination of different elements. First, the production and delivery system should be analysed based on the reduction of lead time and the management of variability. Next to this, concepts such as lean management and Six Sigma can be used for business (process) improvement and optimisation matters. Moreover, resources such as people and machines should be handled in a clever way to enable continuous improvement. And finally, strong leadership and a good change management system should be in place.

Lean management is mainly focused on the reduction of waste which can be formalised as, among others, over-production, waiting time, excess processing, delays or batch processing [35, 110, 154]. Waste reduction can be operationalised by identifying and analysing the value flow of a business process, implying that only activities and tasks that add value to the customer should be executed. Six Sigma, on the other hand, focuses more on the quality of business processes and aims for the minimisation of defects and errors in the process, in order to eliminate variation and improve the overall quality of the company [6, 8, 93]. The most well-known improvement method in the field of Six Sigma is the DMAIC-cycle, which stands for define, measure, analyse, improve, and control [6, 84, 154]. Next to lean management and Six Sigma, another improvement methodology that has been recognised is the theory of constraints (TOC), which is more focused on detecting and removing constraints or bottlenecks in business processes [110]. Although these methods and techniques have been employed for many years in industry, they are defined to be less useful in volatile or fast changing environments in which service companies are operating. Many of these operational excellence techniques are also still based on rather qualitative and "paper-and-pencil"-based approaches implying that the results of these methods and the decisions taken based on these results may be rather subjective and dependent on the person or team that has been performing the analyses.

Moreover, most studies found in literature focus on the application of lean management and Six Sigma in manufacturing environments. Even though all executed work can be seen as part of a process that possibly includes variability, and each process produces data that can possibly explain this variability, service-oriented companies are until now less convinced of the use of these techniques within their organisation, mostly because of the more volatile and fast-changing environment.

In this light, process mining is recognised to be a potential supporting tool in the field of operational excellence as it focuses on the analysis of business processes in order to get an insight into the processes and to improve them accordingly. Process mining refers to the retrieval of knowledge from process execution data, which is stored in so-called event logs. It mainly entails three types, which are (i) the discovery of process models from event logs, (ii) compliance checking of these discovered models with the underlying event log, and (iii) enhancing the processes accordingly [155]. Event logs are the starting point for a process mining project, as they are composed of data from process-aware information systems (PAIS). An event represents "something" that happens within the process and is captured by a PAIS. It can, for example, refer to the moment at which a clerk starts handling a specific file or to the moment at which a specific task is completed for a customer. A wide range of algorithms has been presented to discover process models from event logs, with the alpha-algorithm as the one that was introduced first [155]. However, these algorithms are mostly based on parameters and assumptions that should be chosen by the process analyst to simplify the process model discovery. As the models that are discovered with these algorithms can therefore become too precise or too generic to reflect the actual behaviour in the business process, our definition of process mining also covers the retrieval of knowledge from event logs without first discovering a process model.

Given the potential of process mining in the field of operational excellence and its recognition as a key challenge for process mining research in the Process Mining Manifesto [160], further research is required on this topic. Existing research efforts seem to be limited and it is not always clear from literature how existing techniques are used to support the process of business process performance measurement and improvement.

1.2 Research objective

A significant research gap exists on the interplay between operational excellence and process mining, implying the need for additional research efforts. Existing operational excellence techniques require more data-based analyses in order to be more objective and therefore more effective. Process mining is a promising field to support these operational techniques, but focuses too often on model-based analyses. Therefore, this dissertation addresses the following **central research question**.

> *How can process mining be applied to business processes in order to complement existing operational excellence approaches?*

The first research objective of this dissertation is therefore the investigation of the problem in order to create an overview of the requirements of the artifact that is needed to solve the problem at hand. To do this, first a literature review on both operational excellence and the interplay between operational excellence and process mining is performed. This way, the problem context will be outlined. The findings from this literature review are then complemented with the findings from a list of interviews that are conducted with three business experts to make sure that the requirements and needs are confirmed by practitioners. Based on this comparison and the analysis of the findings from literature and business experts, the requirements of the artifacts that should be developed to solve the solution can be outlined precisely.

Once the requirements of the artifact are outlined, **the second research objective** of this dissertation entails the identification and development of the artifact(s) that fulfil these requirements. From the problem investigation and requirement analysis it will be clear that the required artifacts use unbiased event log knowledge to support operational excellence techniques. In contrast to most process mining research that is concerned with the alignment between the behaviour that can be observed in the discovered process model and the behaviour in the underlying event log, the artifacts that are required to solve the shortcomings focus on the objective measures that can be directly learned from the event log. In order to investigate the value of the developed artifacts, different evaluation methods will be administered, as the artifacts will be applied to both artificial and real-life cases, and they will be discussed in an iterative manner with business experts during the development and evaluation stages.

1.3 Outline of the thesis

Given the limited existing work on the use of event log knowledge in the light of operational excellence, the first part of this dissertation (Chapters 2-3) contains a literature overview of existing techniques in the field of operational excellence and on the interplay of operational excellence and process mining. From this foundation, the second part of the dissertation (Chapters 4-7) focuses on the development of methods that retrieve event log insights to support specific operational excellence concepts. Throughout this dissertation, the focus will be on service companies, in which the resources mainly concern people. In Figure 1.1 an overview of the outline of the thesis is provided.

Chapter 2 starts with an introduction to the evolution of quality management and the emergence of different methodologies in the field of operational excellence. Improvement concepts such as lean management, Six Sigma and the theory of constraints will be elaborated and compared. As it is not intended to provide an exhaustive overview of all existing philosophies, this chapter will provide an identification of the underlying principles of the operational excellence field, in order to analyse where process mining, which is discussed in Chapter 3, can be useful. From this analysis, it can be concluded that not one single strategy or roadmap exists for companies to follow in order to improve their processes, and a combination of multiple methods and tools should be incorporated. Moreover, the existing analysis techniques are often rather qualitative and based on paper-and-pencil approaches with a lack of support of data-based analyses [118]. This also implies that the operational excellence approaches are rather subjective and depending on the person performing the analyses and the team involved in the process.

Chapter 3 supplements these findings with an introduction to the field of business process management and process mining, and an outline of the match between process mining and the philosophies and principles of operational excellence discussed in Chapter 2. Here we find claims that specific guidelines for business process optimisation are limited in literature and that it is not always clear how existing techniques are used to support the process of business process improvement. Next to this, most of the research on process mining is focused on discovering process models from event logs and checking the conformance between these two. These models are learned from event logs with certain algorithms, based on parameters and assumptions, and are often manually manipulated with sliders and filters, implying that unobserved behaviour possibly appears in the model. Conclusions taken based on these models can therefore be less reliable or even incorrect as they possibly contain unobserved behaviour or they do not contain all information of the business process under analysis.

In order to address this shortcoming, this dissertation will introduce the concept of parameter-free log-based process metrics that present objective measures that are directly learned from the event log. The requirements for these metrics are identified in **Chapter 4**. Firstly, both from the findings in literature and from interviews conducted with business people, an exact overlap was found between the different categories of process performance measures that should be focused on in a business process improvement project. These categories, which are often referred to as the Devil's Quadrangle [47], are (i) time, (ii) cost, (iii) quality, and (iv) structuredness. Secondly, the measurements should also be executed on different levels of analysis in order to provide a realistic view on the underlying process. Therefore, the levels of analysis that arose from the interviews range from the complete end-to-end process to the specific combination of a resource executing a specific activity. Moreover, the developed measures should contain clear descriptions of the measure itself, the requirements for event data, and the underlying calculation, as this is often missing in existing performance measures. And finally, the artifacts that are created in order to overcome the lack should be understandable for business people. This can be realised by adding suitable visual representations and a translation of technical concepts to concepts that are interpretable by the process owners.

The goal of **Chapter 5** is to introduce the concept of log-based process metrics, which provide a picture of the present process behaviour that is not biased by a model,

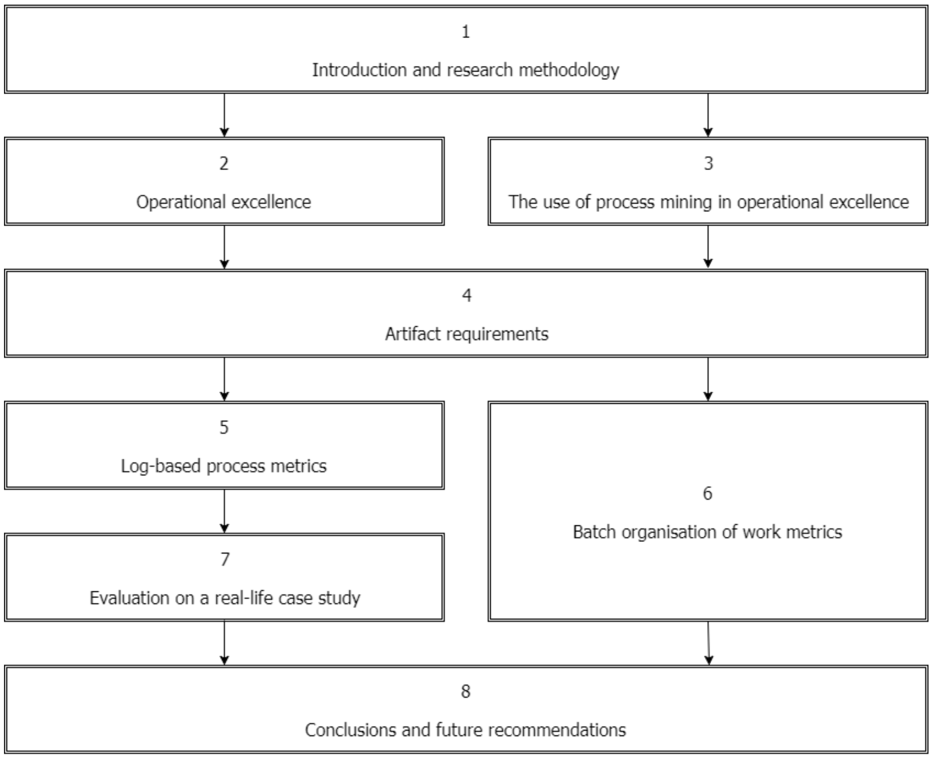

Figure 1.1: Outline of the thesis.

and can be used to compare different event logs in an objective manner. The presented metrics are structured along two dimensions, which are time and structuredness, and are calculated on one of the analysis levels: log-, case-, trace-, activity-, resource- or resource-activity level. Moreover, they are applied to a running example and the results are discussed. Finally, a dashboard that has been created to visualise all metrics is provided.

One of the operational excellence concepts that was stated to be an indication of waste in a business process is batch processing. Batch processing influences process performance as it can, for instance, lead to longer waiting times for certain cases when multiple cases are gathered before processing starts [184]. Consequently, it should be taken into account when modeling and evaluating business processes. To this end, insights in batching behaviour should be generated, which is the topic of **Chapter 6**. Firstly, the concept of batch processing is elaborated upon and three types of batch processing are distinguished and formally defined. Secondly, a resource-activity centered approach is presented to identify these batch processing types from an event log.

Finally, batch processing metrics are defined to gain insight in the characteristics of the identified batches and the implications of batch processing on process execution. Given these contributions, the Batch Organisation of Work Identification algorithm (BOWI) is presented in order to provide useful insights in the batch organisation of work and its influence on process execution.

To demonstrate the applicability of the presented log-based process metrics in Chapter 5, and their added value in the light of operational excellence, they are applied to a real-life event log of a Belgian utilities company in **Chapter 7**. The process under analysis presents the total flow from the request made by a customer to the aftercare which includes the invoicing. From this application, it can be stated that the presented metrics represent the business process behaviour in an objective way, without any influences from underlying algorithms or assumptions. Moreover, different groups of event logs can be more easily and correctly compared and analysed, both over time and based on certain case attributes such as region or building.

The final chapter of this thesis, **Chapter 8**, summarises key conclusions from the various chapters and highlights interesting directions for future research.

1.4 Research methodology

1.4.1 Introduction to design science research

This dissertation will be conducted following the principles and steps of the design science research (DSR) methodology. The origins of design science research are often traced back to Simon [142]'s *The Sciences of the Artificial* [14, 101]. Johannesson and Perjons [78] define design science as *"the scientific study and creation of artifacts as they are developed and used by people with the goal of solving practical problems of general interest"*. In other words, design science is a research methodology that is used to develop an artifact that acts as a solution for a practical problem in reality. Therefore, in contrast to natural or behavioural science, design science is not only focusing on understanding and explaining a specific situation in the world [3, 112, 115], but also on changing and improving this situation. However, despite the outlined differences between natural science and design science, Niehaves [112] states that both are not mutually exclusive, but complementary.

In the past years, there has been some discussion on the exact terminology of design science research. According to Iivari [71], the concepts *design science* and *design research* are used interchangeably in literature. However, Alturki et al. [4] and Johannesson and Perjons [78] declare that design science can be seen as a special kind of design research. According to Winter [181] *"design research is aimed at creating solutions to specific classes of relevant problems by using a rigorous construction and evaluation process, and design science reflects the design research process and aims at creating standards for its rigour"*. This distinction is based on the difference between the *science of design*, which is artifact construction and evaluation at a generic level, and *design science*, the construction and evaluation of specific artifacts, defined by Cross [31]. In this sense, it is not the finality of design science research to solve a specific problem in a specific context, but to develop an artifact and its associated knowledge that enables solving a class of problems [173]. From the previous, it follows that design science research should not focus on one specific case. It has to generate generalisable knowledge and principles that can be applied to a particular class of problems.

According to Alturki et al. [4], there is no agreement on the definition of an artifact. Mettler et al. [104] refer to the definition provided by Simon [142], which is *"The term artifact is used to describe something that is artificial, or constructed by humans, in contrast to something that occurs naturally"*. Johannesson and Perjons [78] add to this definition that an artifact is created *"with the intention that it be used to address a practical problem"*. They also state that artifacts are always embedded in a larger context. As was already stated by McKay and Marshall [101], two important aspects of artifacts are thus that they are (i) created by humans and (ii) have utility.

Venable et al. [174] distinguish between two categorisations of artifacts. On the one hand, a distinction is made between product and process artifacts. While the former is used by stakeholders to perform an activity, the latter describes how an activity can be performed. On the other hand, technical and socio-technical artifacts are distinguished, depending on the necessity of human stakeholders to actively interact with the artifact to achieve its potential [174]. Gregor and Hevner [61] state that design science research in the field of information systems involves the development of socio-technical artifacts. March and Smith [94] identified four types of artifacts which have been adopted by many other researchers afterwards [13, 19, 49, 68, 78, 95, 104, 115, 171, 181]. These types are constructs, models, methods, and instantiations, where the latter can be seen as the aggregations of the previous types of artifacts in specific problem situations. Goldkuhl and Lind [58] separate an instantiation from the other artifact types as the former can be seen as the demonstration of knowledge, while constructs, methods, and models are considered as meta-artifacts. According to Venable [171] and Winter [181], Rossi and Sein [130] add better theories and testable design process hypotheses as additional artifacts. Also Peffers et al. [116] add two extra artifact types to this list, which are algorithms and frameworks.

Although the design science research field is growing steadily, no commonly accepted framework of definitions, methods, and methodological considerations is present [4, 181]. Braun et al. [19] also state that there still is no agreement on design science methods, techniques, and procedures.

1.4.2 Design science research framework

Besides the potential of DSR to develop artifacts to solve problems and contribute to the knowledge base, several issues can be detrimental for the project's success. Baskerville [14] identifies a series of these risks such as an inadequate problem specification and an insufficient analysis of the existing knowledge base. As a consequence, following a structured and systematic approach towards DSR can support a researcher to avoid these risks to materialise.

In literature, several methodological frameworks have been proposed to outline the key activities that need to be conducted when performing design science research. Illustrations of such frameworks can be found in [4, 5, 15, 16, 19, 20, 27, 68, 56, 78, 94, 101, 105, 114, 117, 141, 153, 171, 172, 180, 181]. These provide guidance to researchers wishing to apply the design science principles [171, 173] and presents an effort to standardise the creation of knowledge [112]. Even though these frameworks tend to have common grounds, Alturki et al. [5] and Mettler et al. [104] state that no consensus is reached as the activities they prescribe can differ. Braun et al. [19] also make note that methods and procedures to conduct design science research with broad support are still lacking.

Despite the wide range of design science activity frameworks and the apparent lack of consensus between them [5, 104], an analysis of the aforementioned DSR frameworks shows that they all have common grounds. Differences mainly stem from the fact that, on the one hand, different frameworks emphasise different activities when conducting design science research and, on the other hand, differences in the scope of the framework, i.e., to which extent factors surrounding the DSR project are taken into account.

Given the previous observations, this dissertation will use a synthesis framework, which is visualised in Figure 1.2. This synthesis framework combines the strengths of several existing frameworks, while preserving its clarity in order to support researchers aiming to conduct design science research. This framework strikes a balance between simplicity and clarity on the one hand and completeness on the other hand.

The developed synthesis framework consists of three components: the problem context, the knowledge base, and the DSR-project. The DSR-project is comprised

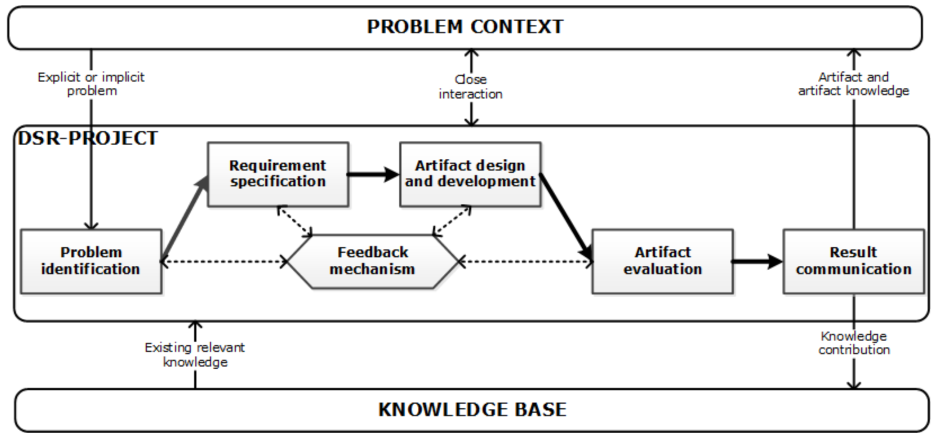

Figure 1.2: Design science research synthesis framework (research in progress)

of five steps: problem identification, requirement specification, artifact design and development, artifact evaluation, and result communication. Moreover, a central position is attributed to the feedback mechanism of the DSR-project. After the design science project has been completed, the artifact and the knowledge about this artifact are communicated to the different stakeholders, which are positioned within the problem context. Interaction with the knowledge base is necessary in order to build on existing knowledge and contribute the design science project findings to the knowledge base for further research. The remainder of this section will briefly discuss the steps in a DSR-project, mainly focusing on how they are applied in this dissertation.

1.4.3 Problem identification

As indicated in the introduction of this section, a DSR-project is instigated by a particular problem originating from a problem context [19, 58, 78, 173]. A wide variety of problems can be tackled, ranging from vague symptoms of an implicit problem to explicitly articulated problems. During the problem identification step in a DSR-project, the researcher has to gain a clear and precise understanding of the problem at hand [171]. The problem domain in this thesis is therefore firstly explored by conducting a literature review of operational excellence which aims to highlight its strengths and weaknesses.

Chapter 2 contains an overview of the evolution of quality management and some well-known improvement concepts such as lean management, Six Sigma and the theory of constraints. From the observations in this literature review it can be seen that many existing techniques are based on qualitative and paper-and-pencil approaches, implying that the actual problems and details are often ignored. Moreover, the requirements for the data that is used for the analyses are often not clear and the principles of lean management are also found to be less useful in volatile or fast changing environments. Six Sigma tools, which are more statistically underpinned, are still experienced to be rather subjective as they are often based on interviews and human opinions.

Next to this, the use of process mining in the field of operational excellence is explored. As process mining is related to the business process management field which focuses on business process improvement, the link with operational excellence, which is focused on optimising the operational ability of companies, is straightforward. However, from the literature overview it was found that specific guidelines for business process optimisation are limited in literature and that some authors in literature claim that it is not always clear how existing techniques are used to support the process of business process improvement. Next to this, most of the research on process mining is focused on discovering process models from event logs and checking the conformance between these two, while process models can be biased representations of the business process as they are based on algorithms and assumptions. Based on these findings, the next step in the design science research approach is to define the requirements for a suitable solution to solve the problem at hand.

1.4.4 Requirement specification

In the requirement specification step, an artifact that could address the problem is identified and described [78]. Moreover, requirements which have to be taken into account when this artifact is designed and developed are outlined [78, 117]. In DSR literature, several types of artifacts are distinguished. A commonly used typology is proposed in March and Smith [94], where a distinction is made between constructs, models, methods, and instantiations [13, 49, 78, 104, 115].

Constructs represent terminology, definitions, and concepts that are required to express the problem and its potential solutions. By interconnecting these constructs, models are used to represent a solution for the problem at hand. Taking both constructs and models as an input, methods prescribe a series of steps through which a solution to the problem can be created. An instantiation is the operationalisation of constructs, models and methods within a particular environment, resulting in an operational system [78, 94].

Chapter 4 identifies the requirements that emerged from literature, which is outlined in Chapter 2 and Chapter 3, and the interviews that have been conducted with people from industry in order to better understand the shortcomings in practice. Given the potential of process mining in the field of operational excellence and the recognition as a key challenge for process mining research in the Process Mining Manifesto [160], further research is required on this topic. Existing research efforts seem to be limited and it is not always clear from literature how existing techniques are used to support the process of business process performance measurement.

1.4.5 Artifact design and development

Using the artifact outline from the requirement specification phase, an artifact is actually designed and developed in this step, taking into account the defined requirements [78]. The necessary design and development efforts can range from small adjustments to existing artifacts to establishing a completely new artifact [171].

In this thesis, significant design and development efforts are required as the artifacts do not have an incremental nature, i.e., they are not merely minor modifications to existing artifacts. Chapter 5 presents an overview of useful, log-based process metrics that provide an unbiased picture of the present process behaviour. Additionally, these metrics can be used to compare different event logs in an objective manner. Next to the definitions and statements explaining the metrics, a visualisation of the metrics, presented as a dashboard, is also provided in order to fulfil the requirement of understandability and interpretability for business people.

Chapter 6, thereafter, presents a list of metrics concerning the concept of batch processing, which was defined as one of the indications of waste in a business process. Next to the list of batch processing metrics, a batch organisation of work identification (BOWI) algorithm is developed, to discover different types of batch processing from an event log.

1.4.6 Artifact evaluation

The following step in a DSR-project is the evaluation of the artifact. Johannesson and Perjons [78] make an explicit distinction between artifact demonstration and evaluation. While the former aims to demonstrate that the artifact can be applied by using it in a particular case, the latter determines the ability of the artifact to tackle the problem at hand and the degree to which the specified requirements are fulfilled. The synthesis framework in Figure 1.2 combines all evaluative actions in a single step, as is the case in the frameworks proposed by Hevner et al. [68], Sein et al. [141], and Venable [171, 172].

Several evaluation methods can be distinguished. For instance: Peffers et al. [116] consider eight ways of evaluating artifacts, i.e., a logical argument, expert evaluation, a technical experiment, a subject-based experiment, action research, a prototype, a case study, and an illustrative scenario.

From these evaluation methods, this thesis mainly uses expert evaluations and illustrative scenarios. Expert evaluation involves artifacts being judged by one or more experts in a particular field [116]. This is used to evaluate the usefulness of the log-based process metrics in Chapter 5. Moreover, papers on this work have been submitted to the peer review system of different conferences. Illustrative scenarios require the application of an artifact to artificial or real-life situations to show its usefulness [116]. In this dissertation, the introduced metrics in Chapter 5 are applied to both artificial event logs, within the same chapter, and to a real-life event log in Chapter 7. The former is also included in the evaluation framework of Hevner et al. [68] and is referred to as simulation evaluation, i.e., the application of an artifact on artificial data. The dashboard that is presented in Chapter 5 is evaluated by expert evaluations on different international conferences and in a real-life business process. The batch processing metrics and the BOWI algorithm to discover batch processing from event logs, which can be found in Chapter 6, are applied to both an artificial event log and to real-life event logs.

1.4.7 Result communication

The final step of a DSR-project involves the communication of the artifact and knowledge about this artifact [78]. Communication should also frame the problem at hand and stress its relevance [117]. The importance of spreading the results of a DSR-project is also emphasised by Hevner et al. [68] in one of their seven DSR guidelines.

Besides the current thesis, formal communication to an academic audience is performed by means of presentations at national and international conferences and publications in peer-reviewed conference proceedings [77, 98, 148, 149]. Moreover, the research output presented in Chapter 6 is published in a scientific journal [99].

At an informal level, the research results are communicated at conferences and events where people from industry are present. All results of the evaluation that has been applied to the real-life case study in Chapter 7 have also been discussed in detail with two process experts and some other employees of the organisation who are all involved in the process under analysis. The importance of the latter, i.e., communication towards a non-academic audience, is stressed by both Hevner et al. [68] and Peffers et al. [117].

1.4.8 Feedback mechanism

Until now, DSR is presented as a linear process, starting from identifying the problem, followed by defining an artifact and its requirements, designing and developing the artifact, evaluating it, and communicating the key results. However, throughout the study, it might be required to return to a prior stage, as recognised by, amongst others, Peffers [117] and Venable [171, 172]. Such a feedback mechanism is integrated in the synthesis framework in Figure 1.2 indicating the need for continuous assessment.

Continuous assessment implies that during each step, project responsibles should remain aware of the outcome of prior steps and determine whether it is required to revisit a particular step to revise its output. Some decisions that seemed appropriate at a particular point in time, for instance related to the requirements, might not be valid when the artifact is being developed. Suppose that during artifact development, the project staff concludes that the implementation of particular features requires significantly more efforts than anticipated. When the budget is fixed, this conclusion might require to return to a prior step to adjust the project's scope. Braun et al. [19] provide another example by stating that requirements can be adjusted during the project, anticipating upon new insights gathered along the way. Continuous assessment is not modelled as an autonomous activity as it can be better characterised as an attitude that is required throughout the project. The latter explains its different shape in Figure 1.2.

Within this thesis, continuous assessment mainly gave rise to a return from artifact evaluation to the design and development step. These steps are closely intertwined when, for instance, the application of a developed method to an artificial dataset leads to the conclusion that particular aspects within the method need to be revised. Another illustration is the addition of artifact requirements during its development, as suggested by Braun et al. [19]. During the development of the metrics presented in Chapter 5, a first evaluation step was already done by applying the metrics to an artificial event log, which provides direct indicators for the need for extra levels of analysis or applications that could be useful in other, related metrics. The need to make the results and analyses visual and interpretable for the process owners also arose after presenting the results of the developed and applied metrics to the business people. Also other smaller changes to the requirements arose during these discussions, implying a return to the requirement specification step.

1.5 Conclusion

This thesis focuses on the use of event log knowledge, i.e., process mining, to support business process improvement in the light of operational excellence. Despite the potential of process mining to support operational excellence concepts such as lean management and Six Sigma, literature on the systematic use of event log knowledge within this context is limited. In order to overcome this research gap, two research objectives are covered with this dissertation. On the one hand, an analysis of existing techniques and uses in the field of operational excellence is provided. Given this foundation, the existing work is supplemented with novel methods that retrieve event log insights to support specific operational excellence concepts. The artifacts that are created are parameter-free log-based process metrics.

Chapter 2

Operational excellence

2.1 Introduction

Customer expectations, new technologies and growing global competition drive companies to continuously modify and improve their business processes. These processes are therefore dynamic by nature and constantly changing. Different methodologies and philosophies are developed to implement these business changes. Nowadays, companies require world-class operating systems and processes which include decent control and planning, good information systems that execute the company processes in an efficient and effective way, and a culture of continuous improvement to become operationally excellent [154]. Following the operational excellence cycle presented by Bigelow [18], which is shown in Figure 2.1, clear requirements should be established, effectively communicated, and continuously assessed. Different internal and external sources such as customers, suppliers, corporate policies, or qualification and validation protocols, provide these requirements. After defining the necessary requirements, all people involved should be informed and trained in order to be familiarised with the requirements. Finally, compliance to the established requirements should be maintained by continuous assessment and auditing. This can only be achieved with a management committed to total compliance, quality, and continuous improvement.

The concept of continuous improvement has been identified by Bigelow [18] and Drohomeretski et al. [46] to be related to methodologies such as lean manufacturing, Six Sigma, business process improvement (BPI), and business re-engineering, which can be positioned in the Total Quality movement. Because these individual programmes are sometimes insufficient for companies, hybrid methods such as Lean Six

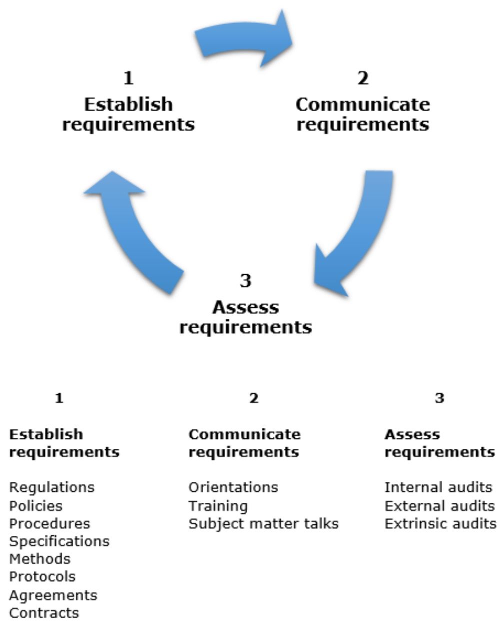

Figure 2.1: The operational excellence cycle [18].

Sigma have been introduced. Also van Assen [154] states that operational excellence can only be reached by an aggregation of different elements. First, the production and delivery system should be analysed, optimised, and controlled by operations research models that are based on the reduction of lead time and the management of variability. Next to this, concepts such as lean management and Six Sigma can be used for business (process) improvement and optimisation matters. Moreover, people and machines should be handled in a clever way to enable continuous improvement. And finally, strong leadership and a good change management system should be in place.

The goal of this chapter (Figure 2.2) is threefold. Firstly, (i) an introduction to the evolution of quality management and the emergence of different methodologies is given. Secondly, (ii) improvement concepts such as lean management, Six Sigma, and the theory of constraints will be discussed and compared. However, it is not intended to provide an exhaustive overview of all existing philosophies, yet (iii) an identification of the underlying principles will be presented, which is the third objective of this chapter.

Section 2.2 contains an overview of the evolution of quality management, followed by an introduction to the main concepts used throughout this text in Section 2.3. Next, an introduction to the lean management philosophy is provided in Section 2.4,

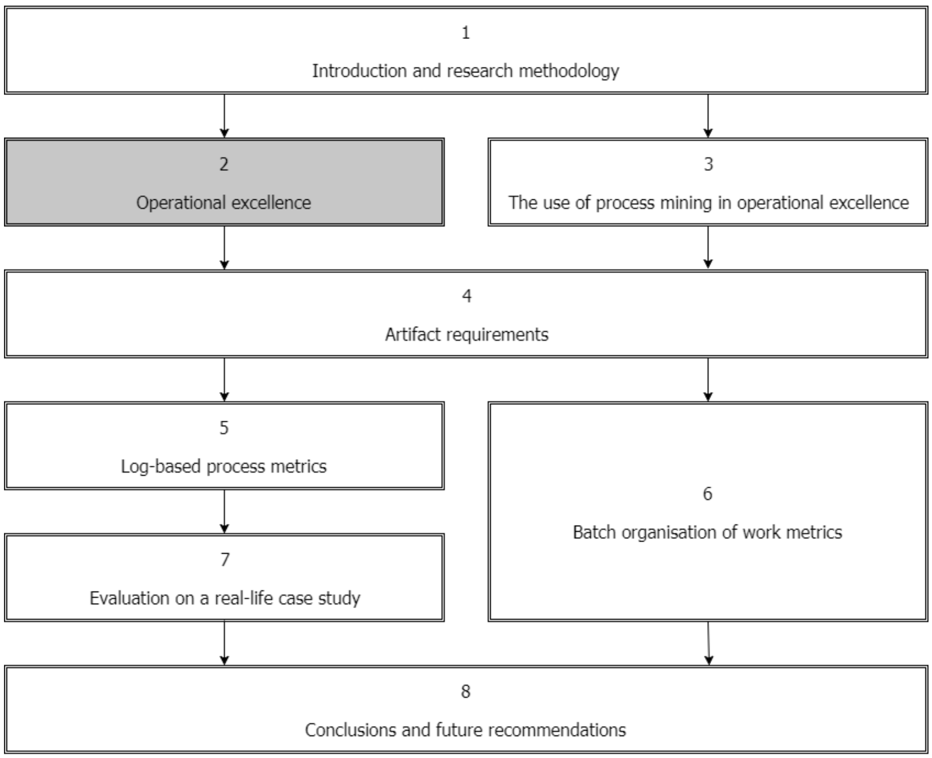

Figure 2.2: Outline of the thesis - Chapter 2.

including different tools and techniques that have been used in literature. Furthermore, Six Sigma and its position within the total quality management field is discussed in Section 2.5. Also the concept of variability and the hybrid methodology Lean Six Sigma are presented here. Finally, the theory of constraints is explained in Section 2.6. After these methodologies have been introduced, their similarities and differences are analysed in Section 2.7, followed by an overview of the underlying principles of these methodologies in Section 2.8. Conclusions are drawn in Section 2.9.

2.2 Evolution of quality management

Quality control goes back in time until the beginning of the 20th century. Different quality evolution frameworks exist to indicate the different periods over time. According to Garvin [55], four separate stages could be identified based on the time in history: *inspection*, which can be located from 1900 until the pre-1930s; *statistical quality control (SQC)*, from the 1930s onwards; *quality assurance (QA)* starting in the 1950s; and *strategic quality management (SQM)* which only started in the 1980s. In the inspection phase, specialists were employed to inspect the quality of the products in order to pursue product uniformity, while the focus in the SQC phase is on the manufacturing process instead of the final product. This way, the products are standardised and manufactured according to the requirements of the customers. Statistical tools and techniques were used, in comparison to the gauging and measurement techniques in the inspection phase. In the QA stage, not only quality specialists were involved, but the full workforce and the management are included to prevent quality failures. In the SQM phase finally, the focus is completely on the needs of the end customer and the market in order to gain a competitive advantage. Total Quality Management (TQM) and Six Sigma only arose after Garvin [55]'s work, but these two programmes can be considered to be equivalent to the SQM era [126]. In 1995, Tuckman [152] developed a framework focusing only on the late 20th century quality developments or the SQM stage in Garvin [55]'s work. First, between the late 1970s to early 1980s, quality circles were developed. During the 1980s, the concern of major companies shifted to the control of suppliers and sub-contractors. From the mid-1980s customer satisfaction received more attention in the manufacturing and service sectors. Finally, from the late 1980s, new areas such as public services became more aware of quality. Some years later, Dooley [45] also expanded the work of Garvin [55]. He predicted the next quality phase and established a link between Total Quality Management (TQM) and the SQM stage in Garvin [55]'s work. The three paradigms defined by Dooley [45] are caveat emptor (pre-industrial), quality control (industrial) and TQM (post-industrial). In this last stage, Dooley [45] predicted that organisational learning and participative management would become more important. Finally, Dahlgaard [33] criticised Garvin [55]'s work for being too much focused on Western companies and being too technical. Therefore, he analysed the Japanese quality evolution and defined that the Japanese culture first imported and learned from the West during the mid-1940s until the early 1960s. Later the imported ideas were improved and implemented into the Japanese culture and from the early 1970s until the early 1990s, the ideas were further mastered and exported to other countries. The research concerning the TQM and Six Sigma concepts is never adequately mentioned in one of these frameworks. Therefore, Rajamanoharan et al. [126] developed a flow structure which is given in Figure 2.3 based on all frameworks discussed here, leading to the development of the Six Sigma and TQM concepts. The four stages identified by Garvin [55] can be found in this top-to-bottom structure. The triangle becomes wider at the bottom to illustrate the application of the frameworks and each quality paradigm in the triangle is connected with a previous or next quality paradigm

Figure 2.3: Flow structure of quality management [126].

to indicate the sequence between them.

In the remainder of this chapter, lean management (Section 2.4), Six Sigma and its position within the broader concept of TQM (Section 2.5), and the theory of constraints (TOC) (Section 2.6) will be discussed, followed by a comparison of these methodologies in Section 2.7. However, to make this overview of existing methodologies easier to understand, first some general concepts concerning the field of operational excellence are introduced in the next section.

2.3 Main concepts of operational excellence

Operational excellence can be defined as a management philosophy that focuses on the excellence or superiority of an organisation based on its overall strategy. The philosophy is mainly concerned with the continuous improvement of an organisation's processes and operations [18, 154]. It contains certain concepts that will be used throughout this dissertation, which are defined in Table 2.1.

In order to measure and determine the level of operational excellence of an organisation, analyses should be undertaken. *Analyses* are detailed examinations of the elements or structure of something, such as a business process or a certain procedure. These analyses can be undertaken with certain *methods*, which are the tools, techniques, or processes that are used to perform analyses and calculations. An extensive overview of tools and techniques used in the fields of the lean management and Six

Table 2.1: Concepts of operational excellence.

OE concept	**Description**
Analysis	A detailed examination of the elements or structure of something, such as a process.
Method	A tool, technique, or process used to perform analyses and calculations.
Tool	An instrument to carry out a specific function.
Technique	A manner to perform certain tasks or an ability to employ certain skills.
Methodology	A (research) strategy that outlines the way analyses are undertaken and how methods are used for this.
Principle	A fundamental truth or a collection of propositions that serve as the foundation for a methodology.
Metric	A quantifiable or verifiable measure that is used to track and assess the status of a specific process.
KPI	A measurable value that demonstrates how effectively a company is achieving key business objectives.

Sigma will be provided in this chapter. Although these concepts are often used interchangeably, *tools* can be defined as instruments to carry out a specific function, and are therefore more narrow in focus, while *techniques* are manners or abilities to perform certain tasks, which have a wider application and require specific skills or training. Moreover, as was mentioned before, the philosophy of operational excellence is based on some *methodologies*, which are defined as strategies that outline the way analyses are undertaken and how methods are used for this. The methodology concept is often used interchangeably with paradigm and philosophy. Methodologies in the field of operational excellence that will be discussed in this dissertation are lean management, Six Sigma, total quality management, and the theory of constraints. These methodologies are based on certain beliefs and ideas, which will be defined as principles in this dissertation. *Principles* are the fundamental truth or propositions that serve as the foundation for the methodologies. Finally, to perform analyses with certain methods, measures are required. These measures can be *metrics*, which are quantifiable measures that are used to track and assess the status of a certain process. An example of a metric is the total processing time of a certain activity within a certain process. Other measures that are often used in the operational excellence field are *key performance indicators (KPIs)*, which are measurable values that demonstrate how effectively an organisation is achieving key business objectives. An example can be the number of new customers during the first quarter of the year. Based on these underlying concepts, some of the most important methodologies in the operational excellence field are discussed next. An elaborate discussion of the concept of a business process is provided in Chapter 3.

2.4 Lean management

The lean manufacturing system or Toyota Production System (TPS) was founded by Toyota in Japan in the 1950s, but was only labelled as lean manufacturing in the 1990s by Womack and Jones [182]. The lean philosophy focusses on **the reduction of waste and elements not adding value to the process [6]**. Lean management emphasises small batch sizes and make-to-order production systems. Unlike the traditional batch-and-queue production system, which results in large batch sizes, excess inventory, and long queue times between production steps, the objective of lean management is waste reduction through continuous improvement so only value creating activities remain. This way, defects can be discovered faster and easier and products are pulled by the customer, as they are only produced on customer demand, resulting in higher quality [6, 12]. An overview of the five basic principles for reducing waste that are generally acknowledged is shown in Figure 2.4 [35, 110, 154]. The five principles are:

1. **Understand the customer value.** Find which products or services and features of products and services add value and which can be identified as waste.

2. **Identify and analyse the value stream.** A value stream map is a process flow of all activities that contribute value to the product or service, which has been extended with data about speed, continuity of flow, and work in progress. All non-value-adding and unnecessary activities are removed from the process.

3. **Improve the value flow.** By reducing work in queue, batch processing, and transportation, the products or services can move through the system without interruptions.

4. **Let the customer pull value from the producer.** Products or services not demanded by customers should be avoided because they take up time, money, and resources at the wrong moment.

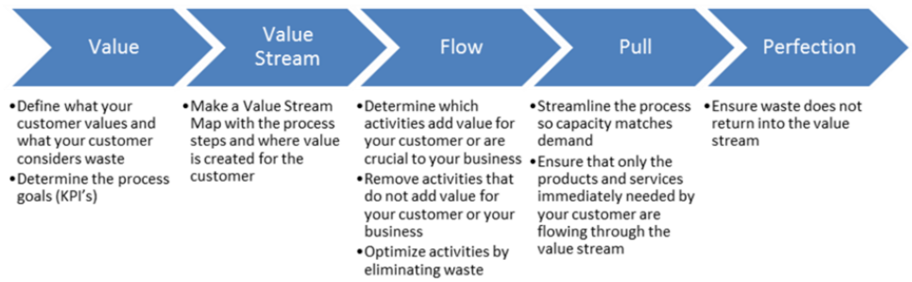

Figure 2.4: The five lean management principles of waste [154].

5. **Pursue perfection and continuous improvement.** Find ways to ensure that the efforts to remove waste can be repeated.

Improvements or benefits owing to the use of lean management are *operational* such as reduced lead times and inventory, and higher productivity and capacity; *administrative* such as less errors and more streamlined customer service; and *strategic* such as lower costs and higher customer satisfaction [6].

2.4.1 Definition of waste

Dahlgaard et al. [35] define waste as *everything that increases cost without adding value for the customer.* Two types of muda or waste can be distinguished according to van Assen [154]. The first type of muda should not be eliminated completely because it refers to non-value-adding activities that are essential to maintain company actions. An example can be a task that is added to the process to assure that the final product complies with all safety standards. The second type of muda is not only non-value-adding but it even destroys company value, such as errors or mistakes that occur throughout the production process. Moreover, van Assen [154] defines mura and muri as two other types of waste, which can be seen as drivers of the first type of waste. Mura is related to excessive variation and unevenness, which for example happens when the production process should rush at the end of the month to meet the targets, using up resources that are not actually required. Muri has to do with excessive loading of resources, causing them to have unnecessary stress, by, for example, providing them unclear instructions or a lack of proper equipment. The most common elements that can be classified as waste concerning to Pepper and Spedding [118] and Hines and Rich [69] are:

1. **Over-production.** This can lead to excessive lead and storage times which

cause defects that cannot be detected early, products that deteriorate, and pressure on the work rate.

2. **Waiting.** This refers to goods laying in the company or people doing nothing. Waiting time for workers may be used for training, maintenance or kaizen activities and should not result in overproduction. van Assen [154] adds the notion of the psychology of waiting, in which is stated that the actual waiting time is smaller than the perceived waiting time. For example, occupied time feels actually shorter than unoccupied time, just like in-process waits feel shorter than pre-process waits.

3. **Excess processing.** This refers to complex solutions that are implemented to solve simple problems.

4. **Delays.** This can be deliveries after due date or activities taking up more time than planned.

5. **Excessive transportation.** This may be double handling or excessive movements.

6. **Unnecessary inventory.** This leads to higher lead times and problems that are hidden in the abundance of inventory.

7. **Defects.** This is one of the most fundamental types of waste, as it is a primary cause of costs.

8. **Movement or unnecessary motion.** This refers to goods and people making more movements than necessary. Similar to excess processing this should be avoided.

Tsironis and Psychogios [151] summarise that the types of waste are composed of resources, which can be machines or people executing tasks, time, or money. Possible methods that are used for the elimination of waste are [6, 154]:

- Value stream analysis. A mapping tool to outline the production flow. Value stream mapping is further explained in Section 2.4.2.

- Total productive maintenance (TPM). An approach in which the resources such as machines or employees working on the equipment are also held responsible for the maintenance of the equipment. More up-time and less defects are a result of an improved environment.

- Kaizen (continuous improvement). A method to incrementally improve the process by combining the talents of the organisation.
- Document management. Best practices should be documented in the process in order to continuously work further on current improvement steps.
- Single Minute Exchange of Die (SMED). Different techniques that can be applied to lower set-up times, the number of set-up steps and unnecessary operations.
- Kanban (pull systems). Kanban cards were used as a signalling system before digital systems became available. The cards were put at different places in the production system to indicate that, e.g., the product or the raw materials were ready to be transferred to the next stage in a production process [183].
- Poka-yoke (error-proofing). Installing different points of error and defect detection in the process in order to prevent defects occurring at later (and more expensive) stages of the process.
- Visual control. Visualisations of the different steps and condition of different steps throughout the process will result in improved communication.
- 5S. A tool that is used to better organise a working area by using five S's: sort, set in order, shine, standardise, and sustain.
- Just-in-time (JIT) pull systems. In a just-in time production system, *"all processes produce the necessary parts at the necessary time and have on hand only the minimum stock necessary to hold the processes together"*. This way the number of products in stock and the throughput time of the process are brought to a minimum [145]. Related to the just-in-time theory is the *DRIFT-principle*, which is defined by Crosby [30]. According to this principle, quality is defined by the concept of "doing it right the first time". Each time an activity has to be executed more than once in the process, this can be defined as waste for the company.

As value stream mapping is one of the most widely known lean management methods, this approach will be introduced in the next section.

2.4.2 Value stream mapping (VSM)

Other than a supply chain which includes all activities executed in the company and its partners, a value stream only shows the activities or steps that add value to the end product or service of the company [69]. Seven categories of value stream mapping tools that are identified by Hines and Rich [69] are:

1. Process activity mapping. These tools eliminate activities that are unnecessary, simplifies or combines others, and seeks sequence changes that will reduce waste.
2. Supply chain response matrix. These tools provide an overview of critical lead time constraints for a particular process. Each of the individual lead times and inventory amounts can be targeted for improvement activity.
3. Production variety funnel. These tools help to decide where to target inventory reduction and making changes.
4. Quality filter mapping. These tools help identifying where defects are occurring and hence in identifying problems, inefficiencies, and wasted effort.
5. Demand amplification mapping. These tools map a product along its distribution and is used to show how demand changes along the supply chain in varying time buckets.
6. Decision point analysis. A decision point is the point at which products stop being produced according to the actual demand, and instead are made against forecasts only.
7. Physical structure mapping. The physical structure of the industry can be divided into a volume structure (number of companies) and a cost structure (where is value added?).

However, while the tools presented above have been proposed by different authors to be solutions on its own, Hines and Rich [69] state that a combination of tools should be implemented to be efficient. Therefore, in the VALSAT approach, which stands for value stream analysis tool, each tool and each type of waste gets a weight depending on its importance. Based on this classification, the most appropriate tool or combination of tools can be identified for each specific situation [69].

Next to the value stream mapping tools presented above, McManus and Millard [102] investigated value stream mapping more specifically in product development contexts. Based on this investigation, they provided the following overview of tools that are used when mapping the value stream:

- Gantt chart. A planning tool to define the sequence and dependency between tasks that should be executed.

- Ward/LEI map. Named after Alan Ward, this tool shows the resources that are necessary to execute all tasks over time.
- Process flow map. An actual flow between objects is depicted by arrows that connect different symbols. These process maps are then used to plot waste and non-value-adding aspects.
- Learning to see. A method introduced by Rother and Shook [131] that implies that companies should pick up a nature of detecting sources of waste.
- System dynamics. Similar to the previous tool, companies should learn to see the dynamics of the process structure and the underlying system instead of only looking at the individual components within the process.
- Design structure matrix (DSM). In this tool, a matrix is created to depict the flow between the tasks in a process. With this matrix, the company can minimise the number of repetitions in the process and can define the tasks that can possibly be executed in parallel.

However, similar to the value stream mapping tools presented by Hines and Rich [69], also here the conclusion was made that not one "best practice" exists and a combination of multiple tools, tailored to the specific circumstances of the case at hand, should be implemented.

Moreover, although value stream mapping combines different tools and techniques of the lean philosophy, **its paper-and-pencil approach has been criticised for not collecting enough detail and ignoring the actual system [118].** Data and information that is used to create the value stream map, or that is used as input for one of the other tools that are presented above, is almost always manually collected from one, some, or maybe all employees involved in the process. It is therefore very subjective and prone to errors due to aspects such as unawareness or forgetfulness. The artifacts presented in this dissertation aim to compensate these shortcoming because the focus will be on more quantitative analyses. Other elements of criticism of the lean management principles are summarised below.

2.4.3 Criticism of lean management

First, the approaches and principles presented above are not applicable when customer demand is volatile or unknown [6]. Lean always focuses on perfection in a particular situation at a specific moment, which makes it difficult to implement it in dynamic or changing situations, or in job-shop companies with high variety and low volume, because of the low flexibility and the inability to standardise the production approach [6, 118]. Other criticism can be attributed to a lack of understanding, direction, or commitment from the management. It should be mentioned that managers should view lean as a philosophy that entails a change in the entire company culture, and not concentrate on teaching their employees new tools and practices.

Moreover, as can be noted, **the tools presented above are rather qualitative**, as they are often based on data collected via interviews, question rounds, or estimates made by process owners. More objective input data and more qualitative analyses could be insightful enhancements to these. Also, instead of choosing one of these approaches, **a mix of lean tools should be applied** according to Pepper and Spedding [118]. And finally, according to Worley and Doolen [183], management support and communication play an important role in the implementation of lean management in companies.

In the next section, another philosophy, Six Sigma, is presented. Next to the fact that this methodology is more statistically underpinned than the principles and tools of lean management, it also takes into account more quantitative analyses.

2.5 Six Sigma

Six Sigma was founded by Motorola Corporation in the 1980s and focuses on **the elimination of variation to minimise defects and errors [93]**. It has evolved from a quality measure to a strategy to improve an entire business [8].

Six Sigma focuses on the reduction of variation by continuously and drastically improving everyday business activities so customer satisfaction is increased and waste and resources are reduced. The objective of Six Sigma is to only have 3.4 defects per million opportunities (DPMO) [6]. Defects can include anything, from missing or dysfunctioning components, to malfunctioning programming code, or numbers that have been wrongly entered by administrative workers [65]. Antony [8] states that instead of focusing on the defects in processes, Six Sigma tries to eliminate causes of problems before they transform into defects by focusing on the number of opportunities that can possibly lead to defects. For example, the waiting time before a service agent answers the phone or the way in which the agent talks to the customer are opportunities that might lead to defects and subsequently to unsatisfied customers in a service environment such as a call center [8]. An extended overview of literature on Six Sigma can be found in Fursule et al. [54].

Two similar improvement methods can be identified in Six Sigma. The first one, DMAIC, which is used for improving existing company processes, includes the five phases given in Figure 2.5 [6, 84, 154].

1. **Define.** In this step the process or product is identified and a cost-benefit analysis is created. After acceptance, a responsible is designated and the next steps can be performed.

2. **Measure.** Here the necessary data is extracted and collected. The problem is divided into different characteristics of the product or process critical to the customer's satisfaction and requirements.

3. **Analyse.** A diagnosis about the current product or process is made to define potential sources of variation for specific parameters.

4. **Improve.** The characteristics that should be improved are designated and changes or improvements are implemented.

5. **Control.** To maintain the improvements, the new conditions are documented and monitored via different control systems. Possibly, one or more of the preceding steps should be repeated.

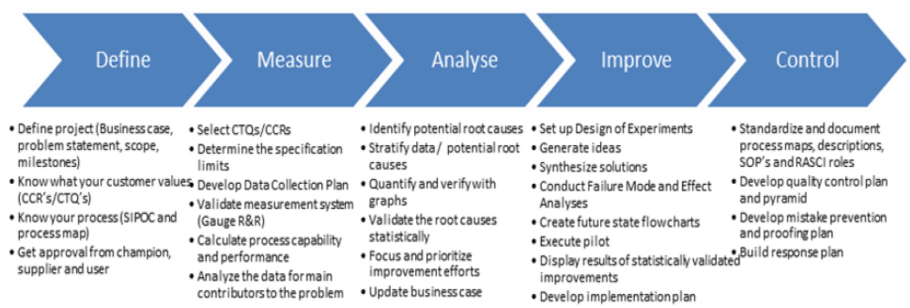

Figure 2.5: The five phases in the Six Sigma cycle [154].

The second method, Design for Six Sigma (DFSS), is used for the creation of new processes in case existing processes do not satisfy the customer needs. It can also be divided into five steps except for the replacement of the *improve* and *control* steps by *design* and *verify* (DMADV), respectively [1, 6]. Both the DMAIC-cycle and DMADV-cycle are grounded in the original "Plan, Do, Check, Act"-cycle which was introduced by Deming [43] [185].

Six Sigma employees use different practical methods and skills, defined as Six Sigma tools and techniques to improve the company's performance. Six Sigma tools such as pareto analysis, root cause analysis, process mapping or process flow chart, gantt chart, affinity diagrams, run charts, histograms, quality function deployment (QFD), kano model, or brainstorming. are narrow in focus and have a specific role. Six Sigma techniques have a wider application and require specific skills, creativity and training. Examples are statistical process control (SPC), process capability analysis, suppliers-input-process-output-customer (SIPOC), SERVQUAL, or benchmarking. A Six Sigma technique can use or can be supported by various tools [1, 8, 62]. Useful analytical tools are [72]:

- Flowcharts. Different types of flowcharts exist.
- Run charts. These tools depict trends in data over time and help to assess the importance of a problem.
- Pareto charts. These tools are based on the notion that a small percentage of causes results in a large percentage of problems.
- Check sheets or forms. These tools are mostly used for standardised data collection.
- Cause-and-effect diagrams or fishbone diagrams. These tools try to classify the elements that are causing the issues [47].
- Opportunity flow diagrams. These tools are used to distinguish between value-adding and non-value-adding activities.
- Process control charts. In these tools, plotted values are compared to an average and some control limits.

Failure mode and effect analysis (FMEA) and Design of experiments (DOE) are other techniques that are frequently used in Six Sigma projects. FMEA identifies, estimates, prioritises and evaluates risks of possible failures in the different phases of a process. DOE or multivariate testing determines the relationship between cause and effect variables [72]. An overview of which techniques and tools can be used in each step in the Six Sigma cycle is given by Yang [185] in Table 2.2.

It can be observed that some of the tools and techniques used in the steps of the Six Sigma cycle are also used in the value stream mapping method of lean management, such as the Gantt chart. Common tools for Six Sigma and lean management will also be discussed in Section 2.5.4 on Lean Six Sigma.

Table 2.2: Techniques and tools used in the Six Sigma cycle [185].

Step	**Specific tasks**	**Tools and techniques employed**
Define	Identify improvement issues	Customer complaint analysis
	Organise project teams	Cost of poor quality (COPQ)
	Set-up improvement goal	Brainstorming
	Estimate financial benefit	Run charts, control charts
		Benchmarking
Measure	Map process and identify inputs and outputs	Process map (SIPOC)
	Establish measurement system for inputs and outputs	Cause and effect matrix
	Understand the existing capability of process	Gauge R&R
		Control charts
		Process capability analysis
		Failure models and effects analysis (FMEA)
Analyse	Identify sources of variation in process	Cause-and-effect diagram
	Identify potential critical inputs	Pareto diagram
	Determine tools used in the improvement step	Scatter diagram
		Brainstorming
		Analysis of variance (ANOVA)
Improve	Conduct improvement actions	Design of experiment (DOE)
	Use experiments	Quality function deployment (QFD)
	Optimise critical inputs	Process capability analysis
		Control charts
Control	Standardise the process	Standard operation procedure
	Maintain critical inputs in the optimal area	Process capability analysis
	Verify long-term capability	Fool-proofing (Poka Yoke)
	Evaluate the results of improvement projects	Run charts

Antony [8] defines a list of measures or KPIs that are used in Six Sigma across service companies. These measures can be, among others, cost of poor quality (COPQ), defects per million opportunities (DPMO), process capability, time to respond to customer complaints, processing time, delivery time or speed of delivery, time to restore customer complaints, waiting time to obtain the service, service reliability, or accuracy of information provided to customers.

Comparable to lean management, which is a rather qualitative methodology and influenced greatly by elements such as the company culture, organisation of work, and management commitment, Six Sigma also makes use of qualitative techniques. Six Sigma is based on training programs, in which different levels of belts can be earned per training level, and it is stated that the training programs are also based on qualitative techniques. However, next to this, Six Sigma is merely a statistical methodology, in which companies strive to have less than 3.4 defects per million opportunities. Therefore, it is more data-driven and makes use of more advanced data analysis tools than other operational excellence initiatives such as lean management and TQM [87], which will be explained in Section 2.5.3. First, a discussion of the concept variability is provided in the next section.

2.5.1 Variability

van Assen [154] defined variability in the context of operational excellence as *"anything that causes a production and delivery system to deviate from its regular behaviour."* Product variety, breakdowns, set-ups, product recycle, material shortages, unavailability of resources, and rework can be seen as possible sources of variability or variation in a production company. Service companies on the other hand, also encounter customer-introduced variability, such as *arrival variability* which is caused by customers that prefer to receive a service at different times or at times that are inconvenient for the company; *request variability* which refers to the extensiveness of products that customers ask for in a service context; *capability variability* which is mostly important if customers with different amounts of knowledge, abilities, and skills need to participate actively in the business process of the company; *effort variability* which depends on the amount of effort that customers want to spend in a service process; and *subjective preference variability* which refers to the varying impressions of customers concerning the service they receive from a company [53].

However, not all variability is undesirable. That is why Suri [146] stated that two types of variability can be distinguished: *functional variability* and *dysfunctional variability*. Dysfunctional variability, which should be eliminated completely, is caused by errors, ineffective systems, and poor organisation. Functional variability on the other hand, is used by companies to maintain a competitive edge in the market. If dysfunctional variability cannot be reduced or adapted, long cycle times, high work-in-progress levels, wasted capacity, lost throughput, and unsatisfied customers can be identified as consequences [154].

2.5.2 Six Sigma in service companies

According to Tsironis and Psychogios [151], most studies found in literature focus on the application of lean management and six sigma in manufacturing environments. Even though all executed work can be seen as part of a process that possibly includes variability, and each process produces data that can possibly explain this variability, service-oriented companies are not yet convinced of the use of Six Sigma within their organisation [8]. Therefore, Tsironis and Psychogios [151] focus on the results and success of applications of these methodologies in service companies. In this dissertation, the focus will also be on service environments, which are much more volatile and influenced by factors such as human interaction and company culture [151].

Antony [8] stated that most projects in service-oriented companies are selected based on subjective judgement. Six Sigma can import a more objective measure to service companies provided that different weights are assigned to different defects based on their risk and consequences. A structured overview of literature concerning the application of Six Sigma in services is provided by Antony [8] and Johannsen et al. [79]. Similar to manufacturing contexts, commonly used indicators for Six Sigma are, among others, processing time, delivery time or speed, waiting time, defects per million opportunities, cost of poor quality, and service reliability [8]. An overview of critical success factors that need to be in order for Six Sigma to be successful in service companies can be found in Johannsen et al. [79]. In addition to this, they investigated the most common problems in the application of Six Sigma in service companies and identified the stages in the Six Sigma implementation cycle to which these problems could be assigned. This was done based on a literature review and expert interviews. They found that most problems are identified in the *define* and *measure* phases.

First, it was found that processes are usually not well documented. Another problem appearing in the *measure* phase is the lack of process orientation and data of high quality [1]. 79 % of the experts contacted by Johannsen et al. [79] states that the process orientation is mostly of inferior quality. However, no problems with process modelling are reported by the experts in the *define* phase, what might indicate that these problems are ignored until the *measure* phase. Therefore, it can be stated that **proper process modelling should start in the *define* step** to avoid difficulties in defining key performance indicators. Furthermore, as different types of problems with data collection and data quality where reported in the *measure* phase, this was given the most attention in the analysis of Johannsen et al. [79] as the quality of all **process performance analyses depend on the quality of the data**. Flow charts, for example, are created by collecting input manually from all employees involved in the process. Or in the FMEA tool, the different steps, together with possible failures and their degree of severity, are collected based on human estimates and experiences via techniques such as interviews. It can therefore be stated that both the data collection phase and the measure phase in which these data are used should be enhanced with more data-driven techniques and adequate measurement systems [79].

2.5.3 Six Sigma within Total Quality Management (TQM)

Klefsjö et al. [83] position Six Sigma as a methodology within the larger framework Total Quality Management (TQM). TQM, mostly known for its improvement cycle (plan-do-check or study-act), is a continuously evolving management system characterised by increased customer satisfaction in which all employees participate. Näslund [109] found that differences other than the period in time between TQM and Six Sigma are difficult to find. TQM was popular in the late 1980s and early 1990s, whereas literature about Six Sigma started to grow rapidly in the late 1990s. However, Aboelmaged [1] and Schroeder et al. [140] identify Six Sigma as having some advantages over TQM, such as focus on financial and business results, use of a structured method for process improvement or new product introduction, use of specific metrics such as DPMO, and a significant number of full-time improvement specialists. Also Antony [7] describes the differences between Six Sigma and older existing quality programmes. Firstly, he states that the focus in Six Sigma on bottom-line results is not present in previous quality methodologies. Secondly, Six Sigma places importance on repeatability and reproducibility of the systems to measure the business excellence. Thirdly, previous quality initiatives placed little emphasis on leadership and management support, as well as human elements such as teamwork, cultural change and leadership skills. Finally, Six Sigma uses a unique system of belts and utilises tools and techniques in a sequential and systematic manner. Aside from the fact that Six Sigma finds its roots in TQM, also similarities with the just-in-time (JIT) and the total productive maintenance (TPM) approaches from the lean management methodology can be found in literature, according to Cua et al. [32], who provide a discussion on the relationship between these approaches. The authors argue that the approaches are often used in a combined way in practice and have the same goal and many common practices. Organisations can therefore benefit from the joint use of these approaches, instead of implementing them separately. The most well-known example of a combination of multiple methodologies is Lean Six Sigma, which is discussed in the next section.

2.5.4 Lean Six Sigma (LSS)

By combining the strengths of the two improvement methodologies lean management and Six Sigma, Lean Six Sigma has been introduced, as companies should not only sell high quality goods but should also provide a high quality of service. By implementing Lean Six Sigma, the company focusses on what is really important to the customer and the reduction of errors.

Six Sigma companies can thus gain from lean management by reducing set-up times and mapping the value stream. Otherwise, the improvements made will start to lose influence after a while because no time is spent on altering the underlying operating systems to reduce activities that create waste. Vice versa, lean management companies should use more methods that promote quality in a more scientific way to benefit from continuous improvement [12]. One argues that lean and Six Sigma complement each other; lean eliminates waste and reduces cycle time in processes by eliminating non-value-adding time and increasing value-adding time while Six Sigma reduces variation and improves processes by applying a problem solving approach using statistical tools [1, 154].

In Figure 2.6 some common tools for lean management and Six Sigma have been identified based on the findings of Drohomeretski et al. [46] and Kumar et al. [86] [9, 118, 136]. A LSS company profits from 3 lean management elements. These are:

1. an overarching philosophy that enhances the value-adding function of all operations,
2. a continuous evaluation system to ensure global instead of local optimisation, and
3. a decision process that takes into account the relative impact on each customer.

Next to this, a LSS company profits from 3 Six Sigma elements [12], which are:

1. scientifically supported decisions,
2. a variation-minimising methodology, and
3. a company-wide education and training system.

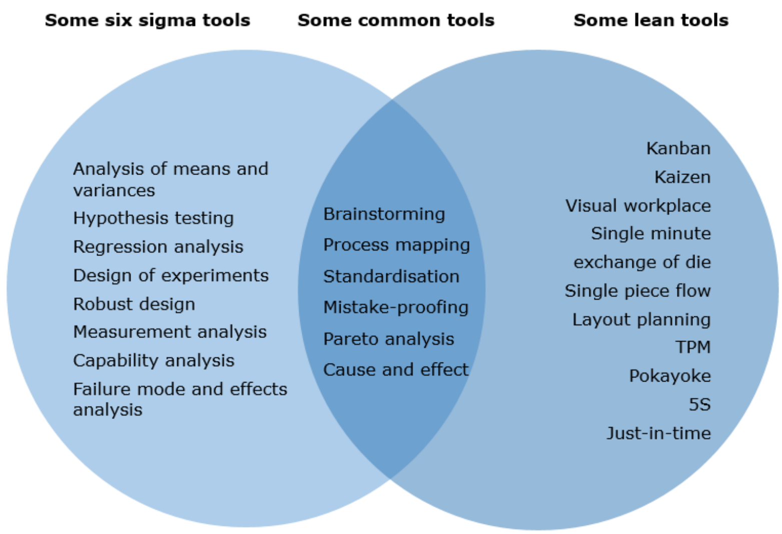

Figure 2.6: Common tools for Six Sigma and lean management [46, 86].

The scope of the two approaches can thus be defined as different. While Six Sigma is more concentrated on a specific project or process inside the organisation, lean focusses on the value chain of the whole organisation [1].

Koning et al. [84] state that an integrated LSS framework requires a structured approach, a project-based deployment, a dedicated workforce of project leaders, clearly defined procedures, and strategic objectives used as a basis for project selection. Because of this multiplicity of requirements, most of the Lean Six Sigma initiatives have not been successfully realised [118]. Another reason for this is defined by Näslund [109] who states that most companies try to change or improve their processes in a functional, operational, or ad hoc manner, while what is actually required is fundamental organisational change and improvement, which can only be accomplished by a more process oriented approach.

Despite the benefits of using Six Sigma tools in combination with lean management, a comprehensive framework or a clear understanding concerning the usage of tools is missing. Pepper and Spedding [118] suggest a framework in which lean philosophy provides the strategic foundation for improvement. From this, key areas for improvement can be identified and anticipated by a Six Sigma methodology. It can thus be stated that lean contributes the strategy or structure and Six Sigma provides the tools to leverage an improvement [46, 118]. The improvement objectives of Lean Six Sigma are summarised in Figure 2.7. An application of Lean Six Sigma on different case studies in service companies is given by Tsironis and Psychogios [151].

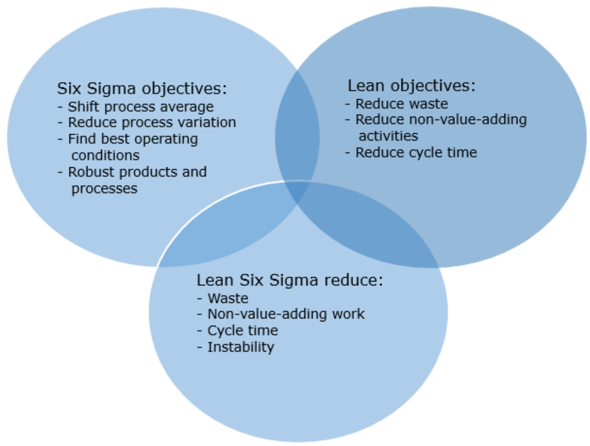

Figure 2.7: LSS improvement objectives [46].

A final philosophy that can be found in literature to be related to operational excellence is the theory of constraints (TOC), which will be explained in the next section.

2.6 Theory of constraints (TOC)

In 1984, Goldratt and Cox [60] introduced the theory of constraints, a management philosophy which focuses on system improvement by eliminating weak links or constraints from the processes in a system because *"a process is only as strong as its weakest link"*. By doing this, waste and variation can be reduced resulting in improved throughput, throughput times, and quality. Five steps can be identified in TOC [110]:

1. **Identify the constraint.** Possible methods can indicate the amount of queue or batch processes.

2. **Exploit the constraint.** The process should be enhanced or corrected without major adjustments.

3. **Subordinate other processes to the constraint.** Other processes that are subordinate to the process are usually found ahead of the constraint in the value stream. Their speed or capacity should be paced to the constraint.

4. **Elevate the constraint.** Changes to the constraint can be necessary if the output of the system is not sufficient. This can be reorganisations, expenditures

of time or money, or capital improvements to eliminate the constraint as much as possible.

5. **Repeat the cycle.** Another constraint will appear after eliminating the first constraint. A re-evaluation of the system is necessary to identify and exploit the new constraint followed by subordination and elevation.

Based on the theory of constraints is the drum-buffer-rope (DBR) methodology, which is also developed by Goldratt and Cox [60]. In a manufacturing context, the *drum*, which is the constraint or the bottleneck in the process, determines the pace of the products that move through the system. The *buffer* represents the time between the faster moving products in front and the constraint or slowest product. This buffer is used to protect the constraint and the system from disruptions such as breakdowns. The *rope* makes sure that all products move through the system at the pace of the drum [137, 139].

Based on an analysis of over 80 applications of TOC found in literature, Mabin and Balderstone [92] conclude that over half of the organisations in the analysis noticed improvements in revenues or profits, and even more than 80 % of the organisations reported improvements in key performance indicators such as lead time and inventory. However, similar to the methodologies of lean management and six sigma, also the tools and techniques of TOC are almost always fuelled with manually collected data, based on experiences and subjective interpretations of reality. The lack of data analysis is therefore stated as a point of criticism on the TOC methodology by Nave [110].

2.7 Similarities and differences between lean management, Six Sigma, TQM and TOC

Lean management, Six Sigma and TQM originated all from the quality evolution in Japan, but developed differently. TQM became popular in the 1990s but did not originate from a company like lean and Six Sigma did. TQM also has elements of accomplishing no defects (such as Six Sigma) and eliminating waste (such as Lean), so Lean and Six Sigma should be seen as methodologies within the larger framework of TQM.

The five phases of Six Sigma seem easier to implement than the principles of lean production [35]. Moreover, in comparison to the Six Sigma DMAIC-phases, the lean principles are not cyclical [6].

Unlike TQM, which stresses the involvement of all employees, project groups are usually committed to perform improvements in the Lean or Six Sigma discipline [6].

The goal of lean management is reducing the lead time and increasing the customer satisfaction. This leads to increased productivity and inventory reduction. Six Sigma projects however, are selected in the context of the overall organisational objectives, which means that the customer satisfaction is not always improved. Furthermore, because a Six Sigma project is mostly undertaken by only a part of the organisation, other departments can experience deterioration. Finally, Andersson et al. [6] state that lean is mostly applicable to manufacturing areas, whereas Six Sigma also applies to service industries. However, Arnheiter and Maleyeff [12] believe that lean management can be practised in all kinds of businesses that try to satisfy customers. An overview of the most important differences between Lean management, Six Sigma, TQM and TOC is created in Table 2.3, which is based on the comparison made by Nave [110] who compared lean thinking, Six Sigma and the theory of constraints.

Table 2.3: Comparison of lean management, Six Sigma, TQM, and TOC.

	Lean management	**Six Sigma**	**TQM**	**TOC**
Theory	Remove waste	Reduce variation	Focus on customers	Manage constraints
Focus	Flow focused	Problem focused	Customer focus	System constraints
Methods	1. Identify value	1. Define	1. Plan	1. Identify constraint
	2. Identify value stream	2. Measure	2. Do	2. Exploit constraint
	3. Flow	3. Analyse	3. Check	3. Subordinate processes
	4. Pull	4. Improve	4. Act	4. Elevate constraint
	5. Perfection	5. Control		5. Repeat Cycle
Primary effects	Reduced flow time	Uniform process output/save money	Increased customer satisfaction	Fast throughput

Table continued on the next page

Table 2.3: Comparison of lean management, Six Sigma, TQM, and TOC (continued).

	Lean management	**Six Sigma**	**TQM**	**TOC**
Secondary effects	Less variation	Less waste	Customer loyalty	Less inventory/waste
	Uniform output	Fast throughput	Improved performance	Throughput cost accounting
	Less inventory	Less inventory		Throughput-performance measurement system
	New accounting system	Fluctuation-performance measure for managers		Improved quality
	Low-performance measure for managers	Improved quality		
	Improved quality			
Criticism	Statistical or system analysis not valued	System interaction not considered	No tangible improvements	Minimal worker input
	Not applicable in all industries	Does not involve everybody	Resource demanding	Data analysis not valued
	Reduced flexibility	Processes improved independently		
Data input	Interviews, human-based experiences and opinions, paper and pencil	Interviews, human-based experiences and opinions, paper and pencil, statistics, manual and automatic data collection	Interviews, human-based experiences and opinions, paper and pencil, manual and automatic data collection	paper and pencil, manual and automatic data collection

For the data that is used as input for the analyses performed within the different methodologies, a joint conclusion on all methodologies can be made. It is clear from literature that many techniques and tools that have been presented above are based too often on human-based experiences and opinions, which are collected by interviews, interrogations, and discussions, implying a high risk of subjectivity and incompleteness. Moreover, analyses that are performed are too often based on paper-and-pencil approaches such as the manual mapping of a value stream or the unsubstantiated estimation of the error and defects impact in Six Sigma. Although more recently more data-driven approaches are employed, to improve the quality and correctness of the input values of the methods, little or no existing research mentions the specific requirements or characteristics of the data which are required to perform the analyses, or the steps that should be undertaken to collect and interpret these data.

2.8 Underlying principles of operational excellence

Based on the presented overview and comparison between the different philosophies, the main principles found in the field of operational excellence can be summarised as follows. Firstly, lean management is focused on identifying the activities in a process that create **value for the end customer** and the **removal of all activities that can be defined as waste**. This way, the flow of the process is converted to a make-to-order production system with small batch sizes and continuous improvement is pursued. Secondly, Six Sigma is a philosophy that focuses on the **avoidance and reduction of (some sorts of) variation or variability in the process flow** by defining and removing the elements that could cause problems. It is also more based on statistics than the lean management philosophy. While Six Sigma can be seen as related to the philosophy of total quality management, the latter is much more focused on customer value and customer satisfaction. Theory of constraints, finally, focuses on the **constraints and weak links that hamper or slow down the process flow**.

Based on these philosophies, there is not one single strategy or roadmap that companies should follow in order to improve their processes. Depending on the focus they want to put on their process improvement, they should pursue different things or they should try to incorporate multiple methodologies. If customer value and customer satisfaction are the highest priority, the principles of TQM should be pursued. However, the question can be asked if the company wants to serve their customers as fast or as cheap as possible, which can be an indication that the tools of lean management should be used. Or if the company wants to serve their customers with as less defects and errors as possible, in which case, the principles and tools of Six Sigma should be used. The TOC methodology focuses, comparable to lean management, on the constraints and activities hampering the optimal flow of the activities in the process. Combinations of these methodologies, such as Lean Six Sigma have already been researched and found to be effective.

2.9 Conclusion

This chapter provides an introduction to the evolution of quality management and some well-known improvement concepts such as lean management, Six Sigma, the theory of constraints, and total quality management. Based on a comparison between these philosophies, some conclusions can be drawn based on the underlying principles of operational excellence. First, existing methods such as value stream mapping make use of lean management techniques that are often based on qualitative paper-and-pencil approaches. This has been argued to be ignoring the actual system and not collecting enough detail. The principles of lean management are also identified to be less useful in volatile or fast changing environments. Six Sigma tools, which are more statistically underpinned, are in some cases based on quantitative data, while others are also based on qualitative methods such as interviews. The results of these methods and the decisions taken based on these results may therefore be very subjective and dependent on the person or team that has been performing the analyses. It can therefore be stated that not one single strategy or roadmap exists for companies to follow in order to improve their processes. A combination of multiple methodologies should be incorporated and a need for more objective and data-based analyses to gain more insights in the operational excellence in companies can be identified.

Chapter 3

The use of process mining in operational excellence

3.1 Introduction

As was shown in Chapter 2, waste and other activities not adding value to the customer of a process should be removed. However, removing these undesirable elements can only be done if one has insight in the business process and all related aspects. Business process management (BPM) is a discipline that focuses on how work is performed in an organisation by managing the organisation's processes. It inherits from the total quality management stream, which was introduced in Chapter 2, in order to optimally align an organisation's processes with its performance objectives. In contrast to these mainly model-based improvement disciplines, process mining refers to the retrieval of knowledge from process execution data, which is stored in so-called event logs. The process mining field originated at the end of the previous century in response to the digitalisation, which implies that more and more data is available to get insights into the performance and operational excellence of business processes. As process mining is a rather new concept within the field of business process management, its approach can be an additional application in operational excellence, to find the different forms of waste of lean management or to implement the different steps in the DMAIC-cycle of Six Sigma.

This chapter (Figure 3.1) provides a short introduction to the fields of business process management and process mining and an outline of the match between process mining and the philosophies and principles of operational excellence from Chapter 2.

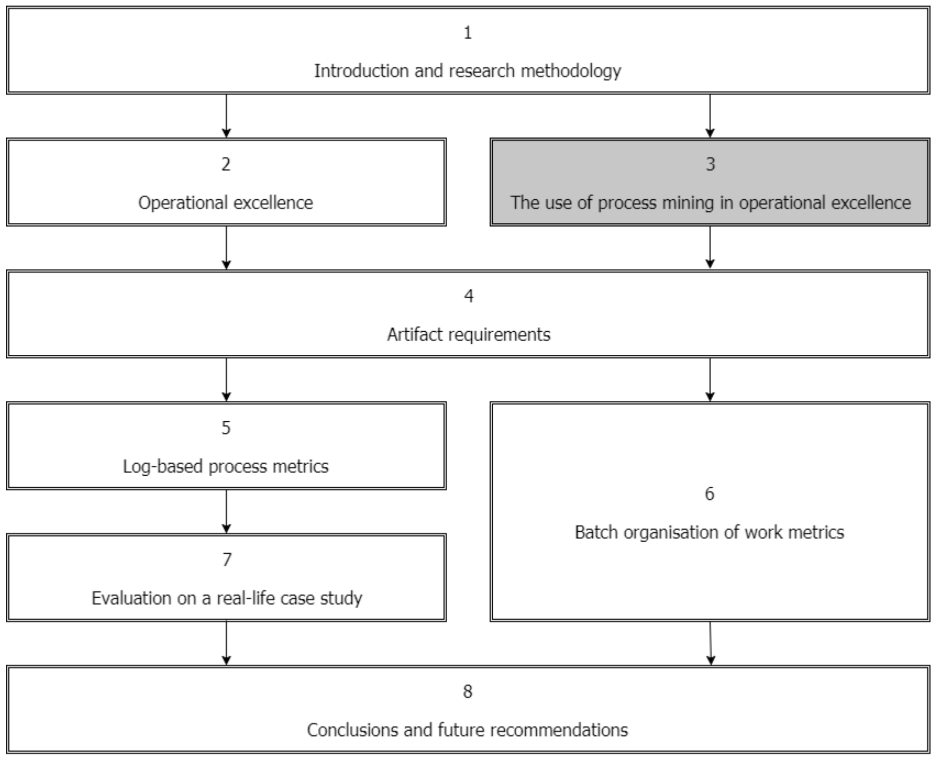

Figure 3.1: Outline of the thesis - Chapter 3.

Section 3.2 starts with an introduction to business process management, including the ingredients of a business process, the relationship with the operational excellence and process improvement concepts introduced in Chapter 2, and the BPM lifecycle. Next, Section 3.3 introduces the field of process mining including the structure of event logs and the different types of process mining. Moreover, the difference between model-based and event log-based performance analysis is explained. Next, Section 3.4 elaborates upon the existing work in the interplay of process mining and operational excellence including an overview of some existing process performance measures from prior literature and a discussion of the added value of process mining in the field of operational excellence. Finally, conclusions are drawn in Section 3.5.

3.2 Business process management (BPM)

Business process management (BPM) is concerned with a wide range of topics such as operations management, business process intelligence and analysis, process automation, and the organisation of work [156]. The main goals of BPM are the improvement of business processes and the optimisation of customer value [28, 47]. As is clear from these descriptions, the focal point of BPM are business processes. A business process entails what companies actually do, which is delivering products or services to their end customers. The quality of these products and services is therefore highly dependent on the way processes are designed and managed [10, 47]. Before going deeper into detail about the field of BPM, the general concepts of a business process are presented in the next section.

3.2.1 Ingredients of a business process

As defined by Anupindi et al. [10], a business process is *"a network of activities performed by resources that transform inputs into outputs"*. The transformation that takes place depends on the process architecture, or structure, which includes some elements that identify the business process. Firstly, a business process consists of activities and events. *Activities* are the tasks or items of work that need to be performed within the process, in a certain order. Each activity can take some time to be completed, such as the inspection of a certain part of equipment or the preparation of an invoice for a customer. *Events* are the atomic things that happen for an activity, which have no duration, such as the arrival of an equipment part or the completion of an invoice. Secondly, a process also contains *decision points*, which may influence the way the process is executed. This can, for example, be the decision that the information required to complete an invoice is insufficient and another activity should be executed first to collect this missing information. Thirdly, *actors* or resources that are involved in the process should also be considered, as the process depends on these human factors or systems to be executed correctly. Next to these resources, also other objects, both physical (materials, documents) and immaterial (electronic documents or records) are involved. Fourthly, the result of the process execution is the *outcome* of the process, which can be both positive (value-adding) or negative (non-value-adding). And finally, the outcome of the process is intended for the *customer* of the process, which can be an employee of another department of the organisation or an external end-customer [47]. Anupindi et al. [10] adds to these concepts the notion of an *information structure* as another element of a business process. The information structure contains all information that is required to take decisions or to execute activities within the process. Based on this overview of the most important concepts of a business process, Dumas et al. [47] defines BPM as *"a body of methods, techniques, and tools to discover, analyse, redesign, execute, and* *monitor business processes"*. From this definition, it can be concluded that BPM is involved in multiple stages and tasks in the lifecycle of a business process, as will be explained further on, where the BPM lifecycle is presented.

3.2.2 BPM in total quality management (TQM)

As is clear from the description of BPM in the previous section, the field of business process management is strongly related to the other operational excellence concepts such as lean management, Six Sigma and total quality management, which were introduced in Chapter 2. BPM is, among others, concerned with the improvement of the performance of organisations and their processes. The differences and relation between BPM and the other operational excellence fields is also described by Dumas et al. [47]. TQM, for example, also entails continuous improvement efforts, but the focus of TQM is more on the products and services that are delivered to the end customer, while BPM puts its emphasis on the improvement of the processes delivering these products and services. BPM can be seen as a modern continuation of these fields as it inherits from the principles and techniques of TQM, lean, and Six Sigma and it enriches them with the capabilities of modern information technology. The goal of BPM is to bring all processes of an organisation in line with its performance objectives, in order to become operationally excellent [47].

3.2.3 Business process improvement (BPI)

One of the underlying approaches of BPM is business process improvement (BPI), which is closely related to the operational excellence field as it is also concerned with improving business processes. These improvements are based on changing the processes to achieve a higher quality or to become cheaper, faster, or more flexible. BPI can be distinguished from the earlier wave business process redesign (BPR) depending on the degree of improvement; while BPI is synonymous to incremental improvement, BPR focuses on more radical changes. Moreover, BPR was only concerned with the planning and organisation of a process, while BPI and BPM in its whole provide a total package of tools, concepts, and techniques that cover all aspects of managing a process, from planning, organising, monitoring, and controlling, to the actual execution of the process. This comprehensive view is made clear by Dumas et al. [47] in Figure 3.2, in which the different aspects of BPM are shown. A structured overview of BPI literature is provided by Zellner [186] together with an evaluation of his findings.

Adesola and Baines [2] developed a seven-step procedural approach that can be used as a guide for both process improvement and re-engineering initiatives. They

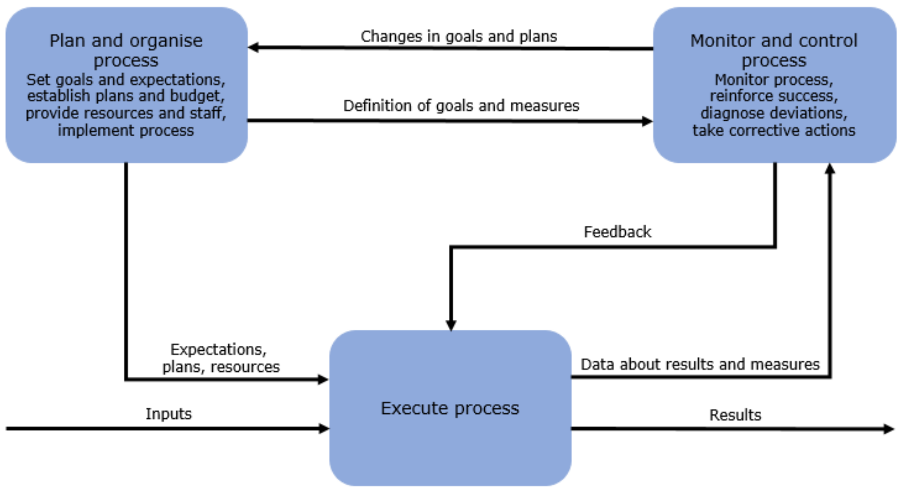

Figure 3.2: BPM: an all encompassing discipline [47].

used older frameworks as a basis for their model-based and integrated process improvement (MIPI) methodology. It will be shown that this methodology is very similar to the BPM lifecycle that will be presented in the next section. The method, which is shown in Figure 3.3, addresses what should be done in each step to improve a process, and how to do it. Firstly, the business need should be analysed by developing a vision, strategic objectives, and an organisational model based on the current practices. An analysis of the competition can be performed and objectives should be prioritised. Furthermore, measurable targets should be established and the process objectives should be developed and benchmarked. Next, the process should be clear for everyone involved, by capturing the as-is process information and identifying the process architecture. The scope and definition of the process can be concretised by modelling the process. Next, the process should be analysed by verifying and validating the model and measuring the existing process performance. After that, the process can be benchmarked to identify performance criteria for redesigning the process. A new to-be process model can be developed and validated and the performance of the redesigned process should be estimated. Next, a plan can be presented to implement the new process. A change management plan should be developed which includes the communication of the change and the training of the staff. The process should be made operational and the changes can be rolled out. In the sixth step, process deployments and performance data reflections can be conducted to assess the new process and methodology. And finally, the strategic view of the business should be developed and process targets and performance analyses can be set to review the new process. A plan to meet these process targets can be developed and implemented.

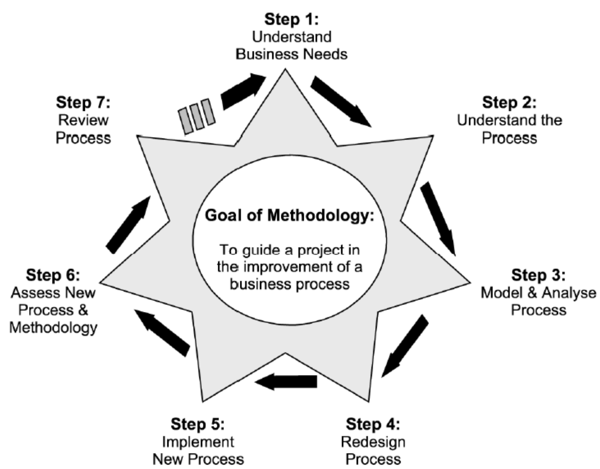

Figure 3.3: The model-based and integrated process improvement (MIPI) methodology, developed by Adesola and Baines [2].

Griesberger et al. [62] analysed 487 existing techniques that aid in the different steps of business process improvement. After filtering less relevant techniques, 36 techniques that actually support the act of improvement are analysed. However, they found that for none of the existing techniques it is clear *how* they support the act of improvement. Also Zellner [186] states that **most approaches lack specific guidelines for actual optimisation of business processes**.

Based on his literature overview on BPI, Zellner [186] also states that there are still some shortcomings to the existing BPI approaches. Firstly, the criteria of what improvement exactly is and how it can be achieved is not specified in any of the existing BPI approaches. Well-defined activities that should be executed to achieve improvement are missing. Next to this, the results following from activities in BPI, such as resulting documents and reports, should be modelled to be able to analyse the relations to each other. Furthermore, an overview of all existing techniques supporting BPI and a new encompassing method to support all steps in BPI should be created. Next to these shortcomings, Satyal et al. [138] recently stated that BPI initiatives do not always lead to actual improvements, and sometimes even result in negative outcomes. This is mainly due to the fact that most information systems are not able to evaluate and monitor the implemented changes for improvement. The authors therefore present a new technique that delivers immediate feedback on the improvement results, based on AB testing, to compare the improved business process with the original one [138].

3.2.4 The BPM lifecycle

The BPM lifecycle, which is shown in Figure 3.4, provides an overview of how business processes can be managed in the light of the BPM discipline. The lifecycle starts with the process identification, which can be a very short step if the company already has a good notion of the processes at hand. Also the process performance measures, to identify the *quality* of the processes, should be identified before starting the next step. After this, five phases follow, of which the first one is the (i) process discovery, where the details of the process are uncovered. When the as-is process model is clear, a (ii) process analysis phase takes place to get an insight in what is going on in the process concerning the weaknesses and their impact, such as the amount of rework or the waiting time in the process. Based on these findings, possible alternatives for solving these issues can be presented in the (iii) process redesign phase. Next, the proposed to-be process model is used as a solution based on which the actual (iv) process implementation follows, involving steps such as organisational change management and process automation to put the changes into action. Finally, the running process should be (v) monitored and controlled continuously in order to keep up with changing expectations and situations [47]. Within the BPM lifecyle, two different types of analysis can be defined, which are (i) model-based analysis, based on the developed process model, and (ii) data-based analysis, based on the underlying event data that has been tracked in the event log [156]. It can be stated that traditionally almost all process management work was model-based, yet because of the growing availability of data, the use of data-based analyses increases. In the process discovery phase and the process analysis phase, for example, event log data can be used to analyse the running processes, and to discover unusual elements such as deviations and waste, which can be useful information for the process redesign phase. Analyses can also occur based on process models, e.g., to simulate solutions for these deviating elements. Also monitoring and controlling the process occurs mostly based on process models. Although process mining is useful in each of the phases of the BPM lifecycle, most work in the process mining field can be positioned within the process discovery phase and the process monitoring and controlling phase [160], as will be shown later on in this chapter. Regarding this BPM lifecycle, the research in this dissertation can be positioned between the as-is situation and the to-be situation

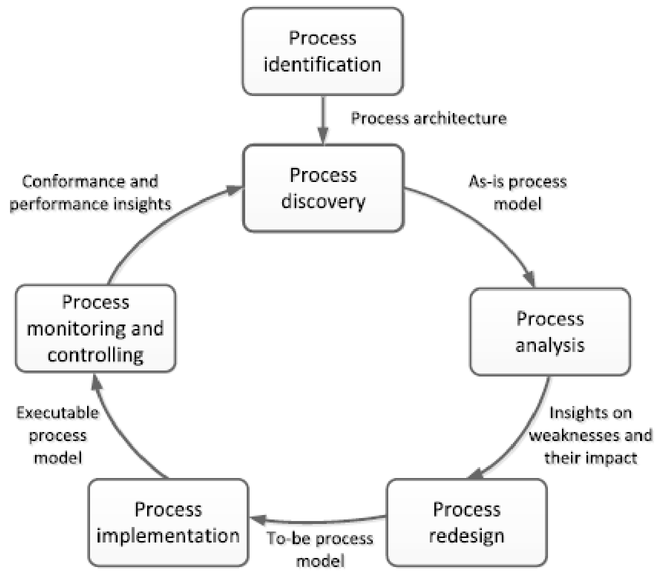

Figure 3.4: The BPM lifecycle as presented by Dumas et al. [47].

of a business process. Event log knowledge will be used to get insights into the current circumstances and status of a business process in order to support the organisation to become more operationally excellent.

3.3 Process mining

Concerning the well-known big data phenomenon, more and more process-related data is recorded and tracked by companies, such as logs of production processes or execution traces of business activities. To respond to this abundance of data, a need for methods and techniques to analyse these data, to gain insight into business processes, to properly track and store activities, and to introduce improvements to the way people and companies are working has grown. With this aim, the process mining field has originated at the end of the previous century to *extract knowledge and the control-flow from process-related datasets, or so-called event logs* [155, 169]. In contrast to the operational excellence techniques presented in Chapter 2 and the BPM field presented above, which are mainly based on models, simplifying assumptions, and human-based information, process mining puts more focus on the actual numbers and data that are collected within the underlying event logs.

From these observations, it follows that event logs are fundamental to process mining. An event log originates from a process-aware information system (PAIS), which is an information system that supports or controls a real-life business process. Specific for a PAIS is that it supports an entire business process instead of a single activity [48]. Examples of such systems are an enterprise resource planning system or a customer relationship management system [160]. The information extraction from these event logs can be done from various perspectives, such as control flow, organisational structures, performance characteristics, or the case data perspective [132, 169]. While process mining has grown much broader than control flow discovery, the latter has dominated the domain for the past decade [37, 52, 57, 63, 164, 175]. Control flow discovery is one of the most mature research tracks within process mining and the reader is referred to De Weerdt et al. [42] for an overview of existing process discovery algorithms. Moreover, other types of process mining, such as conformance checking, bottleneck analysis, or process enhancement become only possible after the discovery of a process model [162].

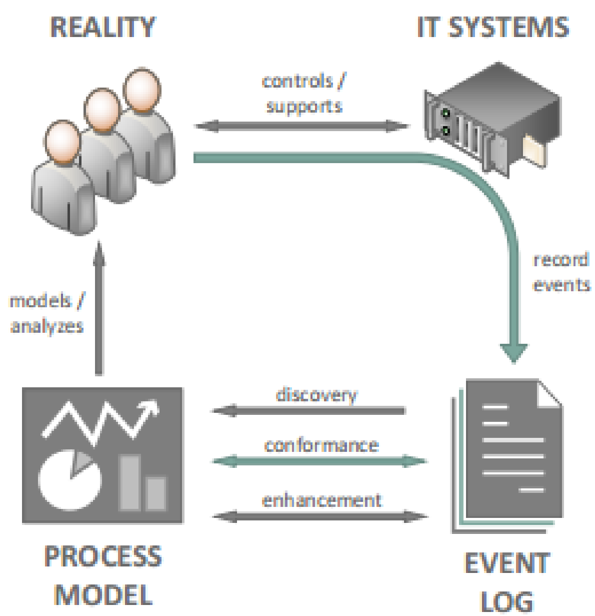

Figure 3.5: Positioning of process mining by Munoz-Gama [106].

In Figure 3.5, a schematic overview of process mining in its bigger context is given. Activities and decisions taken by companies and people are supported or controlled by a variety of process-aware information systems and can be recorded in event logs. From these event logs, process mining models can be discovered that can be used to model, analyse, and improve the process. Visualisation tools such as Disco[1] or the plug-in based framework ProM[2] contain numerous functionalities to support the different types of process mining which will be discussed further on.

The remainder of this section provides some background on the typical structure of an event log (Section 3.3.1) and the different types of process mining that can be distinguished (Section 3.3.2).

3.3.1 Event log

Ideally, process-aware information systems record data for each task that has been executed within a process. The system checks each time when a task is completed, and keeps this as a new entry in the log file. The data that is stored in such files are called event logs, such as transaction logs or audit trails, which can be seen as the starting point for performing a process mining task. Different kinds of event logs can be used to apply process mining techniques to, but some requirements, which distinguish a process-based event log from traditional flat datasets, should always be present for the actual events or entries in the log [47]. vanden Broucke [169] states that firstly, each entry or event in the log should be related to one process case, which can be realised with a case identifier or an instance identifier. Secondly, all events should be labelled with a suitable name that represents the activity, a well-defined step in the process, that has been performed. These names or labels can then be used to make the discovered model self-explanatory and readable for anyone at interest. Finally, a timestamp should be present for each entry in order to discover and display the sequence of events and instances. Also Dumas et al. [47] state that these are three minimal requirements for events in an event log in order to perform certain analyses. If timestamps are not available, the events can be ordered based on relative ordering, for example with a sequence number. However, timestamps add a lot of extra information and should be tracked as accurately as possible. Next to these requirements, a wide range of additional attributes can become desirable, depending on the goal that is set [47, 169]. An example of additional information is the person or device executing or initiating the activity, which is referred to as the resource throughout this dissertation. Other examples are event-related costs, or the system being used. In Table 3.1, an example of an event log is provided. Each row in this event log represents an event. For instance, resource Gert started the print job for case 1 at 14:25 on the 26th of November, 2017, which is shown in the fifth row. The

[1] https://fluxicon.com/disco/
[2] http://www.promtools.org

sixth row shows that he finished this activity an hour later.

The event log in Table 3.1 is presented in a tabular format which holds a list of events with its attributes. In order to promote the usage of event logs, the *IEEE Task Force on Process Mining* recommends the use of the *eXtensible Event Stream (XES)* format, in which each XES-file contains an event log that fulfils certain requirements [64]. This way, the adoption of process mining tools capable of analysing event logs can be leveraged. Different existing process mining tools contain the feature to convert event data such as the example in Table 3.1 easily to an XES-file.

Table 3.1: Example of an event log.

Case id	Event id	Timestamp	Activity	Transaction type	Resource
...	...	...	...	...	...
1	325603	25/11/2017 14:25	Set up print job	start	Hanne
1	325603	25/11/2017 15:25	Set up print job	complete	Hanne
1	325604	26/11/2017 12:35	Check print job	start	Toon
1	325604	26/11/2017 12:45	Check print job	complete	Toon
1	325605	26/11/2017 12:45	Printing	start	Gert
1	325605	26/11/2017 13:45	Printing	complete	Gert
2	325609	28/11/2017 9:30	Set up print job	start	Hanne
2	325609	28/11/2017 9:42	Set up print job	complete	Hanne
2	325612	29/11/2017 10:10	Printing	start	Gert
2	325612	29/11/2017 14:49	Printing	complete	Gert
1	325606	5/12/2017 17:30	Packaging	start	Niels
1	325606	5/12/2017 18:22	Packaging	complete	Niels
1	325607	6/12/2017 15:24	Delivery	start	Jeroen
1	325607	6/12/2017 18:42	Delivery	complete	Jeroen
2	325613	10/12/2017 16:36	Packaging	start	Jeroen
2	325613	10/12/2017 17:02	Packaging	complete	Jeroen
2	325614	11/12/2017 8:32	Delivery	start	Niels
2	325614	11/12/2017 8:59	Delivery	complete	Niels
...	...	...	...	...	...

3.3.2 Types of process mining

As displayed in Figure 3.5, three types or classes of process mining techniques can be distinguished, which are process discovery, conformance checking, and enhancement. These will be shortly described in this section.

When performing a *process discovery* task on an event log, a process model is extracted from the event log without using any additional information [162]. Most discovery research focuses on control-flow discovery, where both the activities and the relationships between the activities in a process are determined [158]. One of the first control-flow discovery algorithms is the alpha algorithm presented by van der Aalst et al. [166]. This algorithm analyses the executions recorded in an event log to build a Petri net that represents the real process by finding basic process patterns. However, the alpha algorithm has problems dealing with some constructs and has been claimed to be sensitive to noise and incomplete event logs. Therefore, de Medeiros et al. [38] and Wen et al. [177] presented the alpha^{+} and alpha^{++} algorithm respectively, in order to address these deficiencies. Other well-known process mining discovery algorithms are the HeuristicsMiner [176], the Fuzzy Miner [63], the Genetic Miner [39], and the Inductive Miner [89]. An elaborate overview of these and other process mining discovery algorithms is given by De Weerdt et al. [42] and vanden Broucke [169]. Despite the fact that many algorithms have been presented, no algorithm can be defined as the best over the other algorithms and each of them is based on a different set of assumptions and parameters [129].

Secondly, in *conformance checking*, the discovered process model and the underlying event log are compared with each other in order to discover deviations between the model and reality. Some existing conformance checking methods are alignment techniques [36, 161] and token-based replay [135]. A comprehensive overview of existing conformance checking algorithms and metrics per dimension is given by vanden Broucke et al. [170]. Rozinat et al. [132] and van der Aalst [155] state that the conformance or the "goodness" of an event log can be expressed with four conformance checking dimensions or quality criteria, which are fitness, precision, generalization, and structure or simplicity. An overview of these dimensions is given in Figure 3.6.

The first criterion, *replay fitness*, specifies if all observed behaviour in the event log "fits" the model, which means that the process model is able to correctly replay the traces in the event log. Secondly, the *precision* or the appropriateness of a process model measures how precise the model is for the event log under consideration. A good process model should not allow for too much behaviour which means that it is too general or underfitting the observed behaviour. On the other hand, the model should not be too precise and overfitting should be avoided. This is measured by the *generalization* dimension. Generalization is defined by Buijs [22] as *"the likelihood of previously unseen but allowed behaviour being supported by the process model"*. Finally, process models should be as simple as possible in order to be comprehensible for humans. This last dimension, the structure or *simplicity* of the process model, is closely related to the modelling language and the semantics that are used to represent the different elements in the process. Simplicity reflects the lack of process complexity and thereby serves as the "Occam's razor" principle, stating that *"one should not increase beyond what is necessary"*. These four quality dimensions are competing which means that a balance between them should be found when discovering process models from event logs [157]. For each of these dimensions, several metrics have been developed and implemented, of which an overview is given by vanden Broucke et al. [170].

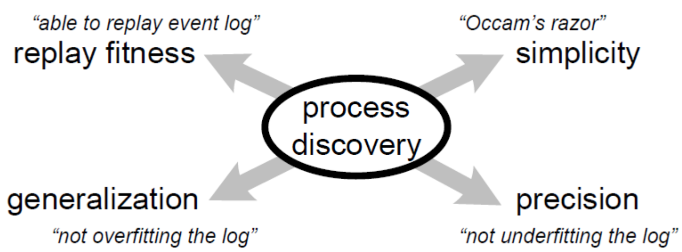

Figure 3.6: The four quality criteria of process models [155].

The distinction between these dimensions was also discussed by vanden Broucke et al. [170], who define fitness, precision, and generalization as accuracy metrics and simplicity and structuredness are seen as two different dimensions that are defined as comprehensibility metrics. Fitness and precision have been given the most attention in recent years, for the obvious reason that both the model and log are known. In contrast, generalization has been more difficult to capture, since the behaviour of the underlying business process is generally unknown. Only a few measures currently exist for this quality dimension. Therefore, CoBeFra was presented by vanden Broucke et al. [170]. This comprehensive benchmarking framework can be used to determine different metrics to measure and analyse the conformance of a process model. However, the effectiveness of existing measures and metrics of these four quality dimensions is questioned by Janssenswillen et al. [76].

Finally, as in the case of conformance checking, for *enhancement* also both an event log and a process model are required to enhance the process model. However, instead of checking whether reality conforms to the model and vice versa, this type of process mining has the objective to modify or improve the a-priori model [155]. Different types of process enhancement are changing the process model, in order to better reflect reality, or extending the process model with more information such as activity duration [161, 165] or decision logic [133, 134].

In contrast to these three types of process mining, which all require by some means the discovery or development of a process model, process mining is in this dissertation defined in a broader form, as it also entails the retrieval of knowledge from event data without the need for a process model. In this respect, the next section focuses on the distinction between model-based and event log-based performance analysis.

3.3.3 Model-based vs event log-based performance analysis

As stated above in the paragraph about conformance checking, the quality of discovered process models is determined by measuring the fitness, precision, generalization, and simplicity of the process model. Although the simplicity dimension only concerns the process model and not the underlying event log that is used to discover it, the other quality dimensions concern the interplay between three different groups. These are the behaviour that can be observed in the discovered process model (M), the event log (L), and the behaviour that is allowed by the organisation or the context of the process, which is referred to as the "system" (S) [22]. As can be seen in Figure 3.7, the observed behaviour is not exclusive for each of the three elements, so an overlap is present. The behaviour in the system is not easy to be captured and described, because of uncertainty and instability in reality. Buijs [22] presents some metrics to compare the behaviour recorded in the different elements and states that *"the traditional goal of process mining is to find a process model that describes the system as accurately as possible, using nothing more than the observed behaviour in the log."* van der Aalst [157] also investigated the quality dimensions in order to find the "right" process that describes reality as correct as possible.

However, Janssenswillen et al. [76] state that most existing metrics measure the quality of a process model with respect to the event log from which it was discovered and not the underlying system, ignoring the fact that event logs only contain a portion of the complete reality. Therefore, the authors present alternative quality dimensions to measure the distance between the event log and the discovered model, and between the process model and the underlying system. The presented quality dimensions are log-fitness, log-precision, system-fitness, and system-precision, which can be seen as replacements of the existing measures fitness, precision, and generalization [76].

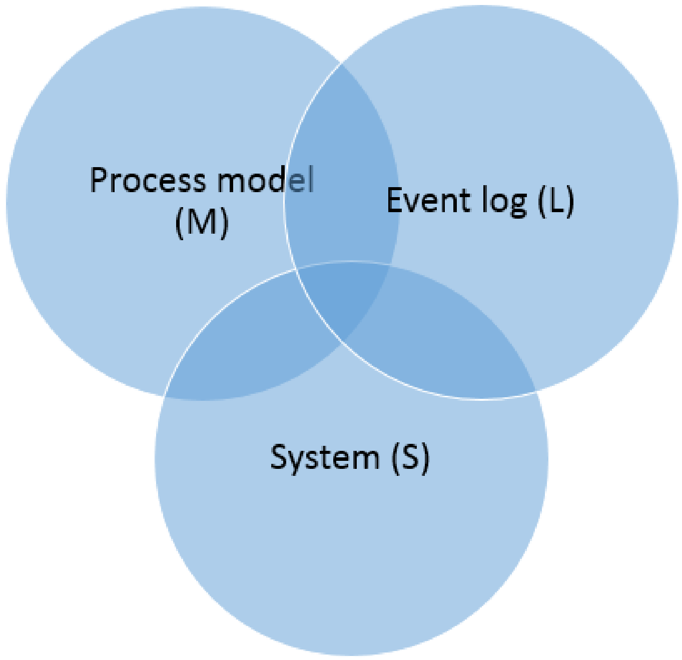

Figure 3.7: The interplay between the behaviour of the process model, the event log, and the system [22]

It can thus be concluded that most process mining research is concerned with the alignment between the process model and the event log [41]. However, as these models are learned from the event logs with certain algorithms based on parameters and assumptions, and are often manually manipulated with sliders and filters in visualisation tools, it may be possible that unobserved behaviour appears in the model. Conclusions taken based on these models can therefore be less reliable or even incorrect as they possibly contain unobserved behaviour or they do not contain all information of the business process under analysis.

In this respect, it can be useful to have a look at the difference between the two types of analysis that were distinguished in the BPM lifecycle in Figure 3.4. These two types are (i) model-based analysis, based on the process model that has been discovered, and (ii) data-based analysis, based on the underlying event data extracted from the system [156]. Analyses that are purely based on a process model can be divided into two types, which are *verification* (checking the accuracy of the process model) and *performance analysis* (improving the process regarding the time, cost, or quality of the process). The latter can be defined as more interesting for business people as the former is more focused on the *soundness* (a minimum correctness criterion that a process model should fulfil) of process models. However, to be able to perform model-based performance analyses, process models of high quality are required as analyses about the reality are only useful in case the model represents the reality as accurately as possible. van der Aalst [156] states that process mining can be seen as a solution for this shortcoming as it uses event data to create process models or to check the conformance of process models. However, after a comprehensive survey of 289 BPM-related papers performed by van der Aalst [156], in which the actual use of process mining was analysed, it was found that monitoring and analysing the performance of processes based on event data occurred in only very few papers, while the largest share could be attributed to process model discovery and process model enactment.

Next to this, Günther and van der Aalst [63] stated that real-life processes are much more unstructured than is expected by people, resulting in what they call "spaghetti models". Too often, process mining experts make assumptions about the event log and about the process models discovered from these event logs, resulting in incorrect interpretations. To overcome this problem, the authors introduced the Fuzzy Mining approach, which supports the ability to simplify the process model discovered from a process. This approach can be compared with classical road maps, which aggregate and abstract information, which emphasise the most important information, and which are customised for specific purposes. Following these analogies, different log-based process metrics are introduced to improve the simplification and visualisation of process models, based on *significance*, i.e., the relative importance of behaviour, and on *correlation*, i.e., the congruence between events. Behaviour that is defined as "uninteresting" will be hidden or even completely removed from the simplified model. However, although the resulting models will be more understandable, they are still based on human-chosen parameters and other settings.

Based on these findings, we can state that the analyses performed on process models that are discovered from event logs can be subjective and prone to errors. Business people should therefore be careful when using process models as the only measure to base their improvement decisions on. In response to this, this dissertation will introduce measures that are based solely on the event log data of a business process in order to provide business people with more objective insights in their processes, without the need of a process model. The requirements of these measures will be identified in Chapter 4. However, first an overview of existing work on the interplay between process mining and operational excellence, and existing performance measures are presented in Section 3.4.

3.4 Process mining in operational excellence

3.4.1 Related work

The link between process mining and operational excellence was already shown in the Process Mining Manifesto [160], by van der Aalst [155], and by Dumas et al. [47]. Process mining is claimed to be a supporting technology for approaches such as Six Sigma, business process improvement (BPI), corporate performance management (CPM), and total quality management (TQM), in order to strive for operational excellence.

van der Aalst et al. [161] built on this for their model-log comparison in order to check conformance and to analyse process performance. They stated that a ‘good’ alignment between the discovered process model and the underlying event log makes it possible to replay the events in the log on the process model and to subsequently identify bottlenecks or perform other performance analyses.

In a case study performed at a Dutch bank, it was shown that the DMAIC-cycle of Six Sigma could be substantially accelerated by applying process mining as an additional technique in each of the phases [167]. They enumerate some difficulties of the use of the Six Sigma cycle that can be reduced by the use of process mining, such as the dependency on relevant and qualitative data, the expensive and time-consuming data analyses, and the translation to practical use cases. Based on an analysis of the relevance of process mining throughout their research, some recommendations for the different steps in the DMAIC-cycle are presented. With or without process mining, starting with an accurate *definition* of the objective and scope of the analysis will pay off in the following stages. Most problems that occur in the *measure* step deal with the collection and quality of the collected data. With process mining, the data can be retrieved from existing information systems, after some requirements are met, in order to lower the disagreements and problems concerning the data and in order to make the analysis repeatable. In the *analyse* phase, the authors used the tools RapidMiner[3] and Disco[4] to create benchmarks for their analyses and to find bottlenecks in the process. Root causes for these bottlenecks can also be found via the visualisation tool. Also involving the people working in the process from an early stage and including them in the analysis is easier when the process is visualised and will result in more accurate conclusions. The root causes can then be translated to measures and concrete actions to *improve* the process. Conclusions can be drawn

[3]http://rapid-i.com
[4]https:// fluxicon.com/disco/

much faster from the visualisations which will also improve the practical use. In the final step, the *control* phase, process mining can be used to support more analyses as data collection can be easily repeated. Based on these operations of process mining, the authors could conclude that the Six Sigma cycle was accelerated from 9-12 weeks to 4-6 weeks [167].

Other examples of use cases in operational excellence for which process mining can be useful are:

- Companies should have an understanding of their as-is situation before improving their business processes using lean management and Six Sigma. This picture of the as-is situation can be accomplished by visualising the process.
- Customers often prefer customised options. Process mining can be applied to find out how frequent a certain process variant is and what the costs will be for deviating from the "happy flow".
- Lean management is all about the flow that adds value for customers and removing waste from this flow. Process mining can be used to identify the points that cause variability in this flow.
- Constraints or bottlenecks in the process should be removed too, which can be visualised with the animation function in the tool Disco.
- "Doing things right the first time" implies that rework is reduced, resources are deployed optimally, and loops are detected in the process model.
- Companies that should comply with legal aspects also need an insight in where deviations occur.
- And finally, based on different process mining analyses and techniques, predictions for future process behaviour can be indicated.

On the other hand, instead of looking at how process mining can be used to implement operational excellence, Six Sigma was also presented to cover techniques that can be useful to perform business process management [28]. By applying methods such as cause-and-effect diagrams, check sheets, pareto diagrams, and root cause analysis to a helpdesk process, Conger [28] found that these techniques can all be useful. However, many of them are required together to uncover all aspects of a problem in a business process and it is not clear which combination of techniques should be used in which situation. Next to this, each technique has its own method of presentation which can result in time-consuming interpretations.

Based on this overview of existing applications at the overlap between process mining and business process management on the one hand, and operational excellence techniques on the other hand, we can state that the work that has been done at this interplay is rather limited given the overlap between both fields. It can be concluded that the opportunity to use process mining as a useful supporting method for operational excellence should be played out more.

3.4.2 Existing process performance measures

This chapter shows that, despite the potential of process mining to support operational excellence, significant research challenges are still ahead to integrate both fields. As was presented in Chapter 2, companies should continuously measure the performance of their processes and are therefore forced to look for the "right" process performance criteria. Heckl and Moormann [66] outline different process performance measurement techniques in order to help process owners to manage their processes in accordance with their performance. However, the authors found that there is no single process performance measurement system that can be recommended to process owners as companies should rely on custom-built approaches that are adapted to its specific objectives and company strategy. Some examples of existing performance measurement techniques can be positioned within a model containing two variables of performance, which are focus and scope. Process performance measurement systems can be defined to be concerned with performance in a broad sense, investigating both efficiency and effectiveness of company processes, in contrast to performance measurement techniques such as activity-based costing, work-flow based monitoring, or statistical process control, which all focus mainly on the efficiency. Process performance measurement systems also focus on single business processes in contrast to techniques such as balanced scorecards and self-assessment, which are usually concerned with the entire company. The analysis of the performance of a process should thus entail both qualitative and quantitative measurements and should not only focus on the efficiency of the process in order to reduce costs, but also on the process effectiveness in the interest of more profit-enhancing matters [66].

In order to perform the measurements in practice, Heckl and Moormann [66] defined three categories of process performance concepts, which are (i) performance indicators, or classes of performance measurement, (ii) performance measures, which are the actual calculations to make the performance indicators operative, and (iii) performance figures, such as the benchmarks against which the performance measures are to be compared to interpret the objectives of the company. The different classes of performance measurement that are presented by Heckl and Moormann [66], based on an elaborate literature review, are (i) quality, (ii) time, (iii) cost, and (iv) flexibility.

These four classes of performance measurement are also discussed by others, such as Dumas et al. [47], who refers to them as the *Devil's Quadrangle* since improvement or redesign of a business process ideally decreases the throughput time, increases the quality of the delivered product or service, lowers the execution cost, and improves the way the business process handles process variation. Improving or focusing on one of the performance measurement dimensions automatically influences the other dimensions in a negative way. Several other authors refer to the same dimensions of performance measurement, such as Reijers and Mansar [128] who use these categories to evaluate different best practices for business process redesign, and Kis et al. [82] who recently presented a framework of process performance measures based on the Devil's Quadrangle. The latter presents a measure for each of the four dimensions, time, costs, quality, and flexibility, based solely on event log data, which is also the focus of this dissertation. In contrast to other work in which it is not clear how the measures are calculated, Kis et al. [82] clearly define the requirements of the event log data that is used as input for the calculations. However, currently the authors try to capture each dimension of the Devil's Quadrangle within a single measure by combining multiple measures, implying the risk that certain results of individual measures are ignored. Moreover, also in the field of operations and production management, the four aspects of the Devil's Quadrangle are used to categorise measures of performance. An overview is given by Bhagwat and Sharma [17], De Toni and Tonchia [40], and Neely et al. [111]. However, as the focus of these performance measurement systems and approaches is mainly on manufacturing and operations, they are excluded from this dissertation. Finally, as was already stated earlier, many metrics have been proposed to measure the compliance of discovered models with their underlying event log in the field of process mining, yet the number of measures applicable solely to event log data is limited.

An overview of existing measures provided by Anupindi et al. [10], Dumas et al. [47], and Kis et al. [82] is presented in Table 3.2, categorised based on the performance measurement dimensions of the Devil's Quadrangle.

Note that this table is not exhaustive, as many performance measures and metrics are not formally defined or mentioned in literature, and an exhaustive overview of which metrics to use in which situation, is missing. Moreover, the input requirements or calculation methods of most measures are not explicitly defined, as well as the level of analysis that is used, which makes it hard to evaluate and compare them. Finally, performance measures defined in other fields such as, among others, opera-

Table 3.2: Existing process performance measures, categorised based on the Devil's Quadrangle [10, 47, 82].

Time	**Quality**
cycle time/throughput time/flow time/lead time	internal quality, such as level of variability or level of control
waiting time	external quality, such as client satisfaction
processing time/service time	product quality
synchronisation time	number of repetitions
	outcome quality (was the planned path followed?)
	technical quality (did any incidents occur?)
Cost	**Flexibility**
cost of production	number of distinct executions
cost of delivery	level of concurrency
cost of (human) resources	number of decision points (in the model)
fixed costs vs. variable costs	resource specialisation
overhead costs	resource capacity
operational costs	run-time vs. build-time flexibility
inventory	ability to handle multiple cases
flow rate	ability to change the structure of the process
	volume flexibility (constant handling of cases)
	technical flexibility (time spent on incidents)

tions management, queuing theory, and simulation, are excluded from this overview. Nevertheless, the overview provides a first insight in existing process performance metrics which can be used as a basis for the artifacts that will be created in this dissertation, to overcome the shortcomings in the interplay between process mining and operational excellence.

3.4.3 The added value of process mining

From Chapter 2 it was concluded that existing operational excellence techniques are mostly based on paper-and-pencil methods which are not able to respond quickly to changing conditions or environments. Most lean management and even Six Sigma techniques are therefore rather qualitative as they are often not based on event data. This results too often in decisions that are taken based on subjective interpretations or that depend on the person or team executing the process or performing the analyses. Moreover, these approaches are considerably time-consuming and not repeatable. Next to this, there is also no overview of which techniques or which combination of methods is prefered in which situation.

Based on these conclusions drawn from Chapter 2, together with the findings in the chapter at hand, we can state that there are some shortcomings in the operational excellence field that can be resolved by using process mining techniques and tools that have been presented during the last decades. Therefore, an insight is provided in the link between the most common principles and methods from the operational excellence methodologies presented in Chapter 2 and how event log data can be useful to support or enhance these methodologies. This link is presented in Figure 3.8, where some common principles and methods of lean management, Six Sigma, and theory of constraints are complemented with specific analysis requirements which were found in literature. Next, the link between these analysis requirements and how event log knowledge can be useful to support these requirements is provided.

Firstly, the analysis of the general insights gained from an event log, such as the identification of activities, resources, cases, and patterns (or traces), is straightforward and therefore omitted from the overview in Figure 3.8. These general event log knowledge aspects are not specifically related to one or some of the analysis requirements from the operational excellence field as they are basic measurements that are used to retrieve and calculate the event log knowledge concepts provided in Figure 3.8.

Secondly, it can be noted from Figure 3.8 that multiple operational excellence principles and methods are related to the same event log knowledge concept, confirming again the overlap and joint nature of the three methodologies.

Based on this overview, it can be stated that the application of process mining and the use of event log knowledge in the field of operational excellence can be beneficial. First, the use of event log data as a source for the concepts of process discovery and knowledge retrieval can be a response to the shortcoming that most operational excellence techniques are based on qualitative, paper-and-pencil techniques, and on human opinions and decisions. With more quantitative and objective measures, the analyses that are performed are more easily repeatable and reproducible, and trends in measurements over time can be analysed, which is also noted as an important underlying principle of Six Sigma and total quality management. Furthermore, insights from event log data can support initiatives which aim to reduce waste and non-value-adding activities in a process and it can support the DMAIC-cycle of Six Sigma as was explained above. Moreover, constraints, bottlenecks, and batch processing can be detected rather easily and predictions for future process behaviour can be suggested. Process mining can thus be defined as an additional approach to analyse the as-is situation of a business process in order to get insights in the current state of the process including its weaknesses, and to lay a foundation for process redesign.

Secondly, by making use of event logs instead of qualitative and interview-based data that can be rather static and subjective, process mining can be used to transform objective and more correct data to knowledge about the different aspects of a business process. However, in order to improve business processes and their performance, the decisions and subsequent actions taken should be as correct as possible, and therefore based on reliable and trustworthy data. As was found in this chapter, the field of process mining focuses mainly on the discovery of process models from event logs. This aspect can be categorised under the 'identify process flow' element in Figure 3.8, which is shown to be an important and useful aspect for the analysis requirements of operational excellence. However, these models often lack the ability to provide an accurate picture of reality as they are discovered based on discovery algorithms containing assumptions and parameters that should be chosen upfront. In the visualisation tool Disco, for instance, process owners can visualise their business process with filters and sliders, determining the level of analysis and precision as desired. As these models and visualisations are presenting only parts of reality, or a simplified version of reality, the actions and conclusions concerning process performance and process optimisation taken based on these insufficient or even incorrect sources may be misleading or unreliable. Companies therefore run the risk to take actions and make changes, and subsequently incur costs, with the wrong intention. The field of process discovery can thus be broadened with the retrieval of knowledge from event logs without the use of an intermediate process model.

Building on these findings, some interviews were conducted to further analyse the needs of business process experts, in order to introduce artifacts that solve the shortcomings in the interplay between both fields defined above. The characteristics and findings from these interviews will be elaborated upon in Chapter 4 in order to create an overview of all requirements for the artifacts, based on the interconnection between the results of the interviews and the findings from literature.

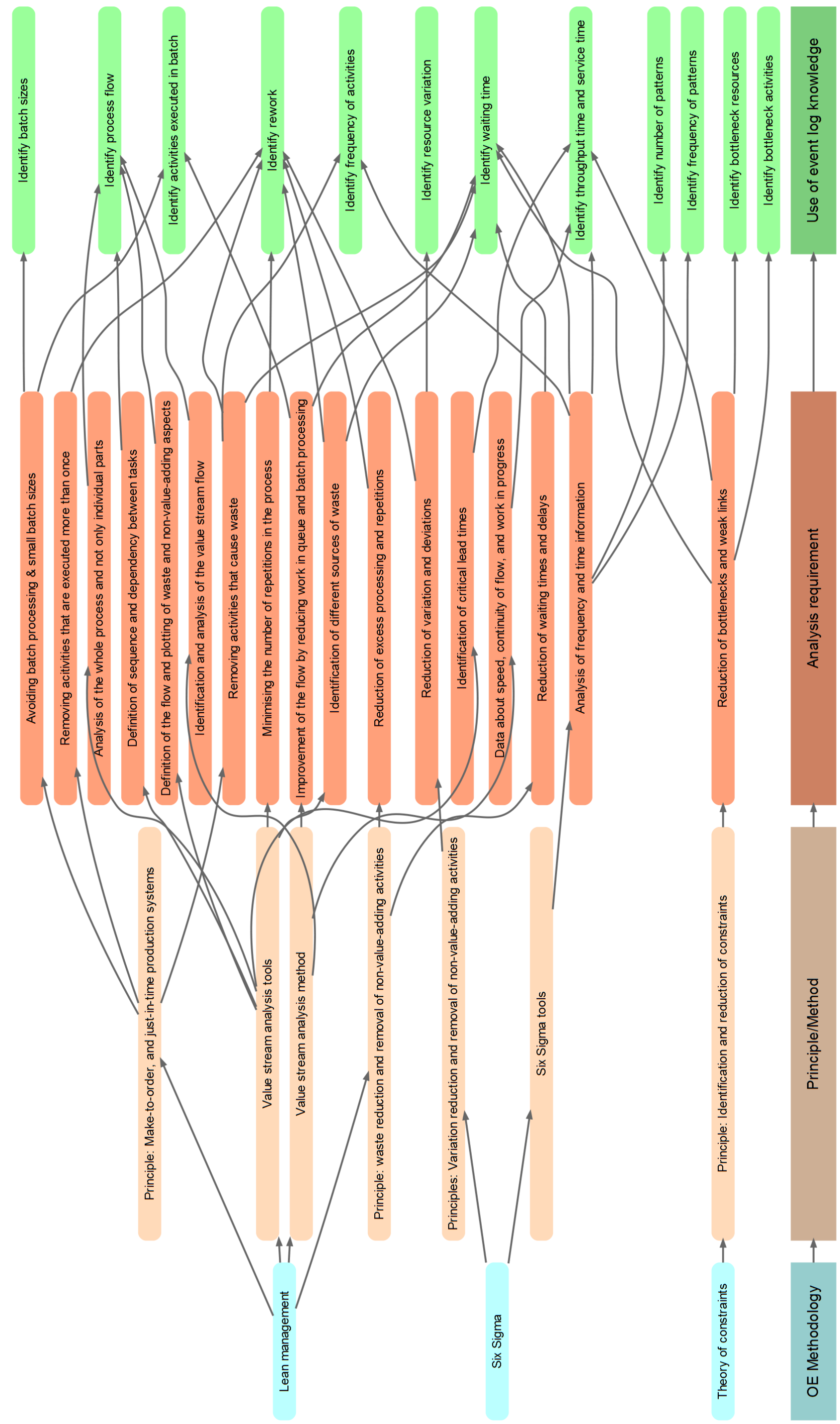

Figure 3.8: Useful insights from event log knowledge in operational excellence.

3.5 Conclusion

This chapter presented an overview of process mining and the work that has been done at the interplay of process mining and operational excellence.

From Chapter 2 it is clear that continuous performance measurement of business processes has been hard to accomplish as no clear overview has been presented of which guidelines and techniques should be used by companies. Many existing techniques are qualitative and based on paper-and-pencil approaches with a lack of support of data-based analyses. This also implies that the operational excellence approaches are rather subjective and depending on the person performing the analyses and the team involved in the process. Therefore, the need for more objective, data-based techniques has been identified. In the field of business process improvement, some authors claim that specific guidelines for business process optimisation are missing in literature and that it is not always clear from literature how existing techniques are used to support the process of business process improvement. Process mining was therefore introduced, as it shows to be promising in the field of knowledge retrieval based on process data that is collected from process aware information systems in companies. However, most research on process mining focuses on the discovery of process models from event logs and the compliance of these models compared to the event log. These models are learned from event logs with certain algorithms, based on parameters and assumptions, and often manually manipulated with sliders and filters. This may result in less reliable conclusions and improvement measures that are possibly based on incorrect or incomplete process models that contain unobserved behaviour or that do not contain all information of the business process under analysis. We can thus conclude this chapter by stating that there is a gap that we want to bridge between operational excellence and process mining and that there is a need for an artifact that uses the knowledge from event logs to support business process improvement in the light of operational excellence. This artifact should focus on one or more of the performance measurement categories that have been identified in literature, which are time, structuredness, cost, and quality. Although existing process performance metrics have been presented, these metrics lack specific input requirements, clear calculation guidelines and methods, and level of analysis, which makes them hard to compare and evaluate.

Chapter 4

Artifact requirements

4.1 Introduction

Chapter 2 outlined the key concepts of operational excellence within the evolution of quality management and some well-known process improvement methodologies. It was noted that existing process improvement techniques are rather subjective and often based on paper-and-pencil approaches. Chapter 3 added a high-level overview of the process mining field and existing applications of process mining in the light of operational excellence to this. The field of process mining focusses mainly on the discovery of process models from event logs that contain activities performed in the underlying business process, and the performance of a process is mostly measured by means of these discovered process models. However, these discovered process models do not always represent reality perfectly as they are discovered from the event logs based on algorithms containing assumptions and parameters that should be chosen upfront, implying that unobserved behaviour can possibly appear in the model. This results in a risk when conclusions and decisions are taken based on these models.

In order to acknowledge the shortcomings that were listed in the problem investigation step of the design science research framework, this chapter (Figure 4.1) investigates the specifications of the artifact that will be introduced in this dissertation to overcome these problems. This investigation will be accomplished by analysing the requirements that should be taken into account when this artifact is designed and developed. Johannesson and Perjons [78] define a requirement as *"an artifact that is deemed as desirable by stakeholders in practice and that is to be used for guiding the design and development of the artifact"*. A multitude of requirement engineering

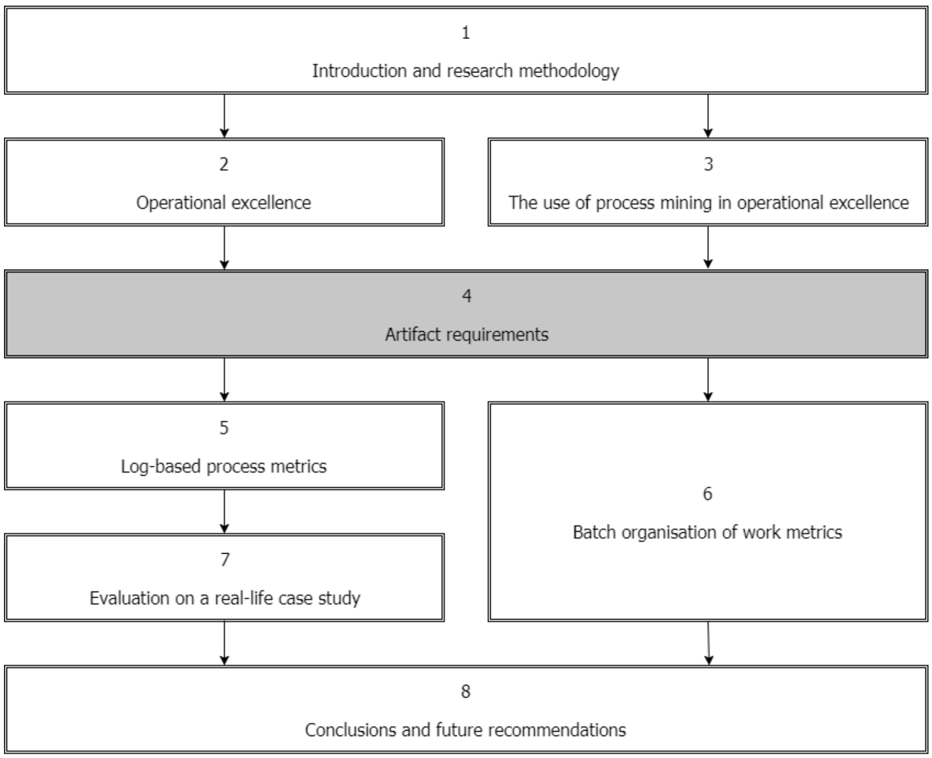

Figure 4.1: Outline of the thesis - Chapter 4.

methods have been developed, of which an overview is presented by Wieringa [179].

For this requirements analysis, the literature review that was provided in Chapter 2 and Chapter 3 and the interviews that are described in the chapter at hand are taken as a starting point. Firstly, the required artifact is introduced in Section 4.2. Secondly, to verify if the findings from literature hold in practical environments, interviews have been conducted with business process analysis people. Section 4.3 gives an overview of the interviews with three business process experts from service companies that have been questioned about the topics that were found in literature. Next to a short overview of the three different respondents, a description of how the interviews were conducted and the insights that were found in the interviews are provided. The results from these interviews are then compared to the findings in literature in order to define the artifact requirements in Section 4.4. Finally, conclusions are drawn in Section 4.5.

4.2 The need for an artifact

Based on the shortcomings described in the previous chapters, it can be stated that there is a need for an artifact that uses unbiased event log knowledge to support operational excellence techniques. As was seen in Chapter 3, in process mining research the discovered model is usually built based on different assumptions and parameters, and in contrast to most process mining research that is concerned with the alignment between the process model and the event log [41], the artifact that is required here to solve the shortcomings focuses on the objective measures that can be directly read from the event log.

Therefore we propose that a list of event log-based metrics is created, which will provide a process owner with unbiased, algorithm-agnostic information of the event log, as a starting point for a process analysis. An overview of existing performance measures was provided in Chapter 3, as well as a visual overview of the link between the operational excellence methodologies that were presented in Chapter 2 and the usefulness of event log knowledge in some of the most important underlying principles and methods of these methodologies. From this overview and the overview of existing performance measures it could be stated that many metrics already exist to support the operational excellence field. However, some drawbacks of these metrics were already mentioned in Chapter 3, which will be repeated here. Firstly, many of the existing metrics are based on models, which are discovered or built on experience, or on qualitative and subjective data such as human experiences and interpretations. Moreover, the collected data is often only a sample or a subset of the entire process data. Secondly, existing measures lack specific guidelines for data collection and for calculations, making it hard for the process analyst to decide which actions to take. Finally, existing measures are presented by different authors, in different fields, such as operations management, business process management, and operational excellence. As a result, their procedures and tasks are all different and hard to compare and evaluate.

It can therefore be stated that the developed metrics should be based on event log knowledge, for which the requirements should be clear. Moreover, the metrics should be concerned with one of the four different categories of process performance measures that should be focused on in a business process improvement project, as was seen in the previous chapter. These are the four aspects of the Devil's Quadrangle: time, structuredness or flexibility, cost, and quality. The measurements should also involve one or more of the aspects of operational excellence that were defined in Chapter 3. These are, among others, batch processing, waste, rework, and non-value-adding activities. In this respect, an overview of required measures was already provided in Chapter 3, where the usefulness of event log knowledge to support operational excellence was defined. Finally, the artifacts that are created should be applicable on different levels of analysis and they should serve as a means to objectively compare different event logs in terms of the different aspects of process performance.

4.3 Business process expert interviews

To get further insights in the requirements that should be taken into account when this artifact is designed and developed, the findings from literature are complemented with the results from business process expert interviews that have been conducted during this research[1]. The interviews are taken from three business process experts from two different service companies who are all responsible for the analysis and/or optimisation of the business processes within their organisational unit. The two companies both are large utility providers in Belgium. The interviews that are conducted can be defined as semi-structured, which means that they consist of a set of predefined questions that do not require a response from a predefined list, but can be answered with an open answer by the respondents [78]. This type of interviews was chosen to be able to ask more information in case an interesting topic would come up, which is not possible with structured interviews. The interviews took place in March 2016 and for the selection of the respondents, the theoretical sampling technique was applied, since the aim is to generalise analytically, not statistically [24, 50]. In contrast with statistical sampling in which a random sample from a population is chosen for analysis, here the respondents are chosen very specifically, as they are all working on process improvement in service companies. Moreover, respondent one and two operate at the tactical level and are mainly responsible for the continuous improvement of the business process, while respondent three is involved in the strategic improvement of multiple business processes within her company. Finally, a fourth person, who has a consultancy function in the field of process optimisation, was excluded from the interview round as he was involved in each step of the design cycle that is used throughout this dissertation. While he was found to be too engaged for these interviews, his input throughout the dissertation funds both the requirement analysis and practical applicability of the developed artifacts.

[1]For privacy reasons, the full transcription of the interviews as well as the identity of the respondents will not be provided in this dissertation. The interview questions can be found in Appendix A.

4.3.1 Interview respondents

Before analysing how the interviews were taken, a short overview of the three respondents is given in this section.

Respondent one has been working as a business process analyst for more than two years, which means she is responsible for the continuous optimisation of different processes within the utility company she is working for. This includes solving small problems within the process as well as more structural changes throughout the complete process. One of the main processes that respondent one is involved with can be divided into two types; a business-to-customer (B2C) process, which is mainly automated, and a business-to-business (B2B) process, which is handled manually for most of the parts of the process. The latter is much more unstructured and contains more exceptions than the former, which therefore represents better the "happy flow" of the process.

Respondent two, also a business process analyst at another utility provider, is responsible for developing the business processes, maintenance, process optimisation, and improvements on the supporting IT systems such as SAP and mobile systems. Next to this, he also collects data and controls different key performance indicators and process indicators that have been defined for the processes in order to report the results to other departments and identify the causes of certain problems. The process he mainly works on is the process concerning customers requesting a connection, which will also be used in the case study in Chapter 7. Most of this process has been automated. However, a lot of manual checks still occur, mostly to customise the process for each customer. Routine tasks have mainly been automated, but in order to automate more, the company should standardise its service, implying that the range of services offered to customers needs to be reduced.

Respondent three finally, is the head of the process optimisation department in the same company as respondent two. She is mainly responsible for supervising the actual optimisation of the processes which also includes the training of employees working on the process and providing them coaching and education courses in order to optimise their techniques and procedures. They also try to provide documentation for all processes, including instructions (how to perform an activity), guidelines, and documents containing knowledge about each subprocess.

4.3.2 Interview design and results

As the goal of the interviews is to analyse the problems and shortcomings with process performance optimisation in practice, different topics are discussed with the respondents. Firstly, their own experience with business processes is discussed, together with an overview of the business process(es) they are working with.

Secondly, in order to get a notion of the shortcomings of process optimisation in practice, the funnel technique was applied. A first set of broad questions was provided to the respondents to start the interviews. In a next step, more specific questions are asked about the different classes of process-oriented performance measures which were presented earlier; time, cost, quality, and structuredness.

An overview of the different topics that were covered during the interviews together with the responses of the three respondents is given next. Firstly, the concept of an *efficient business process* is discussed, followed by some more specific questions and answers about the different categories of performance measures. Finally, some problems with business processes and other remarks on process performance are dealt with. The interview insights will be summarised in Section 4.3.3.

4.3.2.1 Efficient business processes

Firstly, the respondents were asked for their impression or opinion on the concept of an *efficient business process* and how efficient the processes in their company are. Respondent one defines efficient processes as having little business incidents and little eruptions or failures within the process. Processes should also be automated in order to reduce the risk of (human) errors. An indication of the performance of a process can be measured by the number of calls by customers, as they mainly call for problems or complaints which cost a lot of time and money to solve them. This respondent does not have a perfect overview of the performance of the company processes, although different performance indicators exist to measure certain aspects of it. One example of a performance indicator is the throughput time of a customer request. However, this indicator is hard to measure correctly as the system implementation does not support the identification of different customer tickets to be connected to the same customer in case this occurs, resulting in an incorrect view on the total throughput time of a customer. Another difficulty is the decision when to contact customers as this is very time consuming, but in some cases inevitable.

Respondent two and three define an efficient process in a rather similar way which will therefore be described together. Firstly, only (or mostly) activities that add value to the end customer of the process should be in the process, unless it concerns activities that are unavoidable (e.g., because of governmental compliance rules). An efficient process also contains an end-to-end flow without delays or eruptions. An example given by respondent two of a process delay is customer information that is incomplete when it arrives at the company. The information that is required to execute the process should also be captured and stored in a correct and efficient manner, which is especially important for the analyses afterwards. Moreover, constraints should be avoided as much as possible in order to prevent inventories to pile up and all preconditions in order to keep the process running should be satisfied. Finally, the process should be executed within certain throughput time boundaries (which can be implied by an external regulator) and with minimal waiting times.

The respondents also lack a good notion of how "well" the process is running, as the measurements that exist mainly concern subparts of the processes. Moreover, only the aspects that are regulated by the government are monitored, such as the service time for the customers. However, next to these measures, different levels of key performance indicators have been defined throughout the company. But for now, too few measures exist to be able to evaluate the business processes in an end-to-end way. An interesting additional metric, according to one of the respondents, is a measurement of the quantity of inefficiencies in the process, which can be presented by the number of iterations of a certain step. This would be an indication of the amount of waste in the process.

4.3.2.2 Categories of performance measures

The categories or classes of performance measures that are given by the three respondents are very similar to the four classes that where found in literature, which are time, structuredness, quality, and cost [66]. Respondent two listed these classes exactly as they were found in literature. He mentioned timeliness of the execution of the process in order to deliver on-time to the customer what he requests, and the throughput time of the end-to-end process, as two categories which are very important in the service industry. However, he states that the end-to-end throughput time should be evaluated with care as for many steps in the process the company depends on the speed and the reaction time of the customer or other external stakeholders. The waiting time is therefore another measure that was mentioned in this context, implying that the distinction between actual service time, waiting time, and throughput time should be taken into account. These measures can be classified under the *time* dimension. Respondents one and three also defined the throughput time of the process as one of the most important aspects in measuring the performance of their processes. All three respondents testify that different measures to assess the performance of a process concerning the time aspect are already employed in the processes in their companies, as the timeliness of a process is generally linked to governmental requirements.

Concerning the *cost* aspect of a business process, respondent two mentioned that the amount of waste, the actual processing time of the different activities in the process, and the number of iterations all have an impact on the costs. However, this category is broader than only the process aspects, as other factors that are not included in the process, e.g., the materials that are used, also influence the costs. Also the other two respondents indicate that the cost of the process, i.e., how much does it cost to service one customer, is a very important classifier of the process performance, which includes many different aspects and which is linked to the other categories of performance measurement. However, all three respondents regret that currently no indicators to measure the costs of the business processes are implemented in their companies.

The *quality* of the processes was also mentioned as an important measure. Respondent two mainly discussed the quality of the data that was tracked and used for analysis purposes. The more correct the measurements and recordings are done in the different steps in the process, the more correct the results of the analyses will be. Next to the correctness of the data tracking, respondent one also focused on the completeness of the different steps in the process. The example of an invoice was given, in which a lot of elements should be filled in and should not be overseen. Respondent three finally, states that the quality of the process depends on the added value for the customer in the end. The more activities that do not add value are removed from the process, the higher the value will be for the customer, and consequently the higher the process quality will be.

Finally, respondent two and three also indicated that the amount of variation within the process should be seen as an indicator of the process performance. Respondent three clarified the concept of variation by referring to the complexity of the process; the less complex the process is for people to be executed, the more "pleasant" it is to execute it. This means, according to respondent three, that the number of activities should be minimised, and especially rework, waiting time, non-value-adding steps, and overprocessing should be avoided. Based on these cases, this category of measurements can be defined as the *flexibility* or *structuredness* of the process. All three respondents agree that the degree of flexibility or structuredness of a process is a very interesting aspect of processes to be measured, as it influences both the cost of the process and the quality. However, until now none of the respondents' companies are capable of measuring the structuredness of their processes, as metrics for this category of performance measurement have not been defined yet.

According to all three respondents, the importance of the different performance measures depends on the type of the process. It is therefore not possible to define one of the categories as more important or having a higher weight than the other categories. However, performance measures concerning the time dimension are easier to calculate and easier to apply than the other categories. As a result, these measurements are more commonly executed in practice than others. Next to this, it can also be noticed that the different categories of performance measures are not independent from each other as the cost is mainly defined by the timeliness of the process, and the quality depends on both the structuredness and timeliness of the process.

4.3.2.3 Problems and other remarks on process performance

Next to the different categories of process performance, the respondents also mentioned that the performance of processes should be measured at different levels of analysis. Mainly, the existing measures are calculated on parts of the process, as an overview of the complete end-to-end process is often missing. However, the interviews show that measurements on the level of specific activities and specific teams executing the activities can be an added value in measuring the process performance and improving the process at the most valuable places. Insights into which person executes all instances of an activity that does not add value to the end customer, and which employee contains all knowledge about a specific topic are two examples of resource-level measurements that can be of interest for companies.

The respondents also mentioned that the information systems operating the processes and collecting the data are not optimised in order to perform analyses and measurements concerning the performance of the processes. Calculations that are performed are therefore very elementary and still too often based on human interpretations and statements, making them not sufficient for thorough decision making. Moreover, measurements become difficult very fast as business people are not educated to perform and interpret advanced process mining analyses.

4.3.3 Interview insights

Based on the answers of the three respondents, the following conclusions can be stated.

An **efficient business process** is a process that contains only (or as many as) possible steps that add value to the end customer of the process, which can be internal or external, and which can not be avoided because of regulations. Also bottlenecks, delays, and eruptions should be eliminated in order to optimise the process and to minimise the idle time within the process, and all data should be captured and stored in a correct and automated manner, if possible. This all should be done while taking into account all external requirements that impact the process.

Concerning the different **categories of process performance**, the interviews show that the four categories that are referred to as the Devil's Quadrangle in literature are also present in practice. Some examples of specific measures that are used in the organisations of the interview respondents are:

Time

- Timeliness of the execution, which is often related to regulations.
- Throughput time on different levels of analysis (activity, end-to-end, parts of the process). Here the interaction with the customer should be taken into account, which results in actions taking up time that the company has no influence on.
- Waiting time, both the one that the company has no influence on, as the one that is induced by internal causes.

Cost

- The cost related to waste, such as iterations that occur for steps not being executed correctly from the first time.
- The cost of the throughput time of the different steps.
- The cost of the number of iterations of a certain step.
- The cost that is linked to external elements, such as the materials that are used or the underlying information systems.
- The cost of personnel and people working on the process.

Quality

- The quality of the execution of the process, especially the data quality.
- The added value for the customer.
- The completeness of the process execution.
- The number of complaints or calls that are made by the customer. This is related to the first-time-right principle, which should be pursued as much as possible.

Structuredness

- The number of activities executed in the process and the number of iterations for each of the activities.
- The number of activities that are not adding value to the customer.
- Rework and overprocessing.
- The sequence of activities in the happy flow and the percentage of cases that follow this happy flow.

Finally, it can be concluded from the interviews that the following characteristics concerning the measurement of process performance are identified as missing elements in practice:

- Different levels of analysis should be taken into account, such as the activity level, the end-to-end process level, and subparts of the process.
- A clear and straightforward definition of the different categories of process performance in a business context should be provided.
- Especially indicators for structuredness and process complexity are missing.
- The findings should be easy to interpret for business people, and made visually in order to increase the understandability.
- Related to the previous, the concepts that are used in the analyses and in reporting the findings to the business people should be understandable to them. A translation of the rather technical concepts to more business-wise concepts will improve the interpretation by the right people.
- Techniques to identify bottlenecks, delays, and obstacles should be presented.
- The link between the different measures and between the different categories of measures should be taken into account.
- Human-based findings and results should be dismissed as much as possible as they can contain mistakes and can be very subjective. The human input should therefore be complemented with objective data collected by the system in order to perform more correct analyses.

4.4 Overview of the artifact requirements

Starting from the gaps that have been found in literature between operational excellence on the one hand and process mining on the other hand, supplemented with the findings in the business process expert interviews, the requirements for a solution to overcome this divergence can be defined. The artifact that will be created to solve the described problems needs to meet a number of requirements [78]. Before defining these requirements, the choice of the artifact will be explained.

4.4.1 The artifact

As was already stated in Chapter 3, in most existing process mining research event logs are used for the discovery of process models, from which subsequently conclusions and improvement measures are taken. However, this implies a risk that the discovered models, as they are based on algorithms and assumptions, contain information that did not occur in the event log or of which one cannot be sure that it actually happened in reality. This indicates the need for objective measures that are directly learned from the event data, without the need for an intermediate model.

Moreover, in the field of business process improvement (BPI), some authors claim that it is not always clear from literature how existing techniques are used to support the process of business process improvement. Therefore, it was stated in Section 4.2 that, to tackle this gap, a set of suitable metrics should be introduced to gain insights from event logs in an objective and parameter-free procedure. As was explained in Chapter 1, four types of artifacts have been presented in design science literature [94]. Here, the first type, which are the constructs, will be employed to develop the artifact, as it represents a newly defined language and terminology with definitions to express the problem and its potential solutions.

The next subsection will summarise all artifact requirements that can be identified from the literature review and from the business process expert interviews that were analysed above.

4.4.2 The requirements

Johannesson and Perjons [78] state that two types of requirements exist, which are functional and non-functional requirements. Functional requirements specify the functions that the artifact should fulfil, e.g., the reservation system of a library should be able to tell if a book is available, if it is with a reader, or if it is reserved for someone. Non-functional requirements define the structure and capabilities of the artifact, e.g., the reservation system of the library should be able to respond to each information request within three seconds. Van Vliet [168] adds two other types of requirements, which are global conditions, that concern the end product, and project issues, that concern the entire project. Examples of these two types are the requirement that the reservation system will be used by all staff members and visitors of the library, and the requirement that the project that involves the creation of the entire reservation system should be finished on the 1st of May 2018, respectively.

These requirement types can be associated with the field of requirements engineering, which entails different activities in the definition, documentation, and maintenance of requirements for a newly designed product or artifact, and which finds its roots in the field of software engineering [179]. Requirements engineering can be defined as a method to define the requirement specifications that represent the needs of different stakeholders [19]. Many authors define the activities within requirements engineering in a different way. However, they mostly include eliciting, modelling and analysing, communicating, agreeing, and evolving requirements [113]. A detailed overview of the methods and techniques of requirement engineering can be found in, among others, Kotonya and Sommerville [85], Van Vliet [168], and Wieringa [179]. From the literature overview provided by these authors, it can be stated that requirement specifications for an artifact should be *correct, unambiguous, complete, consistent, ranked for importance or stability, verifiable, modifiable, and traceable.* Multiple standards exist to document requirements, however, more important is it to adhere to these constraints when a structure is chosen [168].

Based on these constraints, the requirements of the event log-based metrics that are developed in this dissertation can be described as follows.

Requirement 1 (functional). The event log-based metrics should measure both the general aspects of an event log and the more specific measures concerning the operational excellence field, which were stated to be useful event log knowledge insights in the analysis requirements of the operational excellence methodologies presented in Chapter 3. Both from literature and in practice it was found that different indicators for measuring the level of performance of a business process already exist. However, most of these existing measures are model-based, yet some are event data-based. An overview of the existing metrics in literature was provided in Chapter 3. To structure the list of metrics, four categories of process performance measures, which are often referred to as the Devil's Quadrangle, were identified in literature. These are time, cost, quality, and flexibility or structuredness. These findings from literature are fully supported by the findings from practice.

In this research, we will focus on the dimensions time and structuredness, as these are both shown to be direct indicators of different types of waste and other indicators of operational excellence in Chapter 2. Moreover, all measures that were found to be useful applications of event log knowledge in operational excellence are concerned with the time or structuredness dimension. The concept structuredness is chosen above flexibility, because we want to measure how structured -and not how flexible- the behaviour in the event log is. However, this is mainly a matter of terminology as the content of these two concepts is used interchangeably in literature and in practice. One could argue that structuredness refers to the amount of variation that is allowed in a negative way, as variation should be removed from a process, while flexibility shows the amount of variation within a process in a positive way, as the process allows many different forms of behaviour to, e.g., customise the end product. Although structuredness has been defined by vanden Broucke [169] as a quality metric to measure the ease of interpretation of a process model, we will define structuredness as *the level of variation in the event log*, which should be reduced according to the principles of operational excellence. For the other two categories of performance measurement, no metrics will be presented in this research. Quality is a category which is hard to measure as it is based on the outcome of the process, not the event log data of the process itself. And the cost of a process can not be defined as objective data as it depends on the costing model of the business process under consideration, which is even not always present and also not always granular enough. Moreover, it was stated before that quality and cost depend on the other two dimensions, time and structuredness.

Finally, an overview of the required measures which was found in Chapter 3, and which is supported by the findings from the interviews, is repeated here. The event log-based metrics should be able to identify:

- general aspects of the event log, such as
  - activities,
  - resources,
  - cases,
  - and patterns (or traces),
- and more specific measures concerning the operational excellence field, such as
  - process flow,
  - throughput time and service time,
  - waiting time,

- bottleneck activities and resources,
- frequency of activities,
- frequency of patterns,
- number of patterns,
- activities executed in batch and batch sizes,
- rework,
- and resource variation.

Requirement 2 (functional). The event log-based metrics should measure only one dimension or level of analysis, in order to remain comprehensible. Therefore, each of the measures should be defined on *different levels of analysis* in order to get insights into the different degrees of granularity of a business process. From the interviews it was found that not just one level of analysis should be considered to get a notion of the performance of a business process. Most analyses executed today are based on parts of a business process, which is mainly the part which the business process analyst or the person who asks for the analysis is working on, ignoring the rest of the process. Therefore, different levels of analysis were found to be interesting in a performance measurement exercise. These are:

- the complete end-to-end process,
- the level of specific and separate activities people are working on,
- the different paths the process can follow, which are the process patterns or traces,
- the case for which the process is executed, such as the order of a specific customer,
- the resources executing the process activities, such as the employees or the machines,
- and a combination of the aforementioned levels of analysis.

Requirement 3 (non-functional). The event log-based metrics should contain *clear descriptions of the measure itself, the requirements for event data, and the underlying calculation*. Both from literature and from the interviews described above it could be concluded that existing measures often lack specific descriptions on how the metrics are calculated and which data format or data preparation steps are required before the metrics can be applied. In order to be straightforward and objectively interpretable, the event log-based metrics developed here should be as specific as possible in terms of data input and calculation method.

Requirement 4 (non-functional). The event log-based metrics should be complemented with a *suitable visual representation.* Introducing more objective and parameter-free performance measures on different levels of analysis to improve the operational excellence of a business process would not be sufficient according to the observations that were done. For business people to understand the business processes in their company and to understand the results from the measures applied to the processes, the measures should be created on a rather low and understandable level, and visually accessible. Therefore, the metrics that are introduced should be enriched with suitable visualisations. Further, the used *terminology* should be adapted to the correct level of interpretation in order for business people to understand.

4.5 Conclusion

This chapter provides an overview of the requirements that emerged from the literature review that was provided in Chapter 2 and Chapter 3 and from the interviews that were described in the chapter at hand. Given the potential of process mining in the field of operational excellence and its recognition as a key challenge for process mining research in the Process Mining Manifesto [160], further research is required on this topic. Existing research efforts seem to be limited and it is not clear from literature how existing techniques are used to support the process of business process performance measurement.

In order to acknowledge this lack, this dissertation will introduce the concept of log-based process metrics for which the requirements are analysed in this chapter. Both from literature and practice an exact overlap was found between the different categories of process performance measures that should be focused on in a business process improvement project. These categories are (i) time, (ii) cost, (iii) quality, and (iv) structuredness. Moreover, the aspects of a process that have been recognised to cause waste in a process are, among others, rework, iterations, batch processing, waiting, delays, and non-value-adding activities. The measurements should also be executed on different levels of analysis in order to provide a realistic view on the underlying process. Therefore, the levels of analysis that arose from the interviews range from the complete end-to-end process to the specific combination of a resource executing a specific activity. Moreover, the developed metrics should contain clear descriptions of the measure itself, the required data, and the calculation method that is used in the metric. Finally, the artifacts that are created in order to overcome the lack should be understandable for business people. This can be realised by adding suitable visual representations and by adding a translation of technical concepts to concepts that are interpretable by business people.

Chapter 5

Log-based process metrics

5.1 Introduction

Process mining is intended to detect strategic insights from business processes by extracting valuable information from event logs [155], as was shown in Chapter 3. Next to this, it has also been suggested that process mining can be used to support operational excellence in companies [155, 160, 161].

However, process mining literature is primarily focused on the discovery of comprehensible process models that best capture the underlying behaviour in event logs. Many discovery algorithms have been introduced [42, 169] and each has its specific assumptions. Consequently, the resulting models a) aggregate information, based on algorithm-specific assumptions, and b) transform information into a simplified representation. Both characteristics, which are valuable in certain, different contexts, suffer from the inability to describe the behaviour that is inherent to the event log objectively and in a detailed fashion.

The goal of this chapter[1] (Figure 5.1) is to develop an artifact that fulfils the requirements which were presented in the previous chapter, to extract useful knowledge insights from event log data to support operational excellence techniques. An overview of useful event log-based metrics that provide unbiased and algorithm-agnostic information of the present process behaviour, without the need to first discover a model,

[1]This chapter is based on *Swennen, M., Janssenswillen, G., Jans, M., Depaire, B., Vanhoof, K., 2015. Capturing process behavior with log-based process metrics. CEUR Workshop Proceedings 1527, 141-144* [148] and *Swennen, M., Martin, N., Janssenswillen, G., Jans, M., Depaire, B., Caris, A., Vanhoof, K., 2016. Capturing resource behaviour from event logs. CEUR Workshop Proceedings 1757, 130-134.* [149].

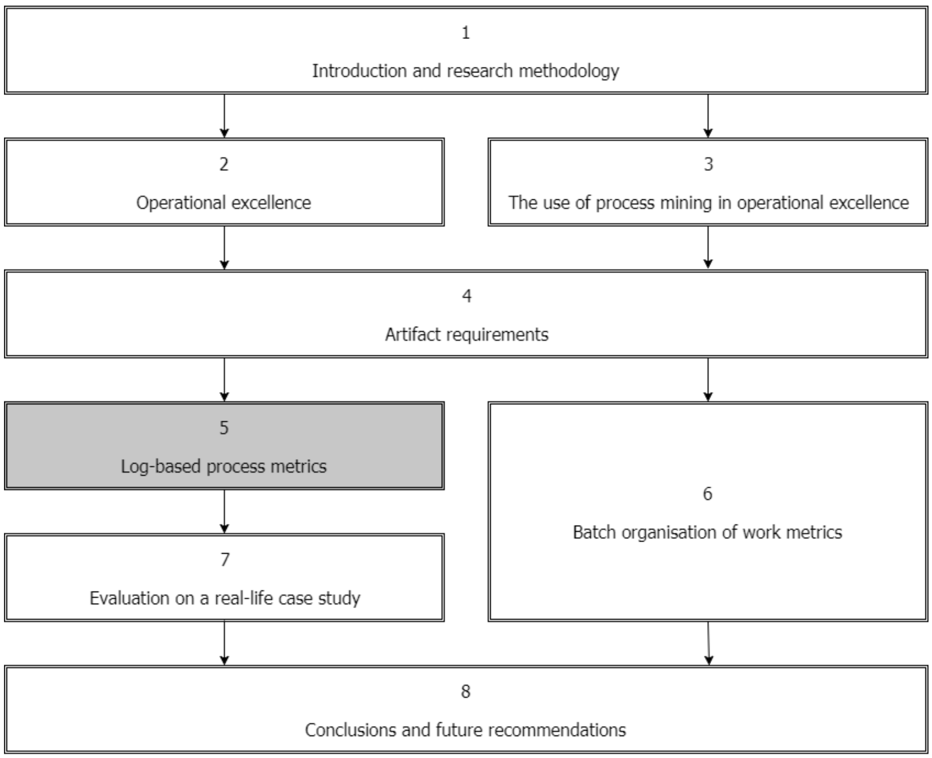

Figure 5.1: Outline of the thesis - Chapter 5.

is therefore presented. This can be a starting point of the process analysis and process redesign, as presented in the BPM-lifecycle in Chapter 3 [47]. Additionally, the constructed metrics also serve as a means to objectively compare different event logs in terms of time-related and variance aspects.

This chapter starts with a short overview of related work in Section 5.2. Next, the event log requirements for the developed metrics to be effective are presented in Section 5.3. In Section 5.4, a running example is presented, followed by an overview and description of the different event log-based metrics in Section 5.5. A dashboard that visualises the metric results is introduced in Section 5.7 and all characteristics of the newly developed artifact are discussed in Section 5.8. Finally, Section 5.9 presents the conclusions and future work.

5.2 Related work

An overview of existing performance measures defined by Anupindi et al. [10], Dumas et al. [47], and Kis et al. [82] was presented in Chapter 3, where they are categorised based on the four performance measurement dimensions of the Devil's Quadrangle: time, cost, quality, and flexibility or structuredness. Also other research fields such as operations and productions management use this categorisation to define and classify performance measurement indicators. An overview of existing performance measures in the field op operations management can be found in Bhagwat and Sharma [17], De Toni and Tonchia [40], and Neely et al. [111]. Although many metrics already exist, only few authors specifically clarify or describe how these measures are calculated and which data conditions are required.

Moreover, as was already discussed in Chapter 3, existing metrics in the field of process mining are primarily comparing the behaviour in an event log with a process model. An overview of existing model-log metrics is presented by De Weerdt et al. [41]. These metrics are divided into four categories which are recall, specificity, precision, and generality. Although these categories resemble the well-known conformance checking dimensions fitness, precision, generalization, and simplicity [132], they are not completely the same. A similar distinction is also presented by vanden Broucke [169], who defines fitness, precision, and generalization as accuracy metrics, and simplicity and structuredness as comprehensibility metrics. Based on the distinction between the event log, the discovered model, and the underlying system presented by Buijs [22], more attention goes to the fact that the underlying system should not be ignored. Janssenswillen et al. [76] state that most existing metrics measure the quality of a process model with respect to the event log it was discovered from and not the underlying system, ignoring the fact that event logs only contain a portion of the complete reality. Therefore, the authors present alternative quality dimensions to measure the distance between the event log and the discovered model, and between the process model and the underlying system. However, no metrics have been defined for analysing the event log behaviour.

The concept of log-based metrics was introduced in the fuzzy mining algorithm by Günther and van der Aalst [63]. Next to this, Ribeiro et al. [129] define a framework to determine which is the best process discovery algorithm. To that end, *features* are defined as numerical characteristics of event logs to capture the distance or diversity between two event logs. Also *measures* are presented, which are used to evaluate the performance and quality of discovery techniques. The categories in which the measures can be divided are again simplicity, fitness, precision, generalization, and an extra category that is added to these four, which is performance. In comparison to these findings, in our approach, we present log-based process metrics that are independent from the process discovery algorithms or process models discovered with these discovery algorithms. Finally, existing metrics that extract knowledge from event logs about resources are presented by Pika et al. [119], who focus on the behaviour of resources within a process, together with the changes of their behaviour over time. These metrics concern different dimensions of the behaviour of resources, such as how employees work together, how productive they are, which activities they prefer doing, which activities they are able to perform, and which activities they are actually executing. Before introducing the developed metrics, the requirements for the underlying event logs are described in the next section.

5.3 Event log requirements

The metrics introduced in this chapter require an event log, which is composed of events related to a particular case and activity. An event log describes one specific process, which consists of a set of activities. An instantiation of the process is called a case and consists of one or more instantiations of activities, which are called activity instances. An activity instance in turn consists of one or more events, which are atomic registrations of actions. Preferably, each activity instance contains a start and an end event, which are both performed by the same resource, and which both contain a timestamp.

Building on the notation used by van der Aalst [159], the event log characteristics that are required to use the log-based process metrics presented in this chapter can be outlined as follows (Figure 5.2):

Definition 1 (Activity)**.** *An activity A is a logical unit of work that is carried out as a single whole. We define $\mathcal{A}$ as the set of all activities within the process.*

Definition 2 (Activity lifecycle)**.** *Each activity has a lifecycle which can be defined as $L = (S, T)$ such that S represents the set of all possible states s and $T \subset S \times S$, where T represents the set of allowed state transitions. Note that by definition each activity can have a different lifecycle model. For example, some activities may be completed or aborted once started and do not have a state 'suspended' in their lifecycle. In this dissertation, we assume that all activities have a very basic lifecycle such that $S = \{\text{start}, \text{complete}\}$ and $T = \{(\text{start}, \text{complete})\}$.*

Definition 3 (Resource)**.** *A resource r is responsible/required for the execution of an activity. We define $\mathcal{R}$ as the set of all resources involved in the process.*

Definition 4 (Event). *An event $e = (A, t, s, r)$ represents an atomic moment in time $t \in \mathbb{R}$ representing a specific action which changes the state of an instance of activity A to state s by resource r. Let $\mathcal{E}$ be the universe of events, then $\#_a : \mathcal{E} \to \mathcal{A}$, $\#_t : \mathcal{E} \to \mathbb{R}$, $\#_s : \mathcal{E} \to S$, $\#_r : \mathcal{E} \to \mathcal{R}$ represent functions that map the event into its corresponding activity, timestamp, updated state, and resource.*

Definition 5 (Activity instance). *An activity instance $a = \{A, \mathbb{E}_a\}$ is the actual execution of an activity A within a process comprising a set of all related events $\mathbb{E}_a$ which change the life cycle state of the activity instance. Let $\mathfrak{A}$ represent the universe of all activity instances and $\#'_a : \mathfrak{A} \to \mathcal{A}$ a function that maps each activity instance into its corresponding activity, then we know that if $e_i, e_j \in \mathbb{E}_a \Rightarrow \#_a(e_i) = \#_a(e_j) = \#'_a(a)$. Note that this only holds in one direction, as events related to the same activity can belong to different activity instances if the activity is executed multiple times in a process execution.*

Definition 6 (Case). *A case $c \subset \mathfrak{A}$ refers to a set of activity instances which represent the work performed for a specific process execution (process instance), with $\mathcal{C}$ denoting the universe of cases. It is common to define a partial ordering relation which allows the case to be written as a sequence $\langle a_1, \ldots, a_n \rangle$ of activity instances. The most common partial ordering relation $\leq_{\text{start}} = \{(a_i, a_j) \in c \times c | e_i \in \mathbb{E}_{a_i}, e_j \in \mathbb{E}_{a_j}, \#_s(e_i) = \#_s(e_j) = \text{start}, \#_t(e_i) \leq \#_t(e_j)\}$ orders activity instances based on the timestamp of their starting event.*

Definition 7 (Event log). *An event log $L \subset \mathcal{C}$ is a set of cases, which were all generated by the same process. We will denote the number of cases in an event log by $|L|_C$ and the number of activities in an event log by $|L|_A$.*

Definition 8 (Trace). *A trace $T \in \mathcal{A}^*$ is a finite sequence of activities, with $\mathcal{A}^*$ denoting the set of all finite sequences over $\mathcal{A}$. A trace $T = \langle A_1, \ldots, A_n \rangle$ is typically the result of applying the function $t^\rho : L \to \mathcal{A}^*$ on a specific case $c \in L$ given a specified partial ordering relation ρ. This function t^ρ maps each case c into a $T = \#'_a(\langle a_1, \ldots, a_n \rangle) = \langle \#'_a(a_1), \ldots, \#'_a(a_n) \rangle$ where $\langle a_1, \ldots, a_n \rangle$ is the activity sequence instance of c defined by partial order ρ. Note that different cases can have the same trace.*

Based on these definitions, we state that each row in the event log should contain at least five different pieces of information, which are a case identifier, an activity label, a timestamp, a resource identifier, and a transactional lifecycle identifier which indicates the status of the event (e.g., start, complete,...). Additionally, each row may

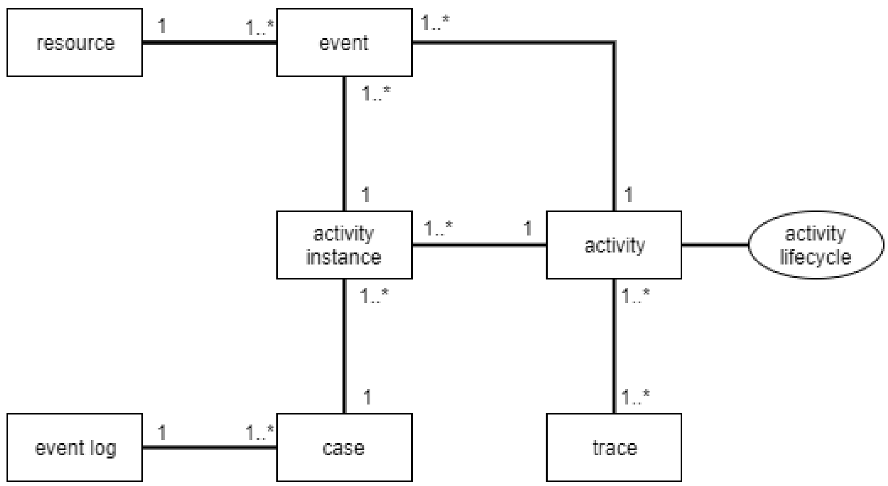

Figure 5.2: Conceptual representation of an event log.

be complemented with any other custom event attribute such as costs. As the data that is stored by organisations is nowadays still lacking many of these attributes, data transformations and manual actions may be required before the event log is ready to be used for the analyses.

5.4 Running example

Throughout this chapter, a running example will be used for illustrative purposes. The running example, which is shown in Figure 5.3, consists of an event log containing 76 activities distributed over 12 cases, C1 to C12, which can be seen as customers. Six different resources, R1 to R6, are employed to execute for each customer a series of activities, consisting of six possible activities, A, B, C, D, E, and F, over a period of almost a month. For each executed activity instance, a start and an end event are recorded. In Table 5.1 this information is summarised. Table 5.2 provides a sample of ten lines in the running example event log, for which the representation is slightly transformed for readability.

5.5 Log-based process metrics

In this section, the process metrics to identify and quantify the behaviour of a process are provided. As was stated in the first artifact requirement defined in Chapter 4, we will focus on the dimensions time and structuredness, as these are both shown to

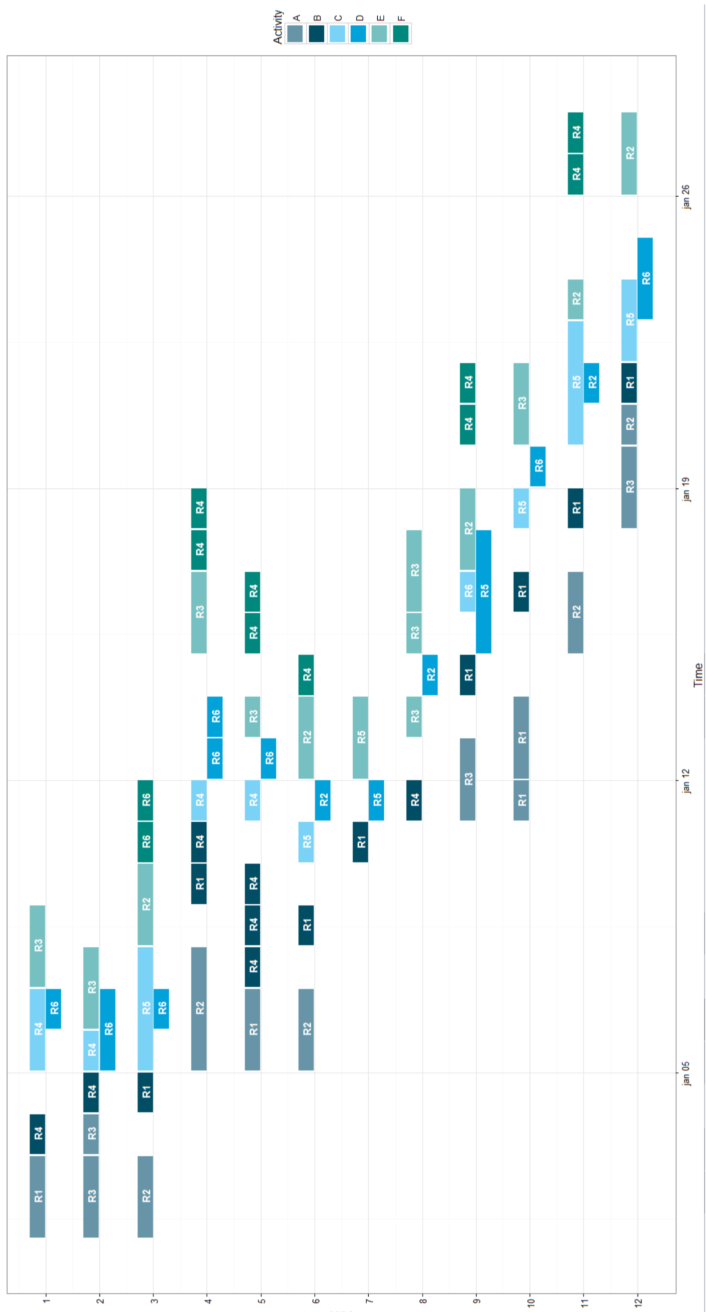

Figure 5.3: The running example event log, which consists of 12 cases.

Table 5.1: Summary of the running example event log.

Number of events	152
Number of activities	6
Number of resources	6
Number of cases	12
Number of traces	9
Average trace length	12.67 events
Start timestamp	2015-01-01 01:23:45
End timestamp	2015-01-28 01:23:45

Table 5.2: Sample from the running example event log.

case ID	activity ID	start date	start timestamp	complete date	complete timestamp	resource
C1	A	1/01/2015	1:23:45	3/01/2015	1:23:45	R1
C1	B	3/01/2015	1:23:45	4/01/2015	1:23:45	R4
C1	C	5/01/2015	1:23:45	7/01/2015	1:23:45	R4
C1	D	6/01/2015	1:23:45	7/01/2015	1:23:45	R6
C1	E	7/01/2015	1:23:45	9/01/2015	1:23:45	R3
C2	A	1/01/2015	1:23:45	3/01/2015	1:23:45	R3
C2	A	3/01/2015	1:23:45	4/01/2015	1:23:45	R3
C2	B	4/01/2015	1:23:45	5/01/2015	1:23:45	R4
C2	C	5/01/2015	1:23:45	6/01/2015	1:23:45	R4
C2	D	5/01/2015	1:23:45	7/01/2015	1:23:45	R6
...	...	...	...	...	...	...

be indicators of different types of waste and other aspects of operational excellence. Structuredness is defined here as the level of variation in the event log. Next to metrics depicting the variance in a process, also metrics concerning rework and resources, which are the people or machines executing activities in the process, will be presented for the structuredness dimension. Concerning the resource metrics, which represent the organisational aspect of a business process, we will elaborate in Chapter 6 on the concept of batch processing, in which resources execute activities of two or more cases at the same or almost the same time.

Figure 5.4: Overview of the developed log-based process metrics.

Figure 5.4 provides a visual overview of all metrics that have been developed with their accompanying levels of analysis. An elaborate list of all metrics is included in Table B.1 and Table B.2 in Appendix B. The developed metrics cover the list of required measures that was presented in the first artifact requirement in Chapter 4. According to the second artifact requirement that was presented, and following the study on model-log evaluation metrics presented by De Weerdt et al. [42], only one dimension or level of analysis should be measured by each metric in order to remain comprehensible. Building on the different feature scopes presented by Ribeiro et al. [129], we will assign each metric to one of the following levels of analysis: (i) the log level, which represents the entire event log, (ii) the trace level, representing characteristics of sequences of activities, (iii) the case level, in which all characteristics for a certain process execution are considered, (iv) the activity level, representing characteristics of the activity types, aggregated over the entire log, (v) the resource level, which represents characteristics of the resources executing the activities, and (vi) the resource-activity level, in which all resource-activity combinations are considered. Note that not all metrics are measurable on each of these levels, as not all combinations are logical or insightful. Other possible analysis levels, that are out of the scope of this research, are multiple traces (characteristics on dependencies between traces) and multiple activities (characteristics on dependencies between activities within a trace). All metrics have been defined based on the findings from literature and based on the interviews with people from industry which are described in Chapter 4.

For each of the metrics, three subjects can be described. Firstly, a definition of the metric will be given in order to avoid misinterpretation between different groups of people. The definitions will be clear and simple, containing terminology that is also understandable for business people. Secondly, the metric will be applied to the running example that was presented in Section 5.4. Thirdly and finally, some metrics will require some extra information about items one should pay attention to when applying the metric, or about shortcomings of the metrics that are not dealt with within this dissertation. Next to their application to an artificial event log, all metrics will be evaluated by applying them to a real-life event log of a Belgian utilities company, which will be described in Chapter 7.

5.5.1 Time metrics

5.5.1.1 Duration

Metrics measuring the duration provide summary statistics concerning the throughput time of cases. The throughput time of a case is defined as the time between the start of the first activity and the completion of the last activity executed in the case. These metrics can be performed on the level of the log as well as the level of specific cases and traces, and was defined as one of the required measures to identify concepts such as critical lead times and bottlenecks in the field of operational excellence. For comprehensibility, the metric will first be explained at the case level, followed by the log and the trace level. For the running example, which was presented in Section 5.4, all durations are expressed in hours. However, other time expressions such as days or weeks, which can also be calculated with the following metrics, can be of interest.

- *Throughput time - case level.* The throughput time of a case is the total duration of the case, or the difference between the timestamp of the end event of the last activity and the timestamp of the start event of the first activity in the case. Possible waiting time is therefore also included in this calculation. In Table 5.3, a list of all cases from the running example presented in Figure 5.3, with their accompanying throughput time, is provided. For example, case C4 has the longest throughput time, which is 336 hours (or 28 days).

Table 5.3: Throughput time, processing time and waiting time (case level) applied to the running example. The cases are sorted on their throughput time.

case	throughput time (in hours)	processing time (in hours)	waiting time (in hours)
C4	336	288	48
C11	312	240	96
C5	288	240	48
C3	264	264	24
C9	264	264	48
C10	264	192	72
C12	240	240	24
C6	240	192	48
C1	192	192	24
C2	168	216	0
C8	168	144	24
C7	96	96	0

An interesting additional measure that can be calculated related to the throughput time of the cases in an event log is *the number of pending cases.* Pending cases are cases that did not finish properly at the time the analysis is performed or at the time the data is extracted from the information system. However, this is very process-specific as the process analyst should know which activities define the end of a case as this can differ throughout the process. For example, case C1 and C2 in the running example end with activity E, while case C3 and C4 end with activity F. As we are assuming for the running example that all cases ended properly, this metric concerning pending cases will not provide any added value for this example. However, in real-life event logs it will be more common to find cases that end with activities that do not imply the end of a process, or cases that are pending after only certain activities have been executed. This could be an indication for companies to analyse these cases more thoroughly, as they probably include one or more bottleneck activities, which may prevent a case from proper completion.

- *Throughput time - log level.* Providing the throughput time or duration only on a case level would possibly not increase the process owner's understanding of the underlying behaviour. Therefore, the summary statistics of these throughput times are presented as a metric, to describe the throughput time of a case in an aggregated fashion. The summary statistics that will be calculated for this and other metrics are the minimum, first quartile, median, mean, third quartile, maximum, standard deviation, interquartile range (iqr), and the total. The average throughput time of all cases in the running example is 236 hours (9.83 days), the standard deviation is 68.522, and the median 252 hours. The shortest throughput time is 96 hours (case 7) and the longest is 336 hours (case 4). Table 5.4 shows the results of this metric applied to the running example.

Table 5.4: Throughput time (log level) applied to the running example.

min	q1	median	mean	q3	max	st.dev.	iqr	tot
96	186	252	236	270	336	68.522	84	2832

- *Throughput time - trace level.* Instead of looking at all cases in the log, it can be interesting to analyse the different process variants or traces in the log. Dividing an event log in homogeneous subsets of traces was presented by Song et al. [143] to overcome the difficulty of analysing large, unstructured processes. The number of traces in the log will be explained further on as a metric of structuredness. As a time-related metric we propose the throughput time which can be calculated for each trace. One example of a trace in the example event log is A,A,B,C,D,E, which is executed in 3 cases, i.e., C2, C10, and C12, with individual throughput times of 168, 264, and 240 hours, respectively. This corresponds to an average throughput time for this trace of 224 hours (9.33 days), a standard deviation of 49.96, and a median of 240. Table 5.5 shows the summary statistics of the throughput time of all traces in the running example. In reality, event logs mostly contain hundreds or even thousands of traces. If this is the case, the metric will, by default, show the top 10 most frequent traces.

5.5.1.2 Processing time

In contrast to the throughput time of the cases in an event log, the metrics concerning the active time or the actual processing time provide summary statistics on the pro-

Table 5.5: Throughput time (trace level) applied to the running example. The traces are sorted on their relative frequency.

trace	relative frequency	min	q1	median	mean	q3	max	st.dev.	iqr	tot
A,A,B,C,D,E	0.250	168	204	240	224	252	264	49.960	48	672
A,B,C,D,E,F,F	0.167	264	276	288	288	300	312	33.941	24	576
A,B,B,B,C,D,E,F,F	0.083	288	288	288	288	288	288	NA	0	288
A,B,B,C,D,D,E,F,F	0.083	336	336	336	336	336	336	NA	0	336
A,B,C,D,E	0.083	192	192	192	192	192	192	NA	0	192
A,B,C,D,E,F	0.083	240	240	240	240	240	240	NA	0	240
A,B,D,C,E,F,F	0.083	264	264	264	264	264	264	NA	0	264
B,D,E	0.083	96	96	96	96	96	96	NA	0	96
B,E,D,E,E	0.083	168	168	168	168	168	168	NA	0	168

cessing time of activities. This metric is developed on the level of the entire event log, the specific cases and traces, the activities, and the resource-activity combinations. Next to insights into the distribution of the processing time of each task, this information can also be used for predicting execution times of running process instances, which is helpful for process monitoring.

- *Processing time - case level.* The actual processing time in a case is the sum of the processing time of all activities that are executed within this case. in Table 5.3, a list of all cases with their accompanying processing time, next to their throughput time, is provided. For example, case C4 has a throughput time of 336 hours, and a processing time of 288 hours. Cases C5, C11, and C12 all three have an actual processing time of 240 hours (10 days). In some cases, no difference between throughput time and processing time is found, while in other cases no activity is tracked for more than a quarter of the throughput time.

- *Processing time - log level.* Next to this, the actual processing time of the entire event log is the sum of the actual processing time of all activities that are executed in the event log. However, it is not useful to add all processing times of different activities executed for different cases (e.g., customers). Therefore, this metric calculates the summary statistics of the actual processing time per case, summarised over the entire event log. Table 5.6 shows that the average processing time of the cases in the running example is 214 hours (8.9 days) with a standard deviation of 54.593. Compared to the average throughput time of all cases in the event log, this is 22 hours or almost one day less, indicating that on average for almost 10 % of the complete throughput time of a case nothing is happening (or nothing is being logged by the system).

Table 5.6: Processing time (log level) applied to the running example.

min	q1	median	mean	q3	max	st.dev.	iqr	tot
96	192	228	214	246	288	54.593	54	2568

- *Processing time - trace level.* On the level of the traces, the summary statistics as provided above can be calculated for each possible sequence of activities that appears in the event log. Table 5.7 shows the results of this metric for each trace appearing in the running example.

Table 5.7: Processing time (trace level) applied to the running example. The traces are sorted on their relative frequency.

trace	relative frequency	min	q1	median	mean	q3	max	st.dev.	iqr	tot
A,A,B,C,D,E	0.250	192	204	216	216	228	240	24.000	24	648
A,B,C,D,E,F,F	0.167	240	246	252	252	258	264	16.971	12	504
A,B,B,B,C,D,E,F,F	0.083	240	240	240	240	240	240	NA	0	240
A,B,B,C,D,D,E,F,F	0.083	288	288	288	288	288	288	NA	0	288
A,B,C,D,E	0.083	192	192	192	192	192	192	NA	0	192
A,B,C,D,E,F	0.083	192	192	192	192	192	192	NA	0	192
A,B,D,C,E,F,F	0.083	264	264	264	264	264	264	NA	0	264
B,D,E	0.083	96	96	96	96	96	96	NA	0	96
B,E,D,E,E	0.083	144	144	144	144	144	144	NA	0	144

- *Processing time - activity level.* Next, if both a start and end timestamp are provided for each activity instance, the duration can also be calculated on the level of each activity. For each activity, an overview of the average processing time, or the service time, of this activity can be of interest. For example, concerning all 13 occurrences of activity A in our event log, the average duration is 44.308 hours (1.85 days). Table 5.8 provides an overview of the summary statistics per activity, sorted on their relative frequency in the event log.

Table 5.8: Processing time (activity level) applied to the running example. The activities are sorted on their average processing time.

activity	relative frequency	min	q1	median	mean	q3	max	st.dev.	iqr	tot
A	0.171	24	48	48	44.308	48	72	13.313	0	576
E	0.184	24	30	48	41.143	48	48	11.251	18	576
C	0.132	24	24	24	38.400	48	72	20.239	24	384
D	0.171	24	24	24	31.385	24	72	15.130	0	408
B	0.197	24	24	24	24.000	24	24	0	0	360
F	0.145	24	24	24	24.000	24	24	0	0	264

Building from this, and based on the findings in literature, a *bottleneck indicator* would be a useful metric to measure waste. In a process, a bottleneck is an activity that obstructs other activities to be executed properly and determines the continuation of the entire process [103]. According to the theory of constraints, introduced in Chapter 2, weak links or constraints should be eliminated from a process [60]. Based on this theory is the *drum-buffer-rope method*, which is also explained in Chapter 2. A bottleneck indicator could be identified by searching for the activity in the process that has the longest duration compared to the other activities in the process. In case C3, for example, activity C takes more time to be executed than all other activity executions.

- *Processing time - resource level.* We can also look at the processing time per case on the level of each separate resource. This way, a company gets an overview of the amount of time each resource spends on a case and which resources spend more time on cases than others. Table 5.9 provides an overview of the summary statistics of the processing time per resource, sorted on the average processing time spent per case. Resource R2 spends on average the longest on a case, which is 44 hours.

- *Processing time - resource-activity level.* On the resource-activity level, finally, we can have a look at the efficiency of resources by looking at the combination of each resource with each activity. This can be more insightful when we want

Table 5.9: Processing time (resource level) applied to the running example. The resources are sorted on their average processing time per case.

resource	min	q1	median	mean	q3	max	st.dev.	iqr	tot
R2	0	0	36	44	78	120	45.575	78	528
R4	0	0	36	42	54	144	44.754	54	504
R3	0	0	48	40	48	120	38.744	48	480
R5	0	0	24	32	72	72	32.897	72	384
R1	0	24	24	30	30	96	25.327	6	360
R6	0	0	24	26	48	72	23.909	48	312

to compare different resources or different activities with each other. Mostly, activities are so diverse that comparing them over the entire event log is not helpful. The same is true for resources. This metric provides an overview of the summary statistics of the processing time per resource-activity combination that occurs in the entire event log. As an example, in Table 5.10 the overview of the summary statistics of the processing time per resource-activity combination for resource R1 is given. As can be seen, R1 is only responsible for activities A and B in the running example event log. Similarly, the summary statistics for all other resource-activity combinations can be calculated and analysed.

Table 5.10: Processing time (resource-activity level) applied to the running example. Only resource-activity combinations with resource R1 are shown.

resource	activity	min	q1	median	mean	q3	max	st.dev.	iqr	tot
R1	A	0	0	0	14	12	72	26.007	12	168
R1	B	0	0	24	16	24	24	11.817	24	192
R1	C	0	0	0	0	0	0	0	0	0
R1	D	0	0	0	0	0	0	0	0	0
R1	E	0	0	0	0	0	0	0	0	0
R1	F	0	0	0	0	0	0	0	0	0

5.5.1.3 Waiting time

In contrast to the actual processing time, the waiting time in an event log can be seen as a direct indicator of waste according to the principles of lean management [182]. Moreover, it has been shown to be a required metric within each of the operational excellence methodologies analysed in Chapter 2 and Chapter 3. The waiting time of a certain activity is here defined as the time that this task is waiting for a resource to start the task, because the resource is possibly still occupied with other tasks. The total waiting time in a case is calculated by taking the sum of all time in the case that is not being used. We already depicted the difference between the processing time and the throughput time of cases in the previous metrics. However, for these metrics concerning the waiting time, we should take into account that waiting time can also happen before the start of the first activity in a case, which is probably never captured in an event log.

- *Waiting time - case level.* Firstly, on the level of the specific cases in the event log, this metric provides an overview of the total waiting time per case. For example, the total waiting time for case C1 in the running example is 1 day or 24 hours (see Table 5.3).

- *Waiting time - log level.* Aggregated on the level of the entire event log, the waiting time metric provides an overview of summary statistics of the waiting time per case, aggregated over the entire event log. In Table 5.11 we can see that a case in the running example contains on average 45.6 hours or almost 2 days of waiting time, with a minimum of 24 hours and a maximum of 96 hours.

Table 5.11: Waiting time (log level) applied to the running example.

min	q1	median	mean	q3	max	st.dev.	iqr
24	24	48	45.6	48	96	23.87	24

- *Waiting time - trace level.* On the level of the different traces that occur in the event log, the waiting time metric provides an overview of the summary statistics of the waiting time for each trace in the event log. Trace A,B,C,D,E,F,F, for example, which appears in case C3 and case C6 in the running example, has an average waiting time of 60 hours and a standard deviation of 50.912.

- *Waiting time - resource level.* Finally, the metric can also be of interest on the level of the resources, to get an insight in the "idle time" of each resource within a business process. For example, resource R1 has a total waiting time of 144 hours (6 days) as this is the amount of time he or she is not active within the event log. Probably the resource is working on another project which is not included in this process or the resource can be taking a break during lunch time, for example. The total waiting time of each resource in the running example can be found in Table 5.12. Organisations could use these values as an input for the evaluation of their resources. Or they could have a look at these values to find out if the waiting time is much higher for one of the resources, or if it is similar for all resources, implying the cause is external to the resources.

Table 5.12: Waiting time (resource level) applied to the running example. The resources are sorted on their waiting time.

Resource	waiting time
R6	240
R4	168
R2	168
R1	144
R3	120
R5	96

Instead of looking at the total waiting time within a trace or case, it can also be interesting to narrow down to the waiting time of a specific activity, which is the time between the arrival of the activity in the case and the start of the execution of this activity. However, as Leemans et al. [88] state, performance measures such as waiting times for specific activities cannot be measured correctly without the presence of a process model. If we, for example, want to calculate the waiting time of activity D in case C9, it is not clear from the event log if this activity instance could start after the end of activity A or after the end of activity B. Information on the concurrency of activities is required to calculate this metric [88], which will be covered as future research.

5.5.2 Structuredness metrics

As was stated earlier, next to the performance aspect of time, variability or structuredness could also be seen as one of the categories of process performance measurement. In the literature review on operational excellence in Chapter 2, we found that variability was one of the key sources of waste within a business process, as it is causing a process to deviate from its desired behaviour. However, as variability or structuredness should not always be eliminated, it is important for a company to learn the type of structuredness that is occurring within the process under analysis. In order to get more insights in this, different classes of metrics concerning the structuredness of a process are developed and presented here. Next to measures identifying the variance within a process, also the notion of rework is analysed, as rework was found to be one of the most significant causes of waste within a process. Finally, also metrics concerning the resources executing the activities within a process are presented.

5.5.2.1 Variance metrics

- *Number of traces.* A first notion of the structuredness or variance in an event log is the number of process variants, or distinct traces, that are recorded in the event log. This metric provides two values, which are the absolute and the relative number of traces that occur in the event log. In order to have a comprehensive metric, the relative number is stated as an average coverage. In the running example event log, nine traces can be observed. These traces were already shown in Table 5.5 where the throughput time was calculated for each distinct trace. The relative number shows that a trace appears on average in 1.33 (of the in total 12) cases in the event log, indicating a rather low level of structuredness (the lower the ratio, the lower the structuredness).

- *Trace length - case level.* This metric provides an overview of the number of activities that occur in each trace. In this metric, instances of an activity, as opposed to the actual activities, are calculated. That way, the number of actual transactions in a trace are calculated. This metric, together with the previous one, can provide the process owner with an indication of the number of process variants within an event log, which was stated to be one of the required event log knowledge concepts in requirement 1 in Chapter 4. Table 5.13 shows the trace length, which is actually the number of activity instances executed in a case, for each case in the running example event log. However, this metric does not add any value other than providing information. Summarising this information on the level of the entire event log or on the level of traces can be of more interest.

- *Trace length - log level.* On the log level, the number of actual activity executions in each trace is calculated and aggregated over the entire event log. This metric shows for the running example that on average 6.33 activities occur per trace with a minimum of 3 and a maximum of 9 activities per trace, as is shown in Table 5.14.

Table 5.13: Trace length (case level) applied to the running example. The cases are sorted on their trace length.

case	trace length
C4	9
C5	9
C3	7
C9	7
C11	7
C2	6
C6	6
C10	6
C12	6
C1	5
C8	5
C7	3

Table 5.14: Trace length (log level) applied to the running example.

min	q1	median	mean	q3	max	st.dev.	iqr
3	5.75	6	6.333	7	9	1.670	1.25

- *Trace length - trace level.* Because the trace length on case level is not very useful in large event logs, this metric shows the number of activity instances executed in each trace. Similar to the trace length on the level of the cases and the entire event log, calculations are done for the occurrences of activities, as opposed to the number of distinct activity types in the trace. In the running example event log, trace A,B,B,C,D,D,E,F,F contains nine activity occurrences of six distinct activities. To make this number relative, an interesting denominator can be the average trace length of the traces that cover a certain percentage of the log, for example 80 %. Because it is not in every log straightforward which traces exactly cover 80 %, the metric should be calculated on a percentage of the traces that can be identified unambiguously (and deviates from 80 % if necessary). Applied to the running example, shown in Table 5.15, the trace length can either be compared to the average trace length of the top 41.67 % of the event log (traces A,A,B,C,D,E and A,B,C,D,E,F,F), or to the average trace length of the entire event log (since we cannot distinguish unambiguously which traces with a trace frequency of 0.083 are included in the 80 %, and which traces are not). This results in a relative number of 1.406 or 1.421 for trace A,B,B,C,D,D,E,F,F, suggesting that this trace is rather long.

Table 5.15: Trace length (trace level) applied to the running example. The traces are sorted on their relative trace frequency.

trace	relative frequency	absolute frequency	relative to top 41.67	relative to top 100
A,A,B,C,D,E	0.250	6	0.938	0.947
A,B,C,D,E,F,F	0.167	7	1.094	1.105
A,B,B,B,C,D,E,F,F	0.083	9	1.406	1.421
A,B,B,C,D,D,E,F,F	0.083	9	1.406	1.421
A,B,C,D,E	0.083	5	0.781	0.789
A,B,C,D,E,F	0.083	6	0.938	0.947
A,B,D,C,E,F,F	0.083	7	1.094	1.105
B,D,E	0.083	3	0.469	0.474
B,E,D,E,E	0.083	5	0.781	0.789

- *Trace coverage - log level.* This metric presents the minimum number of traces that is required to cover a certain percentage, by default 80 %, of the cases. Only the required number of traces, not which traces, is stated, since this is not always straightforward. So in this metric on the level of the entire event log, the number of traces to cover a certain percentage of a log is computed, together with the percentage of traces that is covered. If a tie exists, the two nearest points are returned, which is the case for the running example. In Table 5.15, the traces in the running example are sorted based on their relative frequency. To cover 80 % of the 12 cases in the event log (9.6 cases, rounded to 10), at least 7 traces are required: 1 trace with a relative frequency of 25 %, 1 trace with a relative frequency of 16.7 % and 5 traces with a relative frequency of 8.3 %.

The output of the metric contains therefore two lines. The two most frequent traces together cover 41.7 % of the event log, while all 9 of them cover 100 % of the event log.

- *Trace coverage - case level.* On the case level, for each case the coverage of the corresponding trace can be of interest. This metric therefore provides the absolute and relative trace coverage for each case in the event log under consideration. Table 5.16 shows that the trace that is the most frequent in the running example event log appears in the cases C2, C10, and C12.

Table 5.16: Trace coverage (case level) applied to the running example. The cases are sorted on their absolute trace coverage.

case	trace	absolute trace coverage	relative trace coverage
C2	A,A,B,C,D,E	3	0.250
C10	A,A,B,C,D,E	3	0.250
C12	A,A,B,C,D,E	3	0.250
C3	A,B,C,D,E,F,F	2	0.167
C11	A,B,C,D,E,F,F	2	0.167
C1	A,B,C,D,E	1	0.083
C4	A,B,B,C,D,D,E,F,F	1	0.083
C5	A,B,B,B,C,D,E,F,F	1	0.083
C6	A,B,C,D,E,F	1	0.083
C7	B,D,E	1	0.083
C8	B,E,D,E,E	1	0.083
C9	A,B,D,C,E,F,F	1	0.083

- *Trace coverage - trace level.* Finally, on the level of the traces, the absolute and relative frequency of each trace is returned. For the running example, this is shown in Table 5.17. However, as it is not clear which traces should be assumed to be required to cover 80 % of the event log, the order of the traces with an absolute frequency of 1 should be ignored as it can be in any other order than the one that is shown here.

- *Activity presence.* Another indication of variance can be the presence of the

Table 5.17: Trace coverage (trace level) applied to the running example. The traces are sorted on their absolute frequency.

trace	absolute frequency	relative frequency	cumulative sum
A,A,B,C,D,E	3	0.250	0.250
A,B,C,D,E,F,F	2	0.167	0.417
A,B,C,D,E	1	0.083	0.500
A,B,B,C,D,D,E,F,F	1	0.083	0.583
A,B,B,B,C,D,E,F,F	1	0.083	0.667
A,B,C,D,E,F	1	0.083	0.750
B,D,E	1	0.083	0.833
B,E,D,E,E	1	0.083	0.917
A,B,D,C,E,F,F	1	0.083	1.000

activities in the different cases. This metric shows for each activity the absolute number of cases in which each activity occurs together with its relative presence. In the running example, activities B, D, and E are executed for all customers, while activity F only occurs in half of the cases. This is shown in Table 5.18. This may indicate that activity F is only necessary for a certain type of customers, which could be investigated by including other case attributes in the analysis.

Table 5.18: Activity presence applied to the running example. The activities are sorted on their absolute presence.

activity	absolute presence	relative presence
B	12	1
D	12	1
E	12	1
A	10	0.833
C	10	0.833
F	6	0.500

- *Start activities - case level.* For companies, it can also be of interest to have a clue of which activity is the first that is executed for each customer. On the level of the specific cases in the event log, this metric provides an overview of the start activity of each case. The second column in Table 5.19 shows which are the start activities in the different cases in the running example.

Table 5.19: Start activities and end activities (case level) applied to the running example.

case	start activity	end activity
C1	A	E
C2	A	E
C3	A	F
C4	A	F
C5	A	F
C6	A	F
C7	B	E
C8	B	E
C9	A	F
C10	A	E
C11	A	F
C12	A	E

- *Start activities - log level.* Aggregated on the log level, this metric computes how many distinct activities occur as the first activity in a case, both in an absolute and relative number. The first activity in a case is the one which started the first (oldest start timestamp). In the running example, two out of the six distinct activities, or 33.33 %, are performed as a start activity in the example event log.

- *Start activities - activity level.* This metric calculates for each activity the absolute and relative number of cases that start with an activity instance of this activity. The relative number is calculated as a portion of the number of cases, being the number of "opportunities" that an activity could be the start activity. The cumulative sum is added for an insight in the number of activities required to cover a certain part of the total. For the running example event log, shown in Table 5.20, we find that activity A is a start activity in 10 cases, representing a relative presence of 83.33 % (=10/12). Activity B is the start activity in the remaining 2 cases, which counts for a relative presence of 16.67 %. These two cases are case C7 and case C8, as was shown at the case level.

Table 5.20: Start activities (activity level) applied to the running example.

activity	absolute frequency	relative frequency	cumulative sum
A	10	0.833	0.833
B	2	0.167	1.000

- *Start activities - resource level.* On the level of the distinct resources, an overview of which resources execute the start activity per case can be of interest for a company. This metric calculates for each resource the absolute and relative number of cases that start with an activity instance executed by this resource. Probably this person plays an important role in the communication with the customer as he or she is the initiator of the process instance. Table 5.21 shows that four resources are responsible for executing the start activities of the 12 cases in the running example.

Table 5.21: Start activities (resource level) applied to the running example. The resources are sorted on their relative frequency as a start resource.

resource	absolute frequency	relative frequency	cumulative sum
R1	4	0.333	0.333
R2	4	0.333	0.667
R3	3	0.250	0.917
R4	1	0.083	1.000

- *Start activities - resource-activity level.* Finally, on the resource-activity level, this metric shows for each occurring resource-activity combination in the event log the absolute and relative number of times this resource executes this activity as a start activity in a case. For the running example event log, the results of this metric are given in Table 5.22. The output is sorted on the relative frequency and the resource-activity combinations that do not occur as the first activity in the log are omitted from the table.

Table 5.22: Start activities (resource-activity level) applied to the running example. The resource-activity combinations are sorted on the relative frequency that they include the first activity in a case.

resource	activity	absolute frequency	relative frequency	cumulative sum
R2	A	4	0.333	0.333
R1	A	3	0.250	0.583
R3	A	3	0.250	0.833
R1	B	1	0.083	0.917
R4	B	1	0.083	1.000

- *End activities - case level.* Similar to the metrics concerning start activities, on the level of the specific cases, this metric provides an overview of the end activity of each case. Table 5.19 presented all cases together with its start and end activities.

- *End activities - log level.* Aggregated on the level of the entire event log, this metric shows the absolute and relative number of activities that are the last activity in one or more of the cases. In the running example, two out of the six distinct activities, or 33.33 %, are in one or more of the cases the final activity.

- *End activities - activity level.* This metric calculates for each activity the absolute and relative number of cases that end with this activity type. Similar to the start activities metric, the relative number is calculated as a portion of the number of cases, being the number of "opportunities" that an activity could be the end activity. The cumulative sum is added to have an insight in the number of activities that is required to cover a certain part of the total. Half of the cases in the example event log end with activity E, the other half ends with activity F. So for both activities, the metric will hold the values 6 (absolute frequency) and 50 % (relative frequency), as shown in Table 5.23.

Table 5.23: End activities (activity level) applied to the running example.

activity	absolute frequency	relative frequency	cumulative sum
E	6	0.500	0.500
F	6	0.500	1.000

- *End activities - resource level.* On the level of the distinct resources, an overview of which resources execute the last activity per case can be of interest for a company. Probably this person is also responsible for the correct communication to the customer. Table 5.24 shows that five distinct resources are responsible for executing the end activities of the 12 cases in the running example. Only resource R1 never closes a case.

Table 5.24: End activities (resource level) applied to the running example. The resources are sorted on their relative frequency as an end resource.

Resource	absolute frequency	relative frequency	cumulative sum
R4	5	0.417	0.417
R3	4	0.333	0.750
R2	1	0.083	0.833
R5	1	0.083	0.917
R6	1	0.083	1.000

- *End activities - resource-activity level.* Finally, on the resource-activity level, this metric shows for each occurring resource-activity combination the absolute and relative number of times this resource executes this activity as an end activity in a case. Table 5.25 shows these absolute and relative numbers for the running example event log. The output is sorted on the relative frequency and the resource-activity combinations that do not occur as the last activity in the log are omitted from the table.

Table 5.25: End activities (resource-activity level) applied to the running example. The resource-activity combinations are sorted on the relative frequency that they include the end activity in a case.

Resource	activity	absolute frequency	relative frequency	cumulative sum
R4	F	5	0.417	0.417
R3	E	4	0.333	0.750
R2	E	1	0.083	0.833
R5	E	1	0.083	0.917
R6	F	1	0.083	1.000

5.5.2.2 Rework metrics

The key goal of lean management is waste reduction and avoiding non-value-adding activities [182]. As was stated in Chapter 2, activities that need to be performed more than once within a case can be defined as waste. The identification of waste within a business process has therefore been mentioned as one of the prime event log knowledge concepts in requirement 1 in Chapter 4. A first metric to measure the amount of rework within a process can be the frequency of the different activities in the entire event log or in a specific case or trace. Next to this, activities appearing more than once in a case, such as repetitions or self-loops, can also be indications of waste. It should be noted that rework can be inevitable within a process, as was illustrated by Dumas et al. [47].

- *Activity frequency - case level.* First of all, on the level of the specific cases, this metric shows the absolute and relative number of times the different activity types occur in each case. The absolute number shows the number of distinct activity types that occur in each of the cases. The relative number is calculated based on the total activity executions in the case. For the running example, this is shown in Table 5.26. For example, in case C2 five distinct activities occur, while six activity executions are recorded for this case. Therefore, the relative activity frequency for case C2 is 0.833. However, this metric does not show which activities are occurring more than the others in the case, resulting in a biased view. Therefore, it can be helpful to have a look at the results of this metric on the level of traces and on the level of the distinct activities. However, first of all, an aggregation on the level of the complete event log is provided.

- *Activity frequency - log level.* Next to the complete list of cases with their activity frequency, which can become very long, it can be interesting to have a look at the distribution of the distinct activities in the entire event log. This metric shows the summary statistics of the frequency of activities within a case, aggregated over the entire event log. Table 5.27 shows that a distinct activity appears on average 5.17 times per case with a standard deviation of 1.11.

- *Activity frequency - trace level.* This metric presents the absolute and relative number of times a specific activity type occurs in each trace. In trace A,B,B,C,D,D,E,F,F for example, six distinct activity types appear, while in total nine activities are executed. The relative activity frequency is therefore 0.667. The results of this metric applied to the running example are provided in Table 5.28.

Table 5.26: Activity frequency (case level) applied to the running example. The cases are sorted on their relative frequency.

case	absolute activity frequency	relative activity frequency
C6	6	1.000
C1	5	1.000
C7	3	1.000
C3	6	0.857
C9	6	0.857
C11	6	0.857
C10	5	0.833
C12	5	0.833
C2	5	0.833
C4	6	0.667
C5	6	0.667
C8	3	0.600

Table 5.27: Activity frequency (log level) applied to the running example.

min	q1	median	mean	q3	max	st.dev.	iqr
3	5	5.5	5.17	6	6	1.11	1

- *Activity frequency - activity level.* On the level of the activities, this metric provides the absolute and relative frequency of a specific activity in the entire event log. In our running example event log, in total 15 occurrences of activity B are found, which accounts for 19.7 % of the complete log (=15/76). This is shown in Table 5.29.

Self-loops. Activity instances of the same activity type that are executed more than once immediately after each other by the same resource are in a self-loop (length-1-loop). This was also stated to be an indication of not adding value to the process in Chapter 4. If an activity instance of the same activity type is executed 3 times after each other by the same resource, this is defined as a size 2 self-loop. An activity

Table 5.28: Activity frequency (trace level) applied to the running example. The traces are sorted on their relative activity frequency.

trace	relative trace frequency	absolute activity frequency	relative activity frequency
A,A,B,C,D,E	0.250	5	0.8333
A,B,C,D,E,F,F	0.167	6	0.857
A,B,B,B,C,D,E,F,F	0.083	6	0.667
A,B,B,C,D,D,E,F,F	0.083	6	0.667
A,B,C,D,E	0.083	5	1.000
A,B,C,D,E,F	0.083	6	1.000
A,B,D,C,E,F,F	0.083	6	0.857
B,D,E	0.083	3	1.000
B,E,D,E,E	0.083	3	0.600

Table 5.29: Activity frequency (activity level) applied to the running example. The activities are sorted on their relative frequency.

activity	absolute activity frequency	relative activity frequency
B	15	0.197
E	14	0.184
A	13	0.171
D	13	0.171
F	11	0.145
C	10	0.131

instance not followed by an activity instance of the same activity type, is a size 0 self-loop (no loop). For now, other patterns, such as length-n-loops or frequent episodes [89], are excluded from this research.

The metrics presenting self-loops, together with the metrics concerning the repetitions which will be introduced later, should take into account if the activity is redone

	Immediately following		
Same Resource	Repeat self-loop	Redo self-loop	Different resource
	Repeat repetition	Redo repetition	
	Not immediately following		

Figure 5.5: Dimensions of the rework metrics.

by the same resource or by another one. Therefore, these metrics are created according to four concepts, which are presented in Figure 5.5. Two types of self-loops are presented here, which are repeat self-loops and redo self-loops. Repeat self-loops are activity executions of the same activity type that are executed immediately following each other by the same resource. Redo self-loops are activity executions of the same activity type that are executed immediately following each other by a different resource. Repeat and redo repetitions are explained further on.

These metrics are presented on five different levels of analysis, which are the entire event log, cases, activities, resources and resource-activity combinations. On the trace level, no summary statistics are provided as each different occurrence of a trace is exactly the same concerning the sequence of activities within the trace.

- *Number of self-loops - case level.* A first interesting metric is the number of self-loops. This metric on the level of cases provides an overview of the absolute and relative number of repeat and redo self-loops in each case. This can be interesting for companies to find which cases are the ones with the self-loops. To calculate the relative number, each (repeat or redo) self-loop is counted as 1 occurrence (a self-loop dummy), and the other activity instances are also counted as 1. Case C4 and case C5 contain the highest number of repeat self-loops in the example, which is two or 22.22 % of all nine occurrences in the case. This metric for both the repeat and redo self-loops on the case level is shown in Table 5.30.

- *Number of self-loops - log level.* Aggregated on the level of the entire event log, the summary statistics of the number of self-loops within a case can give an insight in the amount of waste in an event log. In the running example, case C5 contains two repeat self-loops: one for activity B and one for activity F, both executed by resource R4. On average, each case in the example contains 0.833 repeat self-loops. This is shown in Table 5.31. The standard deviation is 0.718. Not all cases contain a repeat self-loop, so the minimum number is zero and the

Table 5.30: Number of repeat and redo self-loops (case level) applied to the running example.

case	absolute number of repeat self-loops	relative number of repeat self-loops	absolute number of redo self-loops	relative number of redo self-loops
C1	0	0	0	0
C2	1	0.167	0	0
C3	1	0.143	0	0
C4	2	0.222	1	0.111
C5	2	0.222	0	0
C6	0	0	0	0
C7	0	0	0	0
C8	1	0.200	0	0
C9	1	0.143	0	0
C10	1	0.167	0	0
C11	1	0.143	0	0
C12	0	0	1	0.167

maximum number of repeat self-loops within one case is two. In total 10 repeat self-loops occur in the event log. The number of redo self-loops on the level of the entire event log is shown in Table 5.32. Only two redo self-loops occur in the running example, one in case C4 and one in case C12, resulting in rather meaningless results on the log level.

Table 5.31: Number of repeat self-loops (log level) applied to the running example.

min	q1	median	mean	q3	max	st.dev.	iqr
0	0	1	0.833	1	2	0.718	1

- *Number of self-loops - activity level.* Furthermore, on the level of the distinct activities in the event log, the absolute and relative number of self-loops per activity can be an indication for the company which activities are causing the most waste in the process. Table 5.33 shows the absolute and relative number

Table 5.32: Number of redo self-loops (log level) applied to the running example.

min	q1	median	mean	q3	max	st.dev.	iqr
0	0	0	0.167	0	1	0.389	0

of both repeat and redo self-loops within the running example. It is remarkable that activity F is in a repeat self-loop five times, which was not clear on the log or case level.

Table 5.33: Number of repeat and redo self-loops (activity level) applied to the running example.

activity	absolute number of repeat self-loops	relative number of repeat self-loops	absolute number of redo self-loops	relative number of redo self-loops
A	2	0.154	1	0.077
B	1	0.067	1	0.067
C	0	0	0	0
D	1	0.067	0	0
E	1	0.071	0	0
F	5	0.455	0	0

- *Number of self-loops - resource level.* Similar to the metric on the level of the activities, the number of self-loops on the level of the resources executing the activities can give a company insights in which employee needs to repeat his or her work most often within a case, or for which employee the work he or she did should be redone by another employee within the same case. This metric shows the absolute and relative number of both repeat and redo self-loops for each resource in the event log. Applied to the running example, this results in the finding that are shown in Table 5.34. It can be stated that resource R4 should be paid attention to, because he or she repeats five times an activity immediately after each other, while all other resources only appear in zero, one, or two repeat self-loops.
- *Number of self-loops - resource-activity level.* Finally, the metric can be applied to the level of the specific resource-activity combinations, in order to get an

Table 5.34: Number of repeat and redo self-loops (resource level) applied to the running example.

resource	absolute number of repeat self-loops	relative number of repeat self-loops	absolute number of redo self-loops	relative number of redo self-loops
R1	1	0.083	1	0.083
R2	0	0	0	0
R3	2	0.167	1	0.083
R4	5	0.250	0	0
R5	0	0	0	0
R6	2	0.182	0	0

insight in which activities are the most crucial for which resources. This metric shows the absolute and relative number of both repeat and redo self-loops for each of the resource-activity combinations that occur in the event log. Two different relative numbers are provided here, one from the resource perspective and one from the activity perspective. At the resource perspective, the denominator is the total number of executions by the resource under consideration. At the activity perspective, the denominator is the total number of occurrences of the activity under consideration. For the running example, especially the resource-activity combination R4-F is remarkable, as it appears four times in a repeat self-loop, while all other resource-activity combinations that occur in a repeat or redo self-loop only do this once. The four repeat self-loops that are recorded for resource R4 executing activity F have a relative number of 36.36 % of in total 11 executions of activity F in the entire event log. Relative to the total number of executions by resource R4 in the entire event log, which is 20, the four repeat self-loops of resource-activity combination R4-F count for 20 % of the occurrences.

- *Size of self-loops - case level.* The size of a self-loop is based on the number of activity executions of the same activity within a self-loop. A distinction can again be made between repeat self-loops, which are executed by the same resource, and redo self-loops, where the second, third and following self-loop instances are executed by another resource than the first one.

On the level of the different cases within the event log, this metric provides the summary statistics of the size of repeat and redo self-loops, for each case (in which a repeat or redo self-loop occurs, respectively). For the running example event log, only one of the cases in the event log contains a repeat self-loop of size 2 (which means that an activity is executed 3 times immediately after each other within a case). This is the self-loop of activity B executed by resource R4 in case C5. All other repeat self-loops in the running example are of size of 1. Showing the summary statistics for this metric applied to the running example would therefore not add any value in this research. The two redo self-loops, which occur in case C4 and case C12 both have size 1, making the summary statistics per case unnecessary to be shown here.

- *Size of self-loops - log level.* On the level of the entire event log, this metric provides summary statistics about the size of the self-loops that occur in the entire event log, separated between repeat and redo self-loops. Next to this, an overview is given of all self-loops showing in which case they occur, who executes them and which activity is involved. However, this can become a very long list in a real-life event log, which makes it less useful. For the redo self-loops, an additional piece of information that is provided by this metric is which resource executes the first occurrence in the self-loop, and which resource executes the last occurrence in the self-loop. This can give a company an insight in which resources are responsible for the first and last occurrence, giving an indication of who started the activity and who solved it when the first resource was not able to finish it correctly. However, this is very case-specific. As was shown on the level of the specific cases, for the 10 repeat self-loops in total, only one has a size 2. All other repeat self-loops in the running example are of size 1. This is shown in Table 5.35. For the two redo self-loops in the running example event log, the size is in both cases 1.

- *Size of self-loops - activity level.* The size of self-loops on the level of distinct activities in the event log can provide insights in which activities are most prone to be executed more than once before finishing them correctly, indicating that they cause more waste within the process. This metric shows the summary statistics of the size of the self-loops per activity together with the number of self-loops each activity occurs in and the relative frequency of the activity in order to get a notion of the extent of the problem. The metric is, similar to the other self-loop metrics, calculated for both repeat and redo self-loops. Applied to the running example event log, we find, for example, that activity B appears

Table 5.35: Size of repeat self-loops (log level) applied to the running example.

min	q1	median	mean	q3	max	st.dev.	iqr
1	1	1	1.1	1	2	0.316	0

repeat self-loop	case	activity	resource	size
1	C2	A	R3	1
2	C3	F	R6	1
3	C4	D	R6	1
4	C4	F	R4	1
5	C5	B	R4	2
6	C5	F	R4	1
7	C8	E	R3	1
8	C9	F	R4	1
9	C10	A	R1	1
10	C11	F	R4	1

once in a repeat self-loop, which has a size of two. The results of this metric for repeat self-loops on the level of specific activities applied to the running example event log are shown in Table 5.36. For the size of redo self-loops in the running example, only two activities occur in a redo self-loop, which are A and B, who both occur in a size 1 redo self-loop. These results are therefore not shown in a table.

Table 5.36: Size of repeat self-loops (activity level) applied to the running example.

activity	relative activity frequency	number of repeat self-loops	min	q1	mean	median	q3	max	st. dev.	iqr
A	0.171	2	1	1	1	1	1	1	0	0
B	0.197	1	2	2	2	2	2	2	NA	0
D	0.171	1	1	1	1	1	1	1	NA	0
E	0.184	1	1	1	1	1	1	1	NA	0
F	0.145	5	1	1	1	1	1	1	0	0

- *Size of self-loops - resource level.* On the level of the resources executing the activities in the event log, this metric shows the summary statistics of the size of both repeat and redo self-loops for each of the resources in the event log. Next to these summary statistics, also the relative resource frequency and the number of repeat or redo self-loops for each of the resources occurring in a repeat or redo self-loop is added to get an insight in the importance of the resource in the process. For the size of the repeat self-loops, this metric can show which resources need more than one try to execute an activity before it is finished correctly for a certain case. For the size of the redo self-loops, this metric provides insights in which resources execute an activity that is redone by another resource immediately following the execution by the first resource. Table 5.37 shows the results of the size of repeat self-loops metric in the running example event log. Similar findings can be identified for the size of redo self-loops on the resource level, where the results show that resources R1 and R3 are the only resources that execute an activity in the event log which is redone immediately again by another resource.

Table 5.37: Size of repeat self-loops (resource level) applied to the running example.

resource	relative resource frequency	number of repeat self-loops	min	q1	mean	median	q3	max	st. dev.	iqr
R1	0.158	1	1	1	1.0	1	1	1	NA	0
R3	0.158	2	1	1	1.0	1	1	1	0.000	0
R4	0.263	5	1	1	1.2	1	1	2	0.447	0
R6	0.145	2	1	1	1.0	1	1	1	0.000	0

- *Size of self-loops - resource-activity level.* Finally, on the level of the specific resource-activity combinations that occur in the event log, the size of self-loops metric shows the summary statistics of the size of both repeat and redo self-loops for each of the resource-activity combinations that occur in a repeat or redo self-loop in the event log. For the size of the repeat self-loops, this metric can show which resources need more than one try to execute a specific activity before it is finished correctly for a certain case. For the size of the redo self-loops, this metric provides insights in which resources execute a specific activity which is redone by another resource immediately following the activity execution. Applied to the running example, this metric shows for example that resource R4 occurs four times in a self-loop that involves activity F, which are all of size 1.

Repetitions. Instead of only looking at activity instances that are executed immediately following each other, the notion of repetitions (hereby excluding self-loops) can also be interesting in the context of process behaviour. A repetition is an execution of an activity within a case while that activity has already been executed before, but one or more other activities are executed in between. A repetition might also be an indication of waste, however it should be possible to report on them separately from self-loops. Similar to the self-loop metrics explained above, again a distinction should be made between repeat and redo repetitions, as was shown in Figure 5.5. Repeat repetitions are activity executions of the same activity type that are executed not immediately following each other, but by the same resource. Redo repetitions are activity executions of the same activity type that are executed not immediately following each other and by a different resource than the first activity occurrence of this activity type. In trace B,E,D,E,E in the running example event log, 1 repetition is reported, next to 1 self-loop (both on activity E). Similar to the metrics concerning self-loops, two types of metrics will be introduced here, the number of repetitions and the size of repetitions.

- *Number of repetitions - case level.* First of all, on the level of the specific cases, this metric provides the absolute and relative number of repetitions in each case, for both repeat and redo repetitions. In the running example, one repeat activity is found in case C8, where activity E is performed by resource R3 followed by activity D performed by resource R2. Next, resource R3 repeats activity E. As this is the only repeat repetition in the entire event log, and no redo repetitions are recorded, this metric does not add any value in the case of the running example.

- *Number of repetitions - log level.* Next to this, the number of repetitions within a case can be aggregated on the level of the entire event log, which can provide insights in the amount of waste in an event log. Each combination of two occurrences of the same activity, executed not immediately following each other, by the same resource is counted as one repeat repetition of this activity. In the running example, only case C8 contains one repeat repetition.

- *Number of repetitions - activity level.* On the level of specific activities, this metric shows which activities occur the most in a repetition, implying that the company should analyse them in order to prevent these activities from causing waste. The absolute and relative number of both repeat and redo repetitions is provided by this metric, giving an overview per activity. In the running example event log, activity E appears once in a repeat repetition, which counts for 7.14 % of the total number of occurrences of activity E. None of the activities appear in a redo repetition.

- *Number of repetitions - resource level.* When looking at the different resources executing activities in the event log, it can be interesting to have an overview of which resources need more than one chance to execute an activity in a case or which resources need to have an activity redone later on in the case by another resource. This metric provides the absolute and relative number of times each resource appears in a repetition. In the running example, resource R3 is the only resource that executes an activity more than once within the same case, and not immediately following each other. This counts for 8.33 % of the total executions resource R3 performs within the entire event log. No other resource executes activities that need to be repeated or redone later on within the same case.

- *Number of repetitions - resource-activity level.* Finally, the same metric can be looked at on the level of specific resource-activity combinations, providing the company with specific information about which activities and which resources are involved in the repetitions. For this metric the absolute and relative number of repeat and redo repetitions is provided. Again two different relative numbers are provided, one relative to the total number of executions of the activity in the entire event log, and one relative to the total number of executions performed by the resource throughout the entire event log. For the repeat repetitions, this metric provides one result for the running example event log, which is the resource-activity combination R3-E which occurs once in a repeat repetition in the entire event log, which accounts for 7.14 % of the total amount of executions of activity E in the entire event log, and for 8.33 % of the total activity executions by resource R3. No output comes from the metric for the number of redo repetitions when it is applied to the running example event log.

- *Size of repetitions - case level.* Next to the number of repetitions that occur in an event log, it can also be interesting to have a look at the amount of activity executions that occur within the repetitions. On the level of the specific cases, this metric provides the summary statistics of the size of both repeat and redo repetitions for each case in the event log. In the running example event log, only one repetition occurs, which has size one. This repeat repetition occurs in case C8, which has the following trace: B,E,D,E,E. Activity E is repeated after activity D is executed in between. The last occurrence of activity E is not part of the repetition as it is a self-loop of the same activity, which are strictly kept separated in the metric calculations to prevent that double calculations occur.

- *Size of repetitions - log level.* On the level of the entire event log, this metric provides the summary statistics of the size of repetitions throughout the entire event log, both for repeat repetitions and redo repetitions. Next to this, also an overview of all repetitions occurring in the entire event log is provided, showing the case in which they occur, the activity that is repeated, the total length of the trace, the resource executing the different occurrences of the activity in the repetition and the total amount of activity occurrences within the repetition. As the running example only contains one repetition, which is a repeat repetition of size one, this metric does not provide any new insights for this event log.

- *Size of repetitions - activity level.* To get an insight into which activities are repeated within a process, this metric can be calculated on the activity level. This way, a company gets an overview of how many times the activity within a repetition is repeated, which is possibly a notion of waste and not adding value to the process. For both repeat and redo repetitions, this metric provides for each activity that occurs in a repeat or redo repetition the summary statistics of the amount of times this activity is repeated within the same case. For the running example, this metric only identifies activity E to be in a repeat repetition of size 1, providing no new insights.

- *Size of repetitions - resource level.* Next to the activity level, the size of repeat and redo repetitions can also be calculated on the resource level, providing for each resource the summary statistics of the amount of times activities are repeated within a case. Companies can gather information on which resources are possibly less efficient, what should be investigated more thoroughly. For the running example, this metric provides the summary statistics for resource R3 as he or she executes an activity more than once within the same case with at least one other activity in between. As this is the only repetition occurring, no interesting findings can be drawn from this.

- *Size of repetitions - resource-activity level.* Finally, to get a better notion of which resources and which activities are responsible for the repetitions within an event log, the size of repetitions can be calculated for each resource-activity combination. Therefore, this metric provides an overview of the summary statistics of the size of the repetitions that occur, both for repeat and redo repetitions. However, also on this level no interesting conclusions can be drawn from the application of this metric to the running example as only one resource-activity combination occurs, which occurs in a size 1 repeat repetition.

5.5.2.3 Resource metrics

The importance of resources is already recognised in the metrics above, where resources are seen as one of the levels of analysis for most of the metrics, stating some interesting findings on the resource aspect in the field of operational excellence. A classification of resources in the domain of project management can be found in Jugdev and Mathur [80]. However, this research focuses on resources that are defined as process participants, software systems, or equipment in the field of BPM [47]. Complementary to the findings above, resources can also be a source of process variability and their behaviour is essential in the light of continuous process improvement as was defined by van Assen [154]. Consequently, this dimension has been defined as one of the crucial required measures in the field of operational excellence, as can be found in Chapter 4. It should therefore be taken into account to convey a more comprehensive picture on process behaviour to organisations. This is consistent with the research recommendation of Recker and Mendling [127] as it targets the resource perspective in process mining.

Given the need to include the resource perspective, this section presents some more resource-related process insights. Within the context of quantifying the resource perspective using event logs, metrics that mainly focus on the relationship between resources are proposed by Song and van der Aalst [144]. While the latter specify metrics with the purpose of mining organisational models, Huang et al. [70] and Pika et al. [119] focus on defining resource behaviour measures. The metrics presented here complement this as well as the recently introduced resource availability metrics [96] and the work prioritisation patterns [147], which are also based on event log data. Besides the general contribution of providing algorithm-agnostic resource insights to organisations, these metrics can also support organisations in performing knowledge management, for instance when creating a knowledge map [29], or project management with applications such as resource levelling or resource allocation [81].

Resources are assigned to activities and typically carry these out on multiple cases such as files or products. Getting insights in the behaviour of these resources and the amount of "waste" they cause can be very interesting for companies who want to optimise their business processes. However, it should be stressed that resources can possibly be involved in multiple processes within an organisation, while the metrics presented here only concern a single process. Related to this concept of resources, are the metrics concerning the concept of batch processing, which are presented in Chapter 6. The metrics presented here, concerning resource frequency, resource involvement, and resource specialisation, are also explained in Swennen et al. [149] where they are applied to an artificial event log containing medical activities executed by the staff members of a hospital.

- *Resource frequency - case level.* Comparable to the concept of the activity frequency presented earlier, the frequency of resources in a business process can also be very insightful for companies, e.g., during company restructuring. Dumas et al. [47] already defined a similar measure as the "level of busyness of resources", and states that waiting time depends on this level of busyness, i.e., the more active resources are, the higher the waiting times can become. To get insights in the resource variance between the different cases, the summary statistics of the frequency of resources can be calculated on the level of the cases. This way, a company gets an insight in the resource variation by analysing the number of different resources working on each case together with the number of activities that a resource executes per case. In Table 5.38 we see, for example, that the six activities in case C2 are executed by three different resources, which gives an average of two activities per resource. At the trace level, this metric is less informative because, even though the sequence of activities is the same, the persons or machines executing the activities in the trace can be completely different per trace occurrence. For example, trace A,A,B,C,D,E appears three times in the event log, in case C2, C10, and C12. However, only activity D is in the three cases executed by the same resource, while all other activities are executed by two or even three different resources. Providing summary statistics for each trace could thus be misleading.

- *Resource frequency - log level.* On the level of the entire event log, summary statistics show the number of times a resource executes an activity in the entire event log. For the running example event log, we see in Table 5.39 that a resource executes on average 12.67 activities in the entire event log, with a standard deviation of 3.983. It is clear that there is a lot of diversification between the resources, because there is one resource only executing eight activities, while another resource executes 20 activities in total.

Table 5.38: Resource frequency (case level) applied to the running example. The cases are sorted on the number of different resources in the case.

case	number of resources	min	q1	median	mean	q3	max
C9	6	1	1.00	1.0	1.667	1.00	2
C4	5	1	1.00	1.0	1.800	2.00	4
C12	5	1	1.00	1.0	1.200	1.00	2
C1	4	1	1.00	1.0	1.250	1.25	2
C3	4	1	1.00	1.5	1.750	2.25	3
C5	4	1	1.00	1.0	2.250	2.25	6
C6	4	1	1.00	1.0	1.500	1.50	3
C10	4	1	1.00	1.0	1.500	1.50	3
C11	4	1	1.00	1.5	1.750	2.25	3
C2	3	1	1.50	2.0	2.000	2.50	3
C8	3	1	1.00	1.0	1.667	2.00	3
C7	2	1	1.25	1.5	1.500	1.75	2

Table 5.39: Resource frequency (log level) applied to the running example.

min	q1	median	mean	q3	max	st.dev.	iqr
8	11.250	12	12.67	12.75	20	3.983	1.5

- *Resource frequency - activity level.* On the level of the different activities, the resource frequency states how many different resources are executing a specific activity in the entire event log. For the running example event log, we find in Table 5.40 that activity A is executed by three different resources throughout the entire event log. Because the activity is executed 13 times in total, this is an average of 4.333 executions of activity A per resource.

- *Resource frequency - resource level.* On the level of the distinct resources in the event log, this metric shows the absolute and relative frequency of occurrences of each resource in the entire event log. Resource R4 executes in total 20 activities in the running example event log, which accounts for 26.3 % of the total number of activities executed in the running example event log. This is shown in Table 5.41.

Table 5.40: Resource frequency (activity level) applied to the running example.

activity	number of resources	min	q1	median	mean	q3	max
A	3	4	4.00	4.0	4.333	4.50	5
B	2	7	7.25	7.5	7.500	7.75	8
C	3	1	2.50	4.0	3.333	4.50	5
D	3	2	2.50	3.0	4.333	5.50	8
E	3	1	3.00	5.0	4.667	6.50	8
F	2	2	3.75	5.5	5.500	7.25	9

Table 5.41: Resource frequency (resource level) applied to the running example. The resources are sorted on their absolute frequency.

resource	absolute frequency	relative frequency
R4	20	0.263
R2	13	0.171
R1	12	0.158
R3	12	0.158
R6	11	0.145
R5	8	0.105

- *Resource frequency - resource-activity level.* Finally, at the most specific level of analysis, the absolute and relative number of times each resource-activity level occurs in the entire event log can be calculated. Two different relative numbers are provided here, one from the resource perspective and one from the activity perspective. At the resource perspective, the denominator is the total number of executions by the resource under consideration. At the activity perspective, the denominator is the total number of occurrences of the activity under consideration. Table 5.42 shows for example that resource R4 executes activity F nine times in total, which counts for a relative frequency of 45 %

of all 20 executions done by resource R4 and 81.8 % of all 11 occurrences of activity F, throughout the entire event log. This information can be useful for organisations to make changes within the resource allocation process.

Table 5.42: Resource frequency (resource-activity level) applied to the running example. The resource-activity combinations are sorted on their absolute frequency.

resource	activity	absolute frequency	relative frequency (resource)	relative frequency (activity)
R4	F	9	0.450	0.818
R6	D	8	0.727	0.615
R1	B	8	0.667	0.533
R3	E	8	0.667	0.571
R4	B	7	0.350	0.467
R5	C	5	0.625	0.500
R2	A	5	0.385	0.384
R2	E	5	0.385	0.357
R1	A	4	0.333	0.308
R3	A	4	0.333	0.308
R4	C	4	0.200	0.400
R2	D	3	0.231	0.231
R5	D	2	0.250	0.154
R6	F	2	0.182	0.182
R5	E	1	0.125	0.071
R6	C	1	0.091	0.100

- *Resource involvement - case level.* Next to the resource frequency, the involvement of resources in cases can be of interest to measure, e.g., how involved or how "indispensable" they are. This metric is provided on three levels of analysis, which are the cases, the resources, and the resource-activity combinations. On the level of the specific cases, the absolute and relative number of distinct resources executing activities in each case is calculated. This way a company gets an overview of which cases are handled by a small amount of resources and which cases require more resources, indicating a higher level of variance in the process. In Table 5.43, we see that in case C9 all six distinct resources, or 100 % of the resources, are involved, while case C7 could be executed by only two distinct resources, which is only 33.3 % of the total number of resources involved in the process. It can be interesting for the company to look into this by checking if these two resources are involved in all of the cases or if the resources differ over the distinct cases.

Table 5.43: Resource involvement (case level) applied to the running example. The cases are sorted on the absolute number of distinct resources in the case.

case	absolute number of distinct resources	relative number of distinct resources
C9	6	1.000
C12	5	0.833
C4	5	0.833
C1	4	0.667
C10	4	0.667
C11	4	0.667
C3	4	0.667
C5	4	0.667
C6	4	0.667
C2	3	0.500
C8	3	0.500
C7	2	0.333

- *Resource involvement - resource level.* On the level of the distinct resources, this metric provides the absolute and relative number of cases in which each resource is involved, indicating which resources are more "necessary" within the business process than the others. Table 5.44 shows that resource R1 is involved in ten of the cases of the running example event log, which is 83.33 % of all 12 cases. There are no resources that are only involved in one or two cases, which could be an indication for the company management that either these cases should be looked at, as they require resources that are usually not involved, or that these resources should be checked because they could maybe spend their time better elsewhere.

Table 5.44: Resource involvement (resource level) applied to the running example. The resources are sorted on the number of cases they are involved in.

resource	absolute number of cases	relative number of cases
R1	10	0.833
R3	8	0.667
R4	8	0.667
R6	8	0.667
R2	7	0.583
R5	7	0.583

- *Resource involvement - resource-activity level.* On the level of the specific resource-activity combinations, this metric provides a list of all resource-activity combinations with the absolute and relative number of cases in which each resource-activity combination is involved. Table 5.45 shows that resource R1 executes activities A and B in respectively three and eight cases in the running example event log. On the level of the distinct activities, this metric is not developed as these values are accommodated in the resource specialisation metric.

- *Resource specialisation - case level.* Finally, we can also have a look at the specialisation level of the different resources in a company. This can give a company an overview of which resources are performing certain activities more than others, and which resources are responsible for containing all knowledge or capabilities about one topic and can therefore be seen as bottlenecks. This information can be used to tackle challenges such as team selection or brain drain, as presented by Creemers and Jans [29]. Based on these results, a company can take decisions to make changes in team compositions throughout the process. On the level of the cases, this metric provides the summary statistics of the distinct activities executed per resource in each case. For comparison reasons, the number of distinct activities that are executed within each case is added, to gain

Table 5.45: Resource involvement (resource-activity level) applied to the running example. The resource-activity combinations are sorted on the number of cases they are involved in.

resource	activity	absolute number of cases	relative number of cases
R1	B	8	0.667
R6	D	7	0.583
R3	E	6	0.500
R2	A	5	0.417
R2	E	5	0.417
R4	B	5	0.417
R4	F	5	0.417
R5	C	5	0.417
R4	C	4	0.333
R1	A	3	0.250
R2	D	3	0.250
R3	A	3	0.250
R5	D	2	0.167
R5	E	1	0.083
R6	C	1	0.083
R6	F	1	0.083

insights in the importance of the results of this metric. Table 5.46 shows for the running example event log that in case C1 five distinct activities are executed. For the same case, we can see that a resource working on this case executes at least one distinct activity and maximum two, i.e., resource R4 is the only one who executes two distinct activities in this case, while all other resources only execute one distinct activity. In the other cases, the maximum number of distinct activities executed by one resource is at most three, while the number of distinct activities is almost always five or six within a case, indicating that the resources within this process are able to perform multiple activities, but no resource is able to perform all activities. This notion of specialisation per activity and per resource will become more clear on the other levels of analysis.

Table 5.46: Resource specialisation (case level) applied to the running example.

case	number of distinct activities	min	q1	median	mean	q3	max	st. dev.	iqr
C1	5	1	1.00	1.00	1.250	1.25	2	0.500	0.250
C2	5	1	1.50	2.00	1.667	2.00	2	0.577	0.500
C3	6	1	1.00	1.50	1.500	2.00	2	0.577	1.000
C4	6	1	1.00	1.00	1.400	1.00	3	0.894	0.000
C5	6	1	1.00	1.00	1.500	1.50	3	1.000	0.500
C6	6	1	1.00	1.00	1.500	1.50	3	1.000	0.500
C7	3	1	1.25	1.50	1.500	1.75	2	0.708	0.500
C8	3	1	1.00	1.00	1.000	1.00	1	0.000	0.000
C9	6	1	1.00	1.00	1.000	1.00	1	0.000	0.000
C10	5	1	1.00	1.00	1.250	1.25	2	0.500	0.250
C11	6	1	1.00	1.00	1.500	1.50	3	1.000	0.500
C12	5	1	1.00	1.00	1.200	1.00	2	0.447	0.000

- *Resource specialisation - log level.* On the level of the entire event log, this metric provides summary statistics of the distinct activities executed per resource. For the running example event log, we find in Table 5.47 that resources execute on average 2.667 distinct activities within the process.

Table 5.47: Resource specialisation (log level) applied to the running example.

distinct activities	min	q1	median	mean	q3	max
6	2	2.250	3	2.667	3	3

- *Resource specialisation - activity level.* On the level of the distinct activities, this metric provides an overview of the absolute and relative number of different resources executing this activity within the entire event log. This will give a company insights in which activities resources are specialised in. Activity A, for example, is being executed by three different resources throughout the running example event log. Table 5.48 shows the calculations for each of the distinct activities.

Table 5.48: Resource specialisation (activity level) applied to the running example. The activities are sorted on the number of different resources executing them throughout the event log.

activity	absolute frequency	relative frequency
A	3	0.500
C	3	0.500
D	3	0.500
E	3	0.500
B	2	0.333
F	2	0.333

- *Resource specialisation - resource level.* Finally, the resource specialisation can also be calculated on the resource level, showing the absolute and relative number of distinct activities that each resource executes. In Table 5.49 we find that resource R1 and resource R3 only work on two distinct activities throughout the entire event log, which counts for 33.33 % of the total number of distinct activities in the event log. All other resources are responsible for three different activities.

5.6 R-package edeaR

All metrics presented above have been implemented as functions in the R-package edeaR [74], which stands for exploratory and descriptive event-based data analysis in R [77]. R is an open-source programming language which is used extensively for the purpose of statistical analysis and data mining, and furthermore contains extensive functionalities for data visualisation. EdeaR enables the handling and analysis of event logs within R, and is fully compatible with the existing XES standard [64]. The

Table 5.49: Resource specialisation (resource level) applied to the running example. The resources are sorted on the number of distinct activity types they execute throughout the event log.

Resource	absolute frequency	relative frequency
R2	3	0.500
R4	3	0.500
R5	3	0.500
R6	3	0.500
R1	2	0.333
R3	2	0.333

package is available through cran[2], and comes with several vignettes, which provide an illustrative walk-through. EdeaR is part of the overarching open-source suite bupaR[3] [73], which is developed by Janssenswillen et al. [75]. BupaR provides support for different stages in process analysis, such as importing event data, calculating descriptives, process monitoring, and process visualisation. Also the preparations of an event log before the presented metrics can be applied can be easily done with the functionalities of bupaR.

5.7 Metric dashboard

As was stated in Chapter 4, the last requirement that was defined in the requirement analysis states that the results of the metrics should be easy to interpret for business people and supported with suitable visual representations in order to increase the understandability. Therefore, a dashboard was created with the R-package shiny[4] [23] in which the results of the metrics are visualised. The dashboard homepage provides an overview of the dataset that is analysed by outlining some general descriptions of it. Here, the number of cases, the number of activities, the number of traces, the number of activity executions, the number of distinct resources, and the number of events is provided. Moreover, it also shows the start and end date of the entire event log, showing the time range in which the data was collected.

[2]https://cran.rstudio.com/web/packages/edeaR/
[3]http://bupar.net/
[4]https://cran.r-project.org/package=shiny

Next, for each of the different categories of metrics presented above, a separate page is provided in which the results of the metrics are shown with a clear and easy-to-read visualisation in the form of a chart. An example of a metric visualisation is given in Figure 5.6. As can be seen here, the time unit that is used in most of the metrics can be changed easily in the lower left of the dashboard.

Another option within the metric dashboard are the filters, allowing people from business to easily "play" with the data in order to get deeper insights in the different aspects of the dataset. Here it is possible to choose between two types of filters, which are case filters and event filters. All metric visualisations are updated at run-time when filters are applied, adding much more value to the dashboard and the presented metrics. To get a notion of the different filters that are available in the dashboard, an overview is provided here. For the case filters, possibilities to filter the event log are (i) activity presence, in which certain activities can be ignored in the analyses, (ii) throughput time, which gives one the option to only include cases of a certain length, and (iii) time intervals, making it possible to specify certain time ranges. The event filters add to this the possibility to filter the event log on the level of events, by activity, by resource, or by time interval.

These filters, combined with the metrics presented above, provide business analysts with much more information than solely the output of each metric. When reference values or process variants are present, for example, benchmarking analyses can be performed in order to create an integrated view on the results and to get insights into the quality of the business process. Comparing the throughput time of the cases within the event log with a desired or modelled throughput time is much more interesting than only looking at the calculated values, for instance. Next to this, the metrics and their visualisations within the dashboard make it easier to analyse the quality and performance of the business process over time, which can be insightful to cover seasonal or time-bounded characteristics, or to just find out which periods to focus on. Finally, the dashboard provides insights in which metrics can possibly be calculated for a certain event log. This makes it easier for business people to focus on their pains and gains first, and to not get lost in an overload of numbers that are harder (and take more time) to interpret than visual representations.

As the running example dataset, which was presented in Section 5.4, is limited in size, the dashboard visualisations will be presented in Chapter 7, where all metrics that are introduced in this chapter are applied to a real-life dataset. The results are complemented with some visualisations from the dashboard in which the data is analysed. Related to the previous, the fourth requirement that was defined in Chapter 4 also states that the concepts that are used in the analyses and in reporting the findings

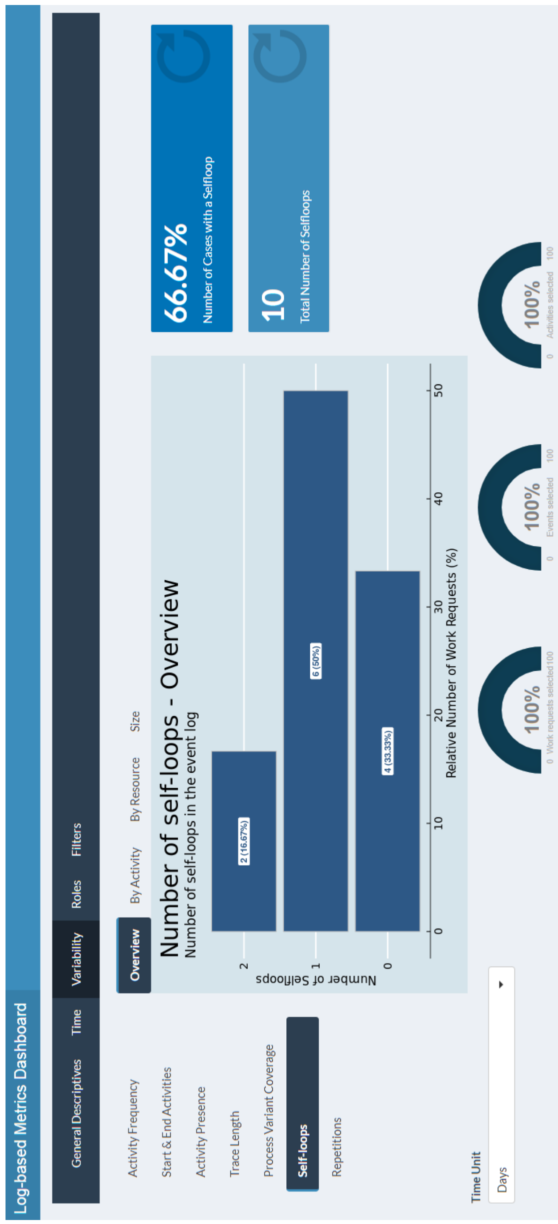

Figure 5.6: Example of the metrics dashboard including the visualisation of the number of self-loops metric.

to the business people should be understandable to them. A translation of the rather technical concepts to more business-wise items would improve the interpretation by the right people. Therefore, the dashboard for the case study in Chapter 7 has been slightly modified to include some more understandable concepts instead of the process mining language that was used during the metric presentation in the chapter at hand.

5.8 Discussion

The presented metrics are developed in accordance with the four artifact requirements that were defined in Chapter 4. Firstly, the metrics can be mapped onto the list of required metrics that was composed in **requirement 1** based on the findings from literature in Chapter 2 and Chapter 3 and from the interviews described in Chapter 4. The metrics that have been developed concern the throughput time, the processing or service time, the waiting time, the process flow, the frequency of activities, the frequency and number of patterns, rework, resource variation, and bottleneck activities and resources. Two other categories of event log knowledge that should be identified from event logs in order to support operational excellence are the number of activities executed in batch and the size of these batches. The concept of batch processing is elaborated upon in Chapter 6. Figure 5.7 shows how the developed metrics fit into the required metrics that have been defined in requirement 1.

Secondly, according to **requirement 2**, the metrics should only measure one dimension and should be measurable on a specific level of analysis. Therefore, different levels of analysis should be taken into account. As could be seen in Figure 5.4, the developed metrics are measurable on the following levels of analysis: log, case, trace, activity, resource, and resource-activity. Not all metrics are measurable on each of these levels, as not all combinations are logical or insightful.

Thirdly, the requirements of the underlying event log data are described within this chapter, together with the event log-based metrics that have been developed. A clear definition and an example of how the metrics should be calculated is provided, which fulfils **requirement 3**. Next to the transparency provided in this chapter, all metrics are also implemented in the R-package edeaR, for which the underlying programming code is openly available through cran, including several vignettes which provide an illustrative walk-trough for each of the developed metrics.

Finally, **requirement 4** states that the metrics should be supported with suitable visual representations and made understandable for business people. To fulfil this requirement, a metric dashboard was created and presented, which includes vi-

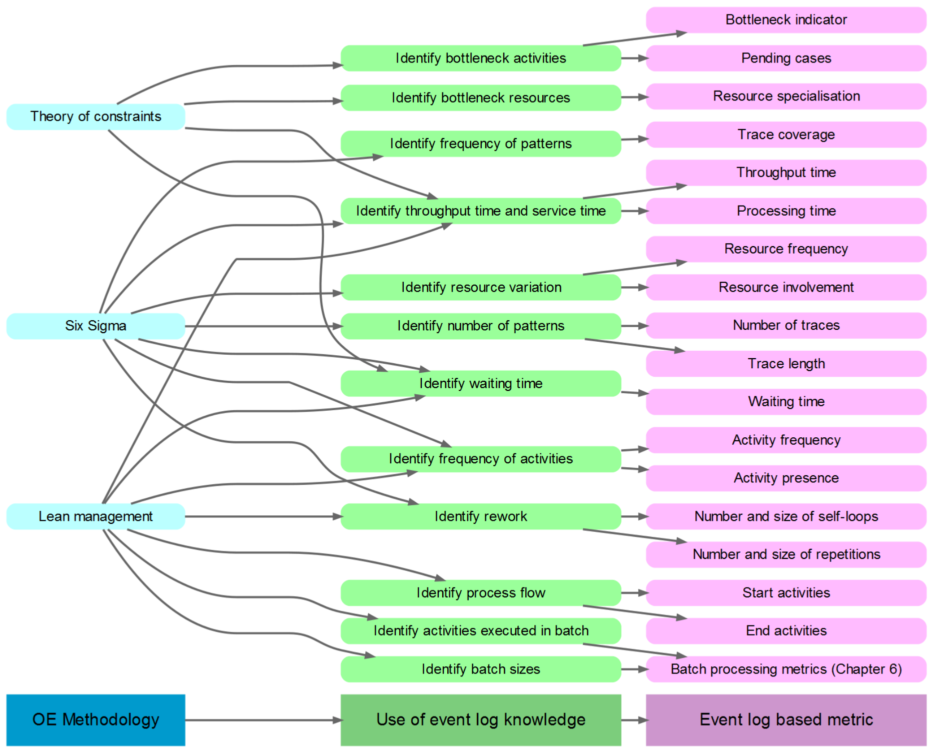

Figure 5.7: The developed log-based metrics mapped onto the required metrics in operational excellence.

sual representations and aggregations of the results of the presented metrics. The dashboard is developed for the running example presented in the chapter at hand, and is easily transformable for other processes. This has been done for the real-life case study that will be analysed in Chapter 7, in which the terminology is slightly changed for comprehensibility.

5.9 Conclusion

From literature we can infer that plenty of metrics exist for checking the conformance of process models with reality or for measuring the performance of discovery algorithms. However, choosing the right process discovery technique and its specific assumptions can be cumbersome for companies that have dynamic and rapidly changing processes. Moreover, the resulting process models are not suitable for or aimed at describing objectively the behaviour that is inherent to the event log. Therefore, log-based process metrics are presented in this chapter, which give a company an objective indication of the behaviour in the event log. The presented metrics are structured along two dimensions, which are time and structuredness, and are calculated on one of the following levels of analysis: log, case, trace, activity, resource, or resource-activity. These metrics provide business people with an objective start to look at their processes and are all implemented in the R-package edeaR, making them easy to apply to any event log. Moreover, to make the results of the metrics more accessible and understandable for business people, a dashboard including visualisations of each metric has been developed.

Although the metrics comply with the artifact requirements, some challenges and different perspectives can provide an even better indication of the process behaviour observed in an event log. For example, indicators or metrics should not be considered to be independent from each other and the results of one metric can be the input of or complement other metrics as stated by Heckl and Moormann [66]. This is an interesting recommendation for future research.

Chapter 6

Batch organisation of work metrics

6.1 Introduction

The reseach on this topic aims to gain insights in batching behaviour of resources in business processes from event logs. Resource behaviour was already introduced in the metrics concerning resources in Chapter 5. Resources, such as process participants, software systems, or equipment [47], are assigned to activities and typically carry these out on multiple cases such as files or customers. Assuming that arriving cases are handled immediately when the resource becomes available can be an undue simplification of reality. Employees might deem it more efficient to accumulate files and treat the entire stack later or machines can process multiple products at the same time. This type of resource behaviour is referred to as batch processing.

Batch processing influences the performance of a process as it can, for instance, lead to longer waiting times for certain cases when multiple cases are gathered before processing starts [184]. It has therefore been mentioned as one of the useful event log knowledge insights in the requirements of the operational excellence methodologies, as was discussed in Chapter 4. Consequently, it should be taken into account when modeling and evaluating business processes, as was also illustrated by van der Aalst et al. [163]. To this end, insights in batching behaviour should be generated, which is the topic of this chapter.

While the occurrence of batch processing might be readily observable for *passive resources* such as machines, it is typically less straightforward to determine how human resources, which are *active resources* according to Dumas et al. [47], organise their work. Direct observation of staff members' behaviour has limitations as it is both time-consuming and the Hawthorne effect can cause observed behaviour to deviate from real behaviour when humans know they are being observed [100]. Consequently, investigating the use of more readily available information sources, such as event logs created by process-aware information systems, is valuable.

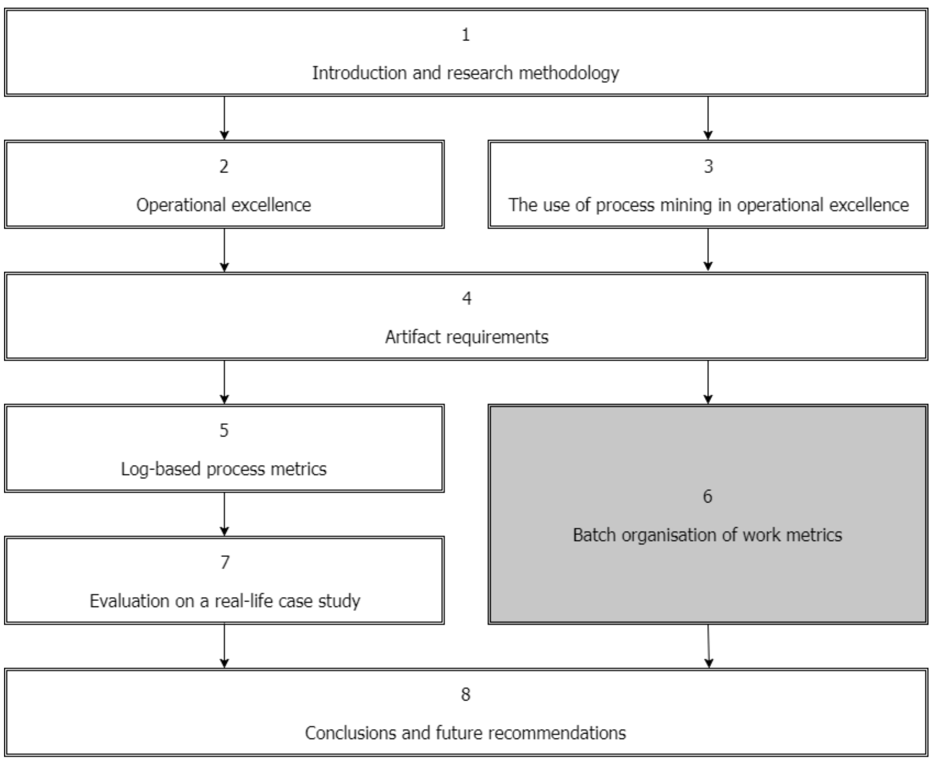

Figure 6.1: Outline of the thesis - Chapter 6.

This chapter[1] (Figure 6.1) focuses on retrieving event log insights on batch processing, which is marked as a research gap by Martin et al. [97]. More specifically, the key contributions of this research on batch processing are threefold. Firstly, the

[1]This chapter is based on *Martin, N., Swennen, M., Depaire, B., Jans, M., Caris, A., Vanhoof, K. 2015. Batch processing: definition and event log identification. CEUR Workshop Proceedings 1527, 137-140* [98] and *Martin, N., Swennen, M., Depaire, B., Jans, M., Caris, A., Vanhoof, K. 2017. Retrieving batch organisation of work insights from event logs. Decision Support Systems 100, 119-128.* [99].

concept of batch processing is elaborated upon and three types of batch processing are distinguished and formally defined. Secondly, a resource-activity centered approach is presented to identify these batch processing types from an event log and the Batch Organisation of Work Identification algorithm (BOWI) is presented. This algorithm provides useful insights in the batch organisation of work and its influence on process execution. Finally, based on the previous steps, a list of batch processing metrics are defined to gain insights in the characteristics of the identified batches and the implications of batch processing on process execution. These metrics enable companies to, amongst others, investigate the phenomenon of batch processing in their processes and judge its desirability. As has been done for the complete dissertation, the research in this chapter has been conducted following the principles of design science research [78], which was already elaborated upon in Chapter 1.

The chapter is structured as follows. Section 6.2 starts with related work that has been found in literature. Next to this, a running example is presented in Section 6.3 and the three types of batch processing are outlined in Section 6.4. After that, the BOWI-algorithm is outlined in Section 6.5 followed by a presentation of the developed batch processing metrics in Section 6.6. The presented algorithm and metrics are evaluated on both artificial and real life data in Section 6.7. Finally, limitations of the BOWI-algorithm are given in Section 6.8 and conclusions and future research challenges are included in Section 6.9.

6.2 Related work

The batch processing metrics that are presented in this chapter are based on a threefold definition of batch processing as a distinction is made between simultaneous, concurrent, and sequential batch processing. While Wu [184] and Pufahl and Weske [124] distinguish between the parallel and sequential execution of activities, other researchers only consider simultaneous batch processing [121, 178]. Consequently, this chapter presents a more versatile perspective on batch processing. As will be declared in Section 6.4, the threefold specification of batch processing also presents richer information than the more generic definition commonly used in operations management literature, as insights are conveyed on how batches are processed.

Batch processing is studied in several domains, but mainly within the field of operations management, with a key focus on topics such as order batching [67], scheduling [21, 120] and operational excellence. In the operational excellence field, the principles of lean management indicate that batch processing should be avoided. In contrast to single-piece flow, batch processing will lead to excess inventory and long queue times between production steps, which can be seen as forms of waste [12, 34]. While batch processing is mainly beneficial for the producer, single-piece flow focuses on the added value for the customer [51]. Moreover, even when single-piece flow fits the goals of the company, the appropriate batch size is often greater than one as setup costs and time needs to be taken into account [11, 59]. The trade-off between execution costs on the one hand and waiting costs on the other hand is also explicitly recognised in business process management literature [90, 121, 122, 123]. Finally, this trade-off is also stressed by Dobson et al. [44], who optimise the timing and number of batches for compounded sterile products in the health care industry.

Within the process modeling and execution domain, Pufahl and Weske [124] specify the concept of batch activities. Specification parameters such as the batch activation rule are identified. While Pufahl and Weske [124] focuses on a single batch activity, Pufahl et al. [123] extends these concepts to batch regions. The latter are a series of model constructs such as activities that handle cases in a batch. Recently, batch processing is studied for activities that are included in different processes by means of object life cycles [125]. As Pufahl and Weske [124], Pufahl et al. [123] and Pufahl and Weske [125] primarily focus on the activity level, their works do not explicitly take into account that the organisation of work for a particular activity can differ among resources. The work presented in this chapter includes this perspective by considering the resource-activity level as the key level of analysis. This is consistent with Liu and Hu [90], who recognise that batching strategies can differ among resources. While the key focus of Pufahl et al. [123] and Pufalh and Weske [124, 125] is on process modeling and the specification of the execution semantics of batch activities, Pufahl et al. [121] focuses on performance evaluation of batch activities. Solely considering the simultaneous batch processing case, cost functions are defined for both service and waiting costs and an analytical solution is proposed making use of queuing theory. This way, the benefits of introducing simultaneous batch processing can be quantified and a recommended batch size can be calculated. However, the suggested approach using queuing theory focuses on a single activity which, moreover, must fulfil the conditions of a particular queuing model [121]. As follows from the above discussion, related work tends to focus on modelling batch processing at design time. However, Pufahl et al. [122] suggest an approach to dynamically adjust the configuration parameters of batch activities depending on, e.g., the planned maintenance of a machine. This more flexible perspective on batching is also utilised by Pufahl and Weske [125], where a set of cases that might be batched are proposed to the resource, without the obligation to perform batch processing in practice.

Besides Wen et al. [178] and Nakatumba [108] as notable exceptions, no research attention is devoted to batch processing within the process mining field. Wen et al. [178] consider the problem of mining the process control-flow when the process under consideration contains activities where simultaneous batch processing occurs. For these activities, the authors assume that, for a particular batch, events are only logged for one of the cases in this batch. This is similar to Liu and Hu [90], who temporarily merge batched cases and decompose them afterwards. In contrast, the work in this chapter assumes that events are recorded for all individual cases in a batch. Wen et al. [178] developed a method that aims to add the missing events of cases in a batch, after which, e.g., existing control-flow discovery algorithms can be applied to the complemented log. The latter can also be used to apply the BOWI-algorithm presented in this chapter.

Nakatumba [108] proposes a method to identify batch processing in which all resource actions, i.e., executions of activities, are placed on a timeline and grouped in so called chunks. A new chunk is started when the elapsed time between the end of an action and the start of the following action exceeds one hour. When a period such as a working day is composed of multiple chunks, Nakatumba [108] states that batch processing occurs. This chapter extends the work of Nakatumba [108] in several ways. Firstly, in contrast to Nakatumba [108], the work in this chapter does not make abstraction from the difference between activities, reflecting the fact that some activities might be more eligible for batch processing. Secondly, the arbitrary delay of one hour between periods of activity is replaced by a formal definition of several types of batch processing. Finally, this chapter complements the work of Nakatumba [108] by distinguishing between batch processing and regular queue handling.

6.3 Running example

Throughout this chapter, a running example, other than the one that is used in Chapter 5, will be used for illustrative purposes. The process model, annotated with all assumed parameters, is visualised in Figure 6.2. Case interarrival times are assumed to follow an exponential distribution and activity durations are expressed in minutes and follow a triangular distribution.

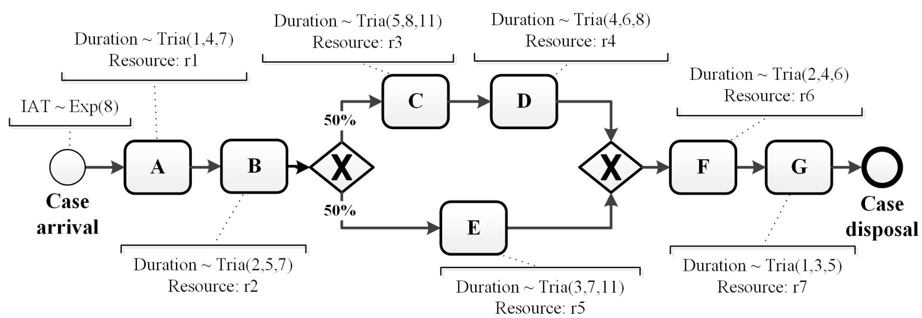

Figure 6.2: Process model of the running example.

6.4 Three types of batch processing

Batch processing is defined as a type of work organisation in which a resource executes a particular activity on multiple cases simultaneously or concurrently, or intentionally defers activity execution to handle multiple cases (quasi-)sequentially. Consistent with Martin et al. [98], a distinction is made between three types of batch processing: simultaneous, concurrent, and sequential batch processing. To exemplify the difference between these types, Figure 6.3 depicts the activities executed for two cases. Note that an activity will always be executed by the same resource for both cases. This section defines the three batch processing types as follows;

- **Sequential batch processing.** Activity instances are in a sequential batch when a resource intentionally defers the execution of this activity on distinct cases, after which they are handled (almost) immediately after each other. Consequently, all cases included in a batch need to be present at the activity under consideration before the resource starts processing the batch's first case. The latter distinguishes sequential batch processing from mere queue handling, stressing its intentional nature. For instance, employees can reply to e-mails twice a day, treating all available e-mails sequentially. In case both instances of activity A in Figure 6.3 were present before t_0, they are a sequential batch as the start time of the second case corresponds to the completion time of the first case.

- **Simultaneous batch processing.** Activity instances are in a simultaneous batch when they are executed by the same resource for distinct cases at exactly the same time. For example, several car parts that need to be painted in the same color can be placed in a spray booth together. In Figure 6.3, the two instances of activity B form a simultaneous batch as both start and completion times correspond across the two instances.

- **Concurrent batch processing.** Activity instances are in a concurrent batch when they are executed by the same resource for distinct cases partially overlapping in time. For example: a clerk can start booking a second invoice when additional information is required to finalise the first one. In Figure 6.3, instances of activities C, D, E, F, and G illustrate different types of concurrent batch processing.

The above batch processing types are largely consistent with the work of Wu [184], where simultaneous and sequential batch processing correspond to the concepts of parallel and serial process batches, respectively. Concurrent batch processing is not included in the work of Wu [184]. In operations management literature, batch processing is commonly referred to as the intermittent production of a particular type of product [107, 150], where production volumes are situated between a job shop setting with small volumes and mass production [107].

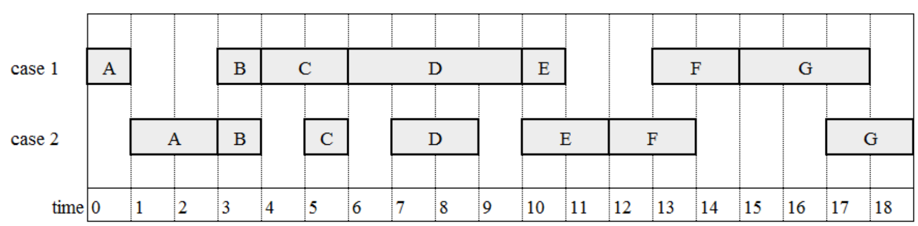

Figure 6.3: Conceptual representation of two cases (A: sequential batch, B: simultaneous batch, C-G: different types of concurrent batches).

6.5 Batch organisation of work identification algorithm

This section proposes the Batch Organisation of Work Identification algorithm (BOWI). The algorithm aims to generate purposeful insights from an event log regarding the batch organisation of work in a business process. A general overview is presented in Section 6.5.1. Afterwards, the algorithm is presented in more detail by respectively detailing the event log requirements (Section 6.5.2), the activity log creation (Section 6.5.3), the resource-activity matrix structure (Section 6.5.4), and the batching matrix structure (Section 6.5.5). Based on this, the batch organisation of work metrics can be identified in Section 6.6.

6.5.1 General overview

As shown in Figure 6.4, which presents an overview of BOWI, the input for the algorithm is an event log. This event log, consisting of atomic events, is converted to an activity log by mapping start events to their accompanying complete events. The activity log is restructured in a resource-activity matrix (RAM), in which each cell contains activity instance information of a particular resource-activity combination. Using the RAM as an input, a batching matrix (BM) is created for each batch processing type specified in Section 6.4. A BM mimics the structure of the RAM, but groups activity instances in the corresponding RAM cell based on the definition of the batch processing type under consideration. Activity instances that fulfil the conditions of this batch processing type are combined in a set, represented as a set of cases. Using this information, batch processing metrics such as the frequency of batch processing and the batch size can be calculated.

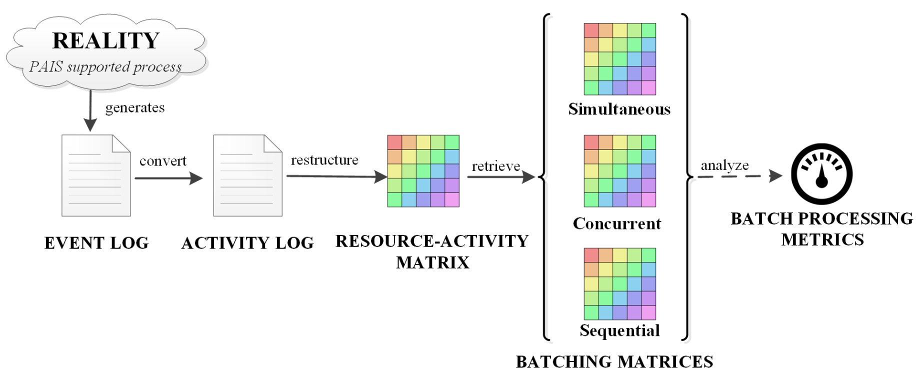

Figure 6.4: Overview of BOWI.

6.5.2 Event log requirements

BOWI requires an event log, composed of ordered events related to a particular case and activity, as input. For each event, the timestamp, executing resource, and transaction type needs to be recorded. Two transaction types have to be registered for BOWI: start and complete, both transitions of the XES lifecycle extension [64]. Moreover, each start event should have an accompanying complete event with the same resource being associated to both events. Note the requirement of matching start and complete events is in accordance with the notion of consistent traces, presented by Leemans et al. [88].

Building on the notation used by van der Aalst [155], the event log characteristics that are required to use BOWI can be outlined as follows:

Definition 9 (BOWI event log characteristics)**.** *Let $\mathscr{E}$ be the set of all events included in event log E. Moreover, let $\forall e \in \mathscr{E}$:*

- $\#_{case}(e)$ *represents the case associated to event e*
- $\#_{activity}(e)$ *represents the activity associated to event e*
- $\#_{resource}(e)$ *represents the resource associated to event e*
- $\#_{time}(e)$ *represents the timestamp associated to event e*
- $\#_{trans}(e)$ *represents the transaction type associated to event e*

Then, in this chapter, $\forall e \in \mathscr{E} : \#_{case}(e) \neq \perp \wedge \#_{activity}(e) \neq \perp \wedge \#_{resource}(e) \neq \perp \wedge \#_{time}(e) \neq \perp \wedge \#_{trans}(e) \in \{start, complete\}$*, where $\perp$ represents a null value. Moreover, every start event should have an accompanying complete event, i.e.,* $\forall e_1 \in \mathscr{E}, \exists e_2 \in \mathscr{E} : \#_{case}(e_1) = \#_{case}(e_2) \wedge \#_{activity}(e_1) = \#_{activity}(e_2) \wedge \#_{resource}(e_1) = \#_{resource}(e_2) \wedge \#_{time}(e_1) \leq \#_{time}(e_2) \wedge \#_{trans}(e_1) = start \wedge \#_{trans}(e_2) = complete$*. When $|e_2| > 1$, it is required that $|e_1| = |e_2|$.*

When considering the running example introduced in Section 6.3, Table 6.1 illustrates the event log structure. Each line in the event log represents a particular event in the process.

Table 6.1: Illustration of the event log structure.

case id	**timestamp**	**activity**	**transaction type**	**resource**
...	...	...	...	...
22	03/01/2016 11:14:41	E	start	r5
22	03/01/2016 11:22:37	E	complete	r5
25	03/01/2016 11:22:37	E	start	r5
34	03/01/2016 11:22:54	C	start	r3
42	03/01/2016 11:25:17	A	start	r1
34	03/01/2016 11:28:02	C	complete	r3
42	03/01/2016 11:31:58	A	complete	r1
25	03/01/2016 11:32:18	E	complete	r5
...	...	...	...	...

6.5.3 Activity log creation

The event log, fulfilling the requirements of Definition 9, is composed of atomic events. To retrieve batch processing insights, the event log is converted to an activity log containing information on activity instances, i.e., information on the execution of a particular activity by a particular resource on a particular case. To this end, each start event is mapped on its corresponding complete event, i.e., the complete event that is associated to the same case, activity, and resource in the event log. When multiple start and complete event are present for a particular case, activity, and resource combination, the first occurring unmapped start event will iteratively be mapped to the first occurring unmapped complete event. The activity log obtained from the event log excerpt in Table 6.1 is shown in Table 6.2.

Definition 10 (Activity log)**.** *Let L be an activity log based on event log E. Then A is composed of a set of activity instances $\mathscr{A}$. Each activity instance $i \in \mathscr{A}$ depicts the execution of an activity a by resource r on case c, started at time τ_{start} and completed at time $\tau_{complete}$. Activity instance i is represented by $\eta_i = (c, a, r, \tau_{start}, \tau_{complete})$, where $\#_n(i)$ represents the value of attribute n for activity instance i as suggested for events in Definition 9. All activity instances in L are sorted according to τ_{start}, i.e., $\forall \eta_i, \eta_{i+1} \in L : \eta_{i,\tau_{start}} \leq \eta_{i+1,\tau start}$.*

Table 6.2: Illustration of an activity log.

case id	**activity**	**resource**	$\tau_{\mathbf{start}}$	$\tau_{\mathbf{complete}}$
...	...	...	...	...
22	E	r5	03/01/2016 11:14:41	03/01/2016 11:22:37
25	E	r5	03/01/2016 11:22:37	03/01/2016 11:32:18
34	C	r3	03/01/2016 11:22:54	03/01/2016 11:28:02
42	A	r1	03/01/2016 11:25:17	03/01/2016 11:31:58
...	...	...	...	...

6.5.4 Resource-activity matrix

As the batch organisation of work reflects the way in which resources execute a particular activity, the activity log is restructured into a resource-activity matrix (RAM). Each cell in the RAM contains the activity instances associated to a particular resource-activity combination. An excerpt of the *r5-E* RAM cell from the running example is shown in Table 6.3.

Definition 11 (Resource-activity matrix)**.** *Let RAM represent the resource-activity matrix and let $RAM(a, r)$ be the cell of RAM related to activity a and resource r. Then* $RAM(a, r) = \{\eta \in L \mid \#_{activity}(\eta) = a \wedge \#_{resource}(\eta) = r\}$.

Table 6.3: Illustration of RAM cell r5-E.

case id	$\tau_{\mathbf{start}}$	$\tau_{\mathbf{complete}}$
...	...	...
22	03/01/2016 11:14:41	03/01/2016 11:22:37
25	03/01/2016 11:22:37	03/01/2016 11:32:18
27	03/01/2016 11:52:03	03/01/2016 12:03:01
28	03/01/2016 12:03:01	03/01/2016 12:11:51
...	...	...

To prepare the *RAM* for the analysis, self-loops are removed. A self-loop, as defined in Chapter 5, refers to the repeated execution of a particular activity by a particular resource on the same case immediately or almost immediately after each other. Consider for example that the activity instances in Table 6.4 are contained in RAM cell *r7-G*. These instances immediately follow each other and are related to the same case, i.e., case 63. When instances that do not immediately follow each other are still considered to be in a self-loop, a value for the maximal time tolerance between the instances (ω) should be specified. When such a tolerance is specified, it should be checked that no resource action is recorded between the instances contained in a self-loop.

Self-loops are not consistent with the definition of batch processing in general, and sequential batch processing in particular as it focuses on activity execution on distinct cases. Consequently, for the research in this chapter, they are removed by replacing them by a single activity instance with τ_{start} the start timestamp of the first self-loop instance and $\tau_{complete}$ the complete timestamp of the last self-loop instance. The self-loop in Table 6.4 is replaced by an instance with τ_{start} set to '05/01/2016 14:22:09'and $\tau_{complete}$ to '05/01/2016 14:31:03'.

Definition 12 (Self-loop removal)**.** *Given the activity instances contained in $RAM(a, r)$, a self-loop S is a set of activity instances, such that $\forall \eta_i, \eta_{i+1} \in S$, the following conditions cumulatively hold:*

- $\#_{case}(\eta_i) = \#_{case}(\eta_{i+1})$
- $\big(\#_{\tau_{start}}(\eta_{i+1}) - \#_{\tau_{complete}}(\eta_i)\big) \in [0, \omega]$
- $\nexists e \in E : \#_{resource}(e) = \#_{resource}(\eta_i) \wedge \#_{\tau_{complete}}(\eta_i) \leq \#_{time}(e) \leq \#_{\tau_{start}}(\eta_{i+1})$

When n represents the number of activity instance in S and η_1 is the first activity instance in S, then S is replaced in $RAM(a,r)$ by $\eta_{new} = \big(\#_{case}(\eta_1), \#_{\tau_{start}}(\eta_1), \#_{\tau_{complete}}(\eta_n)\big)$.

As indicated earlier, the ω value reflects the maximum tolerated time gap between the instances in a self-loop. It is a parameter that can either be set based on domain knowledge or derived from data using a procedure similar to the one outlined for a similar parameter (γ) in Section 6.5.5.

Table 6.4: Illustration of a self-loop in RAM cell r7-G.

case id	$\tau_{\textbf{start}}$	$\tau_{\textbf{complete}}$
...	...	...
63	05/01/2016 14:22:09	05/01/2016 14:25:41
63	05/01/2016 14:25:41	05/01/2016 14:29:17
63	05/01/2016 14:29:17	05/01/2016 14:31:03
...	...	...

6.5.5 Batching matrices

In general, a batch is a set of activity instances. As mentioned in Section 6.4, three types of batch processing can be identified: simultaneous, concurrent and, sequential batch processing. Consequently, using the RAM as input, the BOWI-algorithm creates a separate batching matrix (BM) for each batch processing type. The structure of these BMs mimics the one of the RAM, i.e., it is specified at the resource-activity level. Taking a RAM cell as input, activity instances are grouped in a set based on the conditions of the batch processing type under consideration. These instance sets are recorded in the corresponding cell of the BM. Hence, a BM is created by parsing the RAM once and combining activity instances using the definition of the batch processing type under consideration. Note that instances that do not satisfy the definition are recorded as a singleton set.

Definition 13 (Batch). *A batch b is a set of activity instances $\eta \in L$, for which $\forall \eta_i, \eta_j \in b : \#_{activity}(\eta_i) = \#_{activty}(\eta_j) \wedge \#_{resource}(\eta_i) = \#_{resource}(\eta_j)$, i.e., all instances in b originate from a particular cell $RAM(a, r)$ in the RAM.*

The definitions of the three BMs can, consistent with Section 6.4, be formalised as follows:

Definition 14 (Simultaneous batching matrix). *Let BM_{sim} represent the simultaneous batching matrix and let $BM_{sim}(a, r)$ be the cell of BM_{sim} related to activity a and resource r. Then $BM_{sim}(a, r)$ consists of a set of batches $\mathscr{B}$. When b represents a batch in $\mathscr{B}$, then $\forall \eta_i, \eta_j \in b : \#_{\tau_{start}}(\eta_i) = \#_{\tau_{start}}(\eta_j) \wedge \#_{\tau_{complete}}(\eta_i) = \#_{\tau_{complete}}(\eta_j)$ $(\forall b \in \mathscr{B})$. Moreover, $\forall b_i, b_j \in \mathscr{B} : b_i \cup b_j \notin \mathscr{B}$, i.e., any combination of batches in $BM_{sim}(a, r)$ does not fulfil the aforementioned conditions.*

Definition 15 (Concurrent batching matrix). *Let BM_{conc} represent the concurrent batching matrix and let $BM_{conc}(a, r)$ be the cell of BM_{conc} related to activity a and resource r. Then $BM_{conc}(a, r)$ consists of a set of batches $\mathscr{B}$. When b represents a batch in $\mathscr{B}$, then $\forall \eta_i, \eta_{i+1} \in b : \#_{\tau_{start}}(\eta_i) \leq \#_{\tau_{start}}(\eta_{i+1}) < \#_{\tau_{complete}}(\eta_i) \wedge \big(\#_{\tau_{start}}(\eta_i) \neq \#_{\tau_{start}}(\eta_{i+1}) \vee \#_{\tau_{complete}}(\eta_i) \neq \#_{\tau_{complete}}(\eta_{i+1})\big)$ $(\forall b \in \mathscr{B})$. Moreover, $\forall b_i, b_j \in \mathscr{B} : b_i \cup b_j \notin \mathscr{B}$, i.e., any combination of batches in $BM_{conc}(a, r)$ does not fulfil the aforementioned conditions.*

While the formalisation of the simultaneous and concurrent BMs directly follows from the definition of the respective batch processing type in Section 6.4, the specification of the sequential BM is subject to more restrictions. Firstly, the definition indicates that the elapsed time between the complete timestamp of a case and start timestamp of the next case in a batch should be lower than the parameter γ. The minimum value of γ is zero, indicating that cases are only batched when they immediately succeed each other. This value might be too rigid as, e.g., some set-up time might be required to open a new file when the previous one is processed, requiring a strictly positive value for γ. However, γ should be small to remain consistent with the idea of batch processing. Moreover, no other resource activity can be recorded between activity execution on cases in a sequential batch.

Secondly, to integrate the distinction between sequential batch processing and regular queue handling, a function ϕ is introduced. This function returns the time at which a particular case arrives at the activity under consideration. Case arrival can be approximated by the end of the preceding activity, e.g., in the process visualised in Figure 6.2, arrival at activity D can be proxied by the completion of activity C. Consequently, the preceding activity needs to be known, which is a process notion that can be retrieved using domain knowledge or through the application of a control-flow discovery algorithm on the event log. An overview of the latter is presented by De Weerdt et al. [42] and van der Aalst [155]. Given the large body of research on control-flow discovery, the operationalisation of ϕ is not treated in this work.

Thirdly, it is indicated that none of the cases can be included in a simultaneous or concurrent batch for the activity under consideration. The latter avoids that sequences of the latter two batch types are treated as a sequential batch.

Finally, if multiple cases arrive at the same time at the activity under consideration, they can only form a sequential batch when the first case in this batch is not processed (quasi-)immediately upon arrival. This situation can, for example, be relevant when the activity preceding the activity under consideration is executed in a simultaneous batch.

Definition 16 (Sequential batching matrix)**.** *Let BM_{seq} represent the sequential batching matrix and let $BM_{seq}(a,r)$ be the cell of BM_{seq} related to activity a and resource r. Then $BM_{seq}(a,r)$ consists of a set of batches $\mathscr{B}$. When b represents a batch in $\mathscr{B}$, then $\forall \eta_i, \eta_{i+1} \in b$, the following conditions cumulatively hold:*

- $\left(\#_{\tau_{start}}(\eta_{i+1}) - \#_{\tau_{complete}}(\eta_i)\right) \in [0,\gamma]$, *with* $\gamma \geq 0$
- $\nexists e \in E : \#_{resource}(e) = \#_{resource}(\eta_i) \wedge \#_{\tau_{complete}}(\eta_i) \leq \#_{time}(e) \leq \#_{\tau_{start}}(\eta_{i+1})$
- $\phi(\eta_i) \leq \#_{\tau_{start}}(\eta_1) \wedge \phi(\eta_{i+1}) \leq \#_{\tau_{start}}(\eta_1)$, *where η_1 represents the first processed case in b and $\phi(\eta_x)$ is a function returning the arrival time of case $\#_{case}(\eta_x)$ at activity a*
- $\eta_i, \eta_{i+1} \notin \left\{b' \mid (b' \in BM_{sim} \vee b' \in BM_{conc}) \wedge |b'| > 1\right\}$, *with $|b'|$ expressing the number of activity instances included in batch b'*
- *when* $\phi(\eta_i) = \phi(\eta_{i+1}) = \phi(\eta_1)$, *then* $\#_{\tau_{start}}(\eta_1) > \phi(\eta_i) + \gamma$, *with* $\gamma \geq 0$

($\forall b \in \mathscr{B}$). Moreover, $\forall b_i, b_j \in \mathscr{B} : b_i \cup b_j \notin \mathscr{B}$, i.e., any combination of batches in $BM_{seq}(a,r)$ does not fulfil the aforementioned conditions.

Even though an appropriate value of γ will depend upon the process under consideration, a log-based recommendation for each activity is useful. To this end, a two-step approach is suggested. In a first step, a minimum and maximum potential value for γ are specified. As already indicated, the lowest possible value equals zero. The maximum value builds upon the intuition that the set-up time to start a new case is likely to depend on the activity duration. Consequently, a percentage of the median activity duration is considered as the highest possible value of γ. A default value of 5% is proposed to keep γ small.

The second step involves selecting a recommendation from the obtained range. For this purpose, additional insights from the event log are used. More specifically, the time differences between the complete and start timestamps of subsequent non-overlapping activity instances are calculated for the activity under consideration. From these time differences, only values that are lower than the upper bound specified in the first step are maintained. The median value of the remaining time differences is recommended as a γ value.

To illustrate the batch processing definitions, Definitions 14-16 are applied on the activity instances in the RAM cell *r5-E*, depicted in Table 6.3, showing that two sequential batches of size two are formed: one containing cases 22 and 25 and one consisting of cases 27 and 28. Consequently, the instances in Table 6.3 will lead to the following entries in the corresponding BM cells:

- $BM_{sim,r5-E}$: $\{..., \{c_{22}\}, \{c_{25}\}, \{c_{27}\}, \{c_{28}\}, ...\}$
- $BM_{conc,r5-E}$: $\{..., \{c_{22}\}, \{c_{25}\}, \{c_{27}\}, \{c_{28}\}, ...\}$
- $BM_{seq,r5-E}$: $\{..., \{c_{22}, c_{25}\}, \{c_{27}, c_{28}\}, ...\}$

The three batching matrices will form the basis for further analysis, as will be shown in Section 6.6.

6.5.6 Implementation

BOWI is fully implemented using R[2], a programming language for which a large set of packages is available which can be used to create application-specific functions. The key packages that are used are *dplyr* for data manipulations such as sorting and data summarisations, *lubridate* to work with timestamps, and *reshape* for converting the event log to an activity log.

The pseudocode for BOWI's batch identification component is given in Appendix C. It directly follows from the formalisation introduced in this section and shows that batches are identified from an activity log by parsing it once and comparing each line in the log with the prior one. In this way, the algorithm enriches the activity log with batch information by adding two columns: (i) a batch number, grouping activity instances that belong to the same batch, and (ii) the batch type, indicating which of the three batching types prevails. This is all information required

[2] https://www.r-project.org/

to create BMs and to calculate the metrics that are presented in the next section. Each metric is implemented as a separate function, which makes the framework easily extendable with additional metrics.

6.6 Batch organisation of work metrics

Using the information in the batching matrices as input, a list of batch processing metrics can be defined. These metrics, consistent with the list of log-based process metrics provided in Chapter 5, provide insight in the business value of batch processing. The batch processing metrics, of which an overview is given in Table 6.5, will be explained and illustrated in the remainder of this section. In accordance to the RAM, all presented metrics are defined on the resource-activity level. From this level, aggregations to other levels of analysis such as the activity level, the resource level, and the level of the entire event log can be derived.

6.6.1 Frequency of batch processing

To gain insights in the prevalence of batch processing, the frequency of batch processing can be considered. This metric presents an overview of the absolute and relative number of occurrences of the different batch types in the event log. In the simple example in Figure 6.3, the resource-activity level and the activity level correspond as a single resource is responsible to execute a particular activity on all cases. For instance, one simultaneous batch of activity B is performed by the same resource on the resource-activity level, which accounts for 100 % of the occurrences of activity B. Similarly, on the level of an activity, it can be of interest for a company to have an insight in the number of batches that occur for each type of activity, stating that invoices are more frequently processed together than phone calls, for example. At the resource level, a particular employee can be found to process much more work in batch than others, indicating that he has a tendency to wait until different jobs can be processed together. On the level of the entire event log, this metric is not useful for a company because activities in a process are usually too diverse to be compared to each other.

6.6.2 Batch size

Besides knowing how frequently batch processing occurs, the size of batches is another valuable metric. Building on the notion of the RAM, the batch sizes can be calculated

Table 6.5: Batch organisation of work metrics.

Metric	**Description**	**Levels of analysis**
Frequency of batch processing	The absolute and relative number of times that a set of $BM_x(a,r)$ contains two or more cases.	Resource-activity, Activity, Resource
Batch size	The summary statistics of the number of activity instances in each set of $BM_x(a,r)$, both including and excluding sets of size one.	Resource-activity, Activity
Number of cases included in a batch	The absolute and relative number of cases that appear in each set of $BM_x(a,r)$.	Resource-activity, Activity, Resource
Duration of activity instances in a batch	The summary statistics of the difference between the duration of the activity instances in each set of size two or more in $BM_x(a,r)$ compared to the duration for sets of size one.	Resource-activity, Activity, Resource
Waiting time of activity instances in a batch	The summary statistics of the difference between the waiting time of the activity instances in each set of $BM_x(a,r)$ compared to activity instances not in this set.	Resource-activity, Activity, Log
Overlap in concurrent batches	The summary statistics of the amount of time that the activities in a concurrent batch are actually concurrent.	Resource-activity, Activity, Resource, Log

on the resource-activity level. For example, a particular employee always processes five invoices in a sequential batch, while all other employees handle them as they arrive. Batch sizes can be calculated for each batch processing type for each resource-activity combination. As BMs also include sets of size one, indicating that a case is not included in a batch of this type, summary statistics can be calculated both including and excluding sets of size one. The latter can be useful as it states how large batches tend to be when multiple cases are combined. In the example in Figure 6.3, a simultaneous batch of size two for activity B is observed. The values on the resource-activity level can easily be aggregated to the activity level, enabling the comparison between activities. Because a resource can, e.g., handle phone calls in one activity and documents in another activity, an aggregation of this metric to the resource or event log level might not be valuable. Calculating for example an aggregated size of a batch of three phone calls and another batch of four invoices does not add value for the company.

6.6.3 Number of cases included in a batch

This metric combines the insights from the two preceding metrics, by presenting an overview of the number of cases that is included in each type of batch. Similarly, as an activity is always executed by the same resource in the running example in Figure 6.3, 100 % of the cases is included in a simultaneous batch of activity B executed by the same resource on the resource-activity level. At the level of a specific activity, it can be interesting to have an overview of how many times this activity is executed in batch. For example, of all invoices that the company processed during a certain period, 40 % is executed in a batch. At the resource level, it can be interesting to have an overview of which percentage of a certain employee's work is executed in a batch, e.g., a particular employee executes 60 % of his work in batch while other employees only execute 20 % of their work in batch. On the level of the entire event log, this metric is again not useful. An additional calculation derived from this metric can be performed for concurrent batches as activities in this type of batch processing are not all overlapping by definition. The number of cases actually overlapping in a concurrent batch, can therefore be of interest.

6.6.4 Duration of activity instances in a batch

It can be valuable to quantify the effect of batch processing on the duration of activities as resources might become more efficient when multiple cases can be dealt with, e.g., sequentially. To calculate the effect, the duration of the activities executed in batch is compared to the same activities performed by the same resource not in a batch. On the resource-activity level, an employee might need, e.g., 50 minutes to execute three invoices sequentially while he needs 20 minutes to execute one invoice separately. At the activity level, it can be interesting to have an overview of the duration of all activities of this activity type executed in a batch. For example, the average duration of an invoice processed in a batch is 40 minutes compared to an average duration of 45 minutes for all invoices processed at arrival time. At the resource level, it can be interesting to have an overview of how long it takes a certain employee to perform his work.

6.6.5 Waiting time of activity instances in a batch

While batch processing can improve the efficiency at a global level, individual cases possibly have to wait longer when batches are formed compared to a situation where batch processing is absent. The waiting time of an activity can be defined as the elapsed time between case arrival at activity and the start of its execution. At the resource-activity level, some employees may cause more waiting time when they intentionally process certain activities in a batch compared to other employees not doing this. Figure 6.3 illustrates that case 1 has to wait until t_3 before the execution of B starts, even though the execution of the prior activity ended at t_1. At the level of a specific activity, the waiting time when an activity type appears in a batch compared to the waiting time of all appearances of this activity not in a batch can be provided. Next to this, also the level of the entire event log can be interesting for a company. This provides an overview of how long cases, for example customer requests, should wait before they are handled in batch. A comparison with cases that are not handled in batch can provide the company with interesting insights about their process performance.

6.6.6 Overlap in concurrent batches

Another metric that can be of interest concerning the type of concurrent batch processing is the amount of overlap that occurs in a concurrent batch. There is a major difference if an employee only starts working on the next invoice 2 minutes before the finish of the other one, or 2 minutes after its start. Also, not all activities in a concurrent batch are overlapping with each other and their duration can differ. Therefore, the amount of overlap is calculated against the average duration of all activities in the concurrent batch. In the running example, we find for example that for activity G an overlap of 1 time unit against an average of 2.5 time units can be measured which gives an overlap of 40 %. Over all concurrent batches for this resource-activity combination, summary statistics such as the minimum, mean, median, and so on can be of interest. Next to this resource-activity level, this metric can also be calculated on the level of the resources and the activities. On the level of the entire event log, this metric is again not useful.

6.7 Evaluation

A twofold approach is used to evaluate both the algorithm and the presented metrics: Section 6.7.1 focuses on BOWI's ability to correctly rediscover batches in artificial event logs and Section 6.7.2 discusses the application of the algorithm and metrics on real-life logs.

6.7.1 Artificial event logs

6.7.1.1 Experimental design

BOWI's performance is evaluated by investigating its ability to rediscover known batches solely using an artificial event log. To this end, an artificial log is generated based on a generalised version of the process model in Figure 6.2. For each of the seven resource-activity combinations, it is randomly determined whether no, simultaneous, sequential or concurrent batching prevails with all options having the same probability. In the latter three cases, an integer batch size is randomly drawn from the set {2,3,4,5}. Given these inputs, the event log generator autonomously determines which cases are batched for each activity and generates a log considering 500 cases that enter the process. The data file also indicates which cases are grouped as a batch of a particular type. This information is only used for evaluation purposes and is removed from the event log that is provided to BOWI.

After executing BOWI on the event log, the algorithm's output is compared to the real batch composition. For a particular resource-activity combination, a case is correctly classified by BOWI when it is (i) contained in its correct batch in the BM of the batch type prevailing in reality, and (ii) included as a singleton in the BMs of the other two batch types. Consequently, the evaluation centers around the detection of errors, which are (i) cases that are included in a batch of the correct type but in the wrong composition, and (ii) cases being included in a batch of a particular type while they are not included in such a batch in reality. Using these conditions, the number of errors is calculated for each resource-activity combination. The first condition is defined rather rigorously as the composition of discovered batches has to be completely correct. For instance, when BOWI rediscovers a batch for all but one case, all cases in this batch are reported as errors because they are not part of the exact same batch prevailing in reality.

The aforementioned constitutes one experiment. To determine the number of experiments, an a priori power analysis for a one-sample Wilcoxon signed-rank test is conducted. To achieve a power value (i.e., the probability of rejecting the null hypothesis when it is false) of 0.80 [26] and given a family-wise significance level to 0.05 and effect size of 0.20 (the value proposed by Cohen [25] for the detection of small effects), the power analysis shows that at least 185 event logs need to be generated. Consequently, the number of artificial event logs is set to 200, which surpasses this lower bound.

6.7.1.2 Results

The application of the experimental design calculates, for each resource-activity combination in an event log, the number of errors. These results are aggregated by grouping resource-activity combinations in 12 classes, expressing a combination of the real batch type in the event log (no batching, simultaneous, concurrent, or sequential batching) and BOWI's output (simultaneous, concurrent, or sequential BMs). For each of them, a decimal error proportion is calculated by dividing the number of errors by the number of cases that are included in the real batches for that class.

Table 6.6 reports summary statistics on the error proportions detected for the 12 classes over all 200 event logs. With *'seq - seq'* and *'no batch - seq'* as an exception, all classes show that BOWI's output is free from errors. This confirms that BOWI can rediscover existing batches solely using the event log. Moreover, the algorithm does, e.g., not detect sequential batch processing when concurrent batch processing prevails.

Table 6.6: Summary statistics on the error proportion of BOWI's output.

Event log input -	Error proportion				
BOWI output	mean	sd	median	min	max
seq - seq	0.08	0.15	0.01	0.00	1.00
no batch - seq	0.54	0.16	0.57	0.13	0.82
all 10 other classes	0.00	0.00	0.00	0.00	0.00

Regarding BOWI's detection of sequential batch processing, errors are detected when either sequential batch processing prevails in reality or no batch processing takes place. For an event log in which sequential batch processing is introduced, BOWI does not rediscover the exact composition of these batches for, on average, 7.62 % of batched cases, with a standard deviation of 15.41 % point. These errors are fairly concentrated as an exact match, i.e., an error proportion of zero is present for 243 of the 352 observations (69.03 %). For the remaining 109 observations, several explanations for the observed deviations can be identified. When sequential batch processing is inserted in the event log for the first activity, no arrival proxy will be available in the resulting event log as no prior activity is present. Consequently, conditions related to the arrival proxy in Definition 16 cannot be checked, leading to a less stringent definition. This can cause multiple batches of a particular size that are executed one after another to be included as a single batch in BOWI's output. The same holds when the activity under analysis is preceded by an activity where simultaneous batch processing prevails with a higher batch size than the batch size for the activity under analysis. When the arriving simultaneous batch is processed immediately upon arrival, BOWI will detect, e.g., a batch of size four instead of two batches of size two. Even though this will be included as an error in Table 6.6, BOWI's output is a valid representation of business intuition in this case.

When no batch processing is included for a resource-activity combination in the event log, i.e., when all cases are expected to be included as a singleton in each of the BMs, the error proportion of BOWI is higher. The mean error proportion equals 54.08 % with a standard deviation of 16.20 % point and a median of 56.77 %. Studying the error proportion on the activity level for the *'no batch - seq'* situation shows that it is the highest for the start activity. This can, once again, be attributed to the less strict definition due to the absence of an arrival proxy. For the other activities, errors can be explained by the arrival of cases in, e.g., a simultaneous batch which is not handled immediately upon arrival. Even though it is recorded as an error, it presents a valid occurrence of sequential batch processing in a business context. Even when cases arrive separately, sequential batch processing can also be detected when long queues are formed. In this case, a subset of queueing cases fulfils the conditions of Definition 16. Despite the fact that Definition 16 aims to distinguish between regular queue handling and sequential batch processing, it should be noted that the definition aims to strike a balance between accuracy and clarity. Instead of enumerating and excluding all possible exceptions, leading to an incomprehensible definition, a limited set of understandable conditions is specified.

6.7.2 Real-life event logs

To demonstrate that BOWI can generate insights in batching behaviour in a real world business context, the algorithm is applied to real-life event logs from two different contexts: a call center and a production company.

6.7.2.1 Event log of a call center

BOWI is applied to a real-life event log, based on data of a bank's call center made available by the Technion Service Enterprise Engineering Center[3]. Incoming calls are directed to a voice response unit (VRU), where automated voice information guides the caller. When the VRU does not enable callers to service themselves, they are

[3] http://ie.technion.ac.il/Labs/Serveng

redirected to a queue, after which they are connected to an agent. After converting the dataset to an event log format, 34 resource-activity combinations are included. More specifically, the log contains the *VRU* - *Handling by VRU* combination and the activity *Handling by agent*, which is executed by 33 distinct staff members. The results reported in this section are based on an analysis of 169065 calls registered in the first semester of 1999.

Within the analysis set, batching behaviour is detected for 31 resource-activity combinations. For *Handling by VRU*, which is always handled by resource *VRU*, both concurrent and simultaneous batching is detected, with respectively 26 % and 0.33 % of all calls being batched. The significant number of calls handled concurrently is due to the VRU's design to handle multiple calls concurrently on different lines. Simultaneous batching is present to a far lesser extent as it requires that, by coincidence, multiple calls arrive at exactly the same time and require the same processing time.

For 30 out of 33 agents performing *Handling by agent*, batching behaviour is detected. Concurrent batching is present, but its prevalence is low as, on average, only 1.85 % of the calls are included in a concurrent batch. Sequential batch processing is also discovered, but to a far lesser extent with an average of 0.03 % of the calls belonging to a sequential batch. When focusing on concurrent batching, Table 6.7 summarises some of BOWI's metrics for the five agents handling calls concurrently the most often.

Table 6.7: BOWI metrics calculated for concurrent batching by five resources for activity *Handling by agent* in the call center event log.

agent	**frequency**	**batch size**		**# batched**	**duration (mean)***		**time overlap**
		mean	sd	**cases (rel.)**	batch	no batch	
SHARON	152	2.23	0.42	2.49	4.27	2.28	0.46
KAZAV	121	2.17	0.39	2.53	4.71	3.21	0.47
MORIAH	114	2.18	0.38	2.64	4.21	3.14	0.50
TOVA	107	2.16	0.39	2.54	4.32	2.84	0.47
STEREN	87	2.18	0.39	2.05	6.13	3.04	0.52

* *expressed in minutes*

Table 6.7 shows that, even for the agents for which concurrent batching is observed the most, the proportion of batched calls is rather limited as it ranges between 2.05 % and 2.53 %. The mean batch size varies between 2.16 and 2.23 calls. Hence, batching is not fundamentally integrated in the operations of a call center, which could be anticipated given its characteristics. Concurrent batching can take place when an agent already takes another call while the caller is looking for a particular document or the agent is awaiting input from the bank. This is supported by the fact that the mean duration tends to be longer for batched calls than for non-batched calls.

6.7.2.2 Event log of a production company

BOWI is also applied to a real-life event log of a production process, which is available at the 4TU Data Center[4]. It contains process execution data for 225 cases undergoing activities such as *flat grinding* and *packing*. In the log, 27 distinct activities and 31 unique resources are included.

Applying BOWI shows that batch processing is detected for 29 of the 57 resource-activity combinations in the event log. More specifically, simultaneous, concurrent, and sequential batching is present for respectively, 9, 25, and 17 resource-activity combinations. This includes 14 resource-activity pairs for which both concurrent and sequential batches are present and 7 resource-activity pairs for which all batch types are detected.

Using the *number of cases included in a batch* metric, it is concluded that concurrent batch processing is the most prevalent. When considering all resource-activity combination where concurrent batch processing occurs, on average 23.50 % of all cases is batched. For simultaneous and sequential batching, this is 15.55 % and 11.31 % respectively. Consequently, the remainder of this discussion focuses on concurrent batching.

When concurrent batching occurs, an important part of the cases is batched. This indicates that batching is fundamentally integrated in the organisation's process. Table 6.8 summarises some BOWI metric values for the five resource-activity combinations for which the highest number of concurrent batches is detected. For these resource-activity combinations, the proportion of cases being part of a concurrent batch ranges from 28 % to 77 %. The batch sizes are situated between 2.34 and 3.33, with standard deviations between 0.61 and 2.11. The influence of batch processing on activity duration outlined in literature does not hold as batched cases tend to take longer than non-batched cases. It might be the case that batching takes place for a particular type of product, which requires less intensive processing. Concerning the difference in waiting times between batched and non-batched cases, the results are mixed depending on the resource-activity combination. From the time overlap

[4] http://data.4tu.nl/repository/uuid:68726926-5ac5-4fab-b873-ee76ea412399

metric, it follows that there is a significant overlap between concurrently handled cases. This indicates that genuine concurrent batch processing is detected, and not sequential batch processing with inaccurate timestamp registration.

Table 6.8: BOWI metrics calculated for concurrent batching for five resource-activity combinations from the production company event log.

res.-act. comb.*	**freq.**	**batch size (mean)**	**# batched cases (rel.)**	**duration (mean)**		**waiting time (mean)**		**time overlap**
				batch	no batch	batch	no batch	
1	121	3.33	0.77	2.23	1.27	22.92	48.11	0.86
2	118	2.83	0.66	1.74	1.27	18.83	17.54	0.79
3	61	2.36	0.39	2.31	1.45	45.90	58.37	0.65
4	34	2.47	0.34	6.07	5.88	7.15	3.35	0.51
5	29	2.34	0.28	6.45	5.11	5.77	9.09	0.50

** 1: Qual. Check 1 - Final Insp. Q.C., 2: Qual. Check 1 - Turn. & Mil. Q.C ,*
3: Machine 1 - Lapping, 4: Machine 4 - Turn. & Mil., 5: Machine 6 - Turn. & Mil.

6.8 Limitations

Despite BOWI's ability to mine and describe batching behaviour from an event log, some limitations need to be recognised. Firstly, the log should contain both start and complete events and resource information, which is often not the case in existing real-life event logs. Moreover, the level of detail at which timestamps and resources are recorded determines the granularity at which batching behaviour is identified. When, e.g., only resource classes are recorded, no distinction can be made between specific resources.

Secondly, BOWI does not explicitly consider the issue of noise in timestamp registration. Hence, it relies on accurate event registration for each case, which can require that a process is backed by a system which automatically logs resource action instead of relying on manual intervention to log events. Nevertheless, some features of BOWI should be highlighted related to inaccurate timestamp registration. For sequential batching, a time tolerance that is allowed between consecutive instances in a sequential batch can be specified. When the start and complete timestamps of cases in a simultaneous batch are not identical, BOWI will label it as a concurrent batch.

However, the value of the time overlap metric will show a high overlap, indicating that it might be an inaccurately recorded simultaneous batch.

Thirdly, the creation of an activity log requires mapping corresponding start and complete events. When a case passes a resource-activity combination multiple times, each start event is mapped to the first occurring unmapped complete event. When this mapping does not correspond to reality, it will influence batch detection as the activity log is its key input.

Finally, a case's arrival time at an activity is needed to distinguish sequential batching from regular queue handling. When this information is not included in the event log, it can be proxied by the completion time of the prior activity. However, this requires control-flow insights, i.e., the prior activity needs to be known, which is not trivial for complex processes. However, the absence of such a proxy does not impede BOWI from being applied, but renders the conditions to detect sequential batching less strict.

6.9 Conclusion

This chapter focuses on the retrieval of event log insights on batch processing. To this end, three types of batch processing, which are simultaneous, concurrent, and sequential batch processing, are defined (Section 6.4) and formalised (Section 6.5.5). Using these definitions, the Batch Organisation of Work Identification algorithm (BOWI) is developed to gather knowledge on batch processing from event logs (Section 6.5). The algorithm groups cases when they fulfil the conditions associated to a particular batch processing type. These case sets can be used as an input for the calculation of batch processing metrics with business value (Section 6.6). The presented algorithm is evaluated on artificial event logs (Section 6.7.1), showing that it can rediscover the prevailing batch size under most circumstances. Only when batch processing is absent for an activity in the artificial log or when sequential batch processing prevails, deviations between the expected output and BOWI's output are observed. However, these differences can be partly attributed to the comparison basis that is used and to the desire to maintain clarity in the operationalisation of the sequential batch processing definition. Besides an evaluation on artificial event logs, the BOWI-algorithm is also applied to real-life call center data and production company data as a proof-of-concept (Section 6.7.2).

Future work can extend the BOWI-algorithm to retrieve even more versatile batch processing insights from an event log. Firstly, while the BOWI-algorithm currently focuses on gathering batch processing knowledge on the level of a single activity, this perspective can be broadened by considering multiple consecutive activities. This phenomenon is recognised by Pufahl et al. [123], who model batch regions. Batch processing over multiple activities is also considered as a potential batch processing pattern by Liu and Hu [90] and Wen et al. [178]. Secondly, insights on the logic behind batch formation can be added as an analysis dimension. While BOWI currently aims to identify which cases are batched, it can be useful to identify the reasoning behind batching behaviour through the identification of batch activation rules. Batch formation can depend merely on the number of queueing cases or can be contingent on, e.g., the time of day or case attributes. Note that grouping cases based on case attributes is consistent with the research of Pufahl et al. [122, 123] and Pufahl and Weske [125], where cases are batched which have identical values on a user-defined set of attributes. Similarly, Liu and Hu [90] and Liu et al. [91] group cases based on identical values for a pre-specified set of case characteristics.

Chapter 7

Evaluation on a real-life case study

7.1 Introduction

To demonstrate the applicability of the presented log-based process metrics in Chapter 5, and their added value in the light of operational excellence, they are applied to a real-life case study of a Belgian utilities company. A first evaluation of the developed artifacts was already performed in Chapter 5 and Chapter 6 by applying the metrics to an artificial event log. As the process under consideration does not meet the requirements to apply the batch organisation of work metrics which were introduced in Chapter 6, these metrics will not be applied to the case study in this chapter. However, these metrics have already been applied to datasets from two different contexts in the previous chapter.

Next to the application of the metrics to the dataset of the case study organisation, a dashboard is also created to visualise the metric results. All applied metrics and accompanying visualisations are discussed with the case study organisation, and small changes or additions to the metrics have been implemented in order to improve the applicability of the artifacts, according to the feedback mechanism in the design science research framework.

This chapter (Figure 7.1) is structured as follows. In Section 7.2, the company and the business process under analysis are introduced together with their requirements and the specifications of the dataset that will be analysed. Next, an overview of the log-based process metrics that have been applied to the dataset is given in Section 7.3.

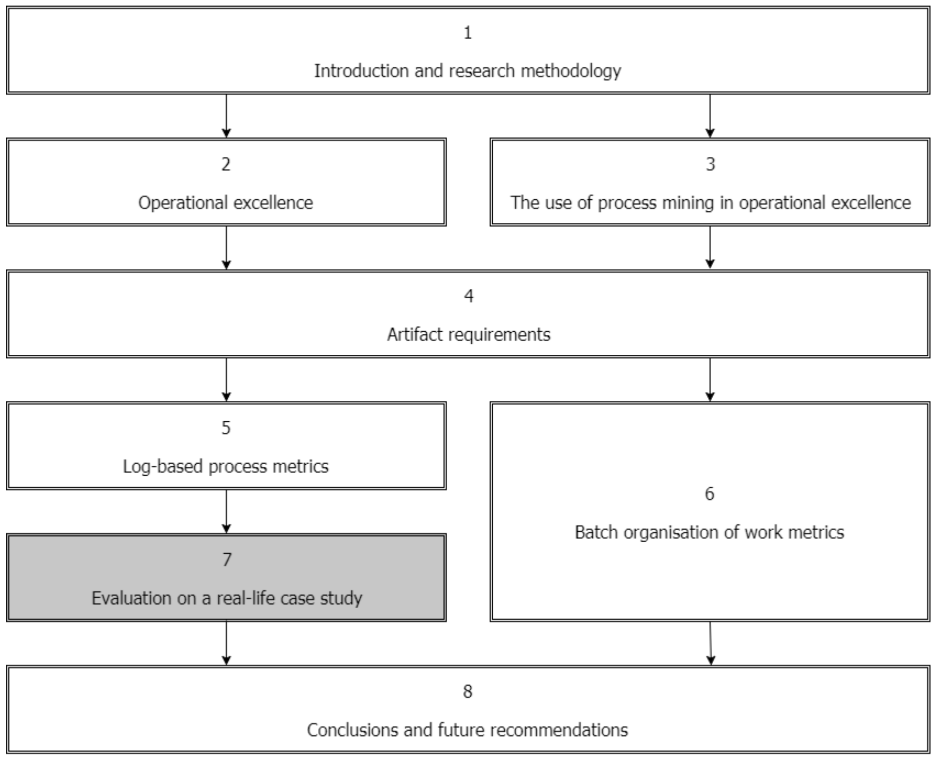

Figure 7.1: Outline of the thesis - Chapter 7.

Each metric that has been applied is analysed and the results are shown in print screens from the metric dashboard that was created for the company. Finally, the findings are discussed in Section 7.4 and conclusions are drawn in Section 7.5.

7.2 Overview of the case study company

The case study company is a Belgian utilities company that is responsible for eight different activities, among which the construction and daily operation of the distribution networks for different utilities, the creation of new connections and the adjustment of existing connections, the monitoring of the distribution to repair breakdowns, defects and leaks, and different social public service obligations.

The process under analysis presents the total flow from the request made by a customer to the aftercare which includes the invoicing. The process, which is shown in Figure 7.2, can be divided into six building blocks or subprocesses which are (i) capturing the customer request, (ii) a possible study in case this is necessary, (iii) the draft of a proposal, (iv) the preparation of the job, (v) the actual execution of the job, and (vi) the aftercare. Each of these building blocks or subprocesses contains one or more activities.

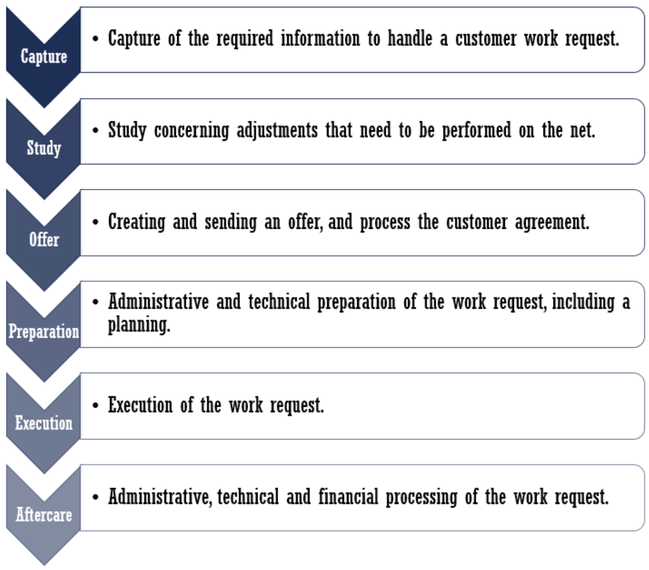

Figure 7.2: Overview of the case study process under analysis.

7.2.1 Company requirements

After interviewing two business process experts from two different levels in the organisation, the process that is analysed here could be defined as rather unstructured and full of exceptions. Moreover, different people working with and within the process have different knowledge about the steps in the process as they are mostly only concerned with their own part of the process. A lack of transparency and objectivity was also added as an argument. From these findings, it could be concluded that a link with the operational excellence principles is currently missing for this process.

Table 7.1: Translation of the process mining concepts within the case study.

Process mining concept	**Translation**
Case	Work request
Trace	Process variant
Activity	Activity execution
Resource	Resource or Role

Next to these conclusions, which could also be drawn from the interviews in Chapter 4, another result from the interviews was that one of the requirements for the developed metrics was a translation from process mining terminology to concepts that are accessible in business contexts. Some of the process mining concepts, such as traces, are therefore translated (e.g., to process variants) to be useful in the communication directed to the company. Table 7.1 shows the translation of the process mining concepts used in this case study. Furthermore, metrics are complemented with more business specific interpretations, if necessary. For example, the resource frequency metric is referred to as giving insights into the workload, indicating how the work is divided over the different resources.

7.2.2 Dataset

The data that was collected from the process contains information on six types of work requests and concerns connections for several types of utilities. The original dataset has a timespan of two years, from January 2014 until December 2015, and contains around 70 000 work requests per year, including unfinished work requests.

Figure 7.3: General descriptions of the dataset used.

However, for this analysis a subset of the data will be looked at. All work requests that started in April 2015 containing the “Register aftercare” activity are analysed by applying the metrics presented in Chapter 5. The metrics concerning the batch organisation of work, which was dealt with in Chapter 6 are not applied to this dataset because the event log does not meet all requirements. As shown in Figure 7.3 the dataset used contains 4 054 (finished) work requests (cases), 28 different activities, and 7 different resources (roles) executing the activities. Divided over 1 113 different process variants (traces), 42 091 activity executions have been performed, which is equal to the number of events, indicating that each activity execution only contains one event, which refers to the“completion” of the activity. An overview of the 28 different activities and 7 different roles within the dataset is provided in Table 7.2. Note that the resources executing the activities can be mapped one-on-one to the activities, implying that each resource is actually just a role defined by the company able to execute one or more specific activities.

7.3 Metrics applied to the dataset

Not all metrics presented in Chapter 5 are applicable to the dataset at hand. Figure 7.4 contains an overview of the metrics that have been applied and for which an interpretation and visual representation will be shown below. All figures added in this section are print screens taken from the dashboard that was presented in Chapter 5 and which was customised for the company.

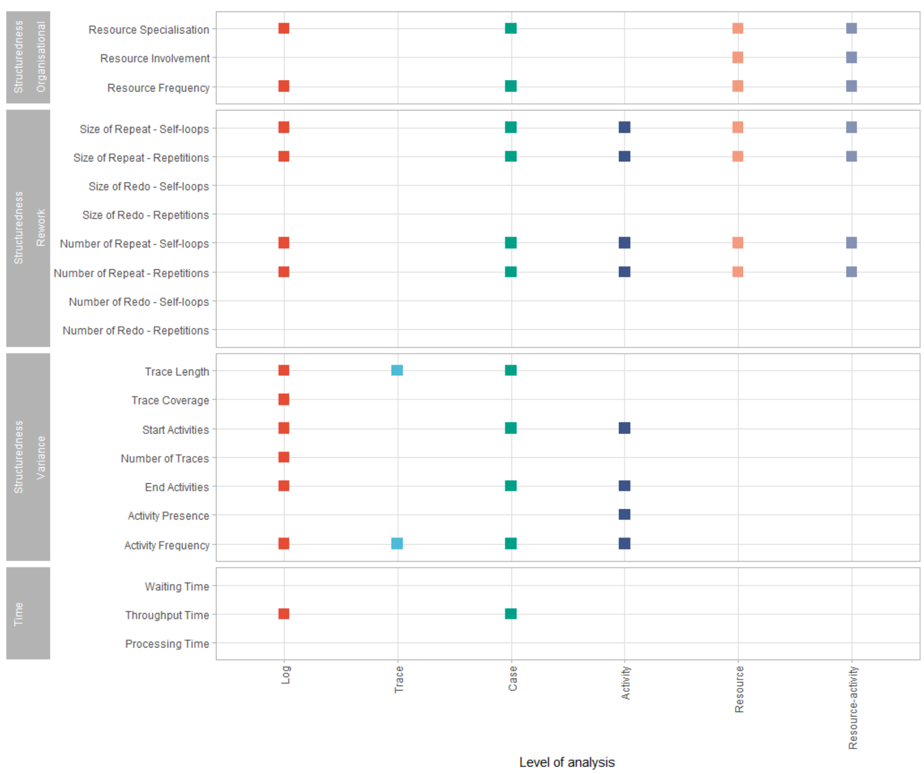

Figure 7.4: Overview of the metrics that have been applied to the case study.

Table 7.2: Activities used in the dataset, together with their executing role.

Activity	**Role**
Creation Work request	Work preparation
Gas request	/
Project request	Work preparation
Analyse customer request	Work preparation
Request EAN	Work preparation
Project info	Work preparation
Project advice	Technical customer advisor
Information customer	Work preparation
Project study	Team Net Development
Project visit	Technical customer advisor
Offer sent	Work preparation
Study	Team Net Development
Technical visit	Technical customer advisor
User agreement	Work preparation
Phone	/
Optimise file	Work preparation
Customer ready	Work preparation
Verification visit	Technical customer advisor
Road admission	Team Backoffice Nets and Studies
Net extension	Team Net Development
Measurement group file sent	MOTS
Mobility MOTS	MOTS
Execution	Work preparation
Measurement group file submitted	MOTS
Register aftercare	Aftercare
Physical document	Aftercare
Execution confirmation	Aftercare
Other	/

7.3.1 Time metrics

Because the dataset at hand only contains one timestamp for each activity, i.e., the completion time, only the throughput time can be calculated from the time metrics. In order to calculate the other time metrics, which are the waiting time and the processing time, each activity requires at least a start and an end timestamp. All durations are expressed in days for this case study.

- *Throughput time.* The throughput time of a work request is the total duration of the work request, calculated from the timestamp of the first activity until the timestamp of the last activity within the work request. Possible idle time is therefore also included in this calculation and because we only have one timestamp for each activity, some deviations should be taken into account. Most work requests take 40 to 50 days, as can be seen in Figure 7.5, with an average of 91 days and a standard deviation of 64 days. The longest work request containing the "Register aftercare" activity, implying a proper termination, took 288 days.

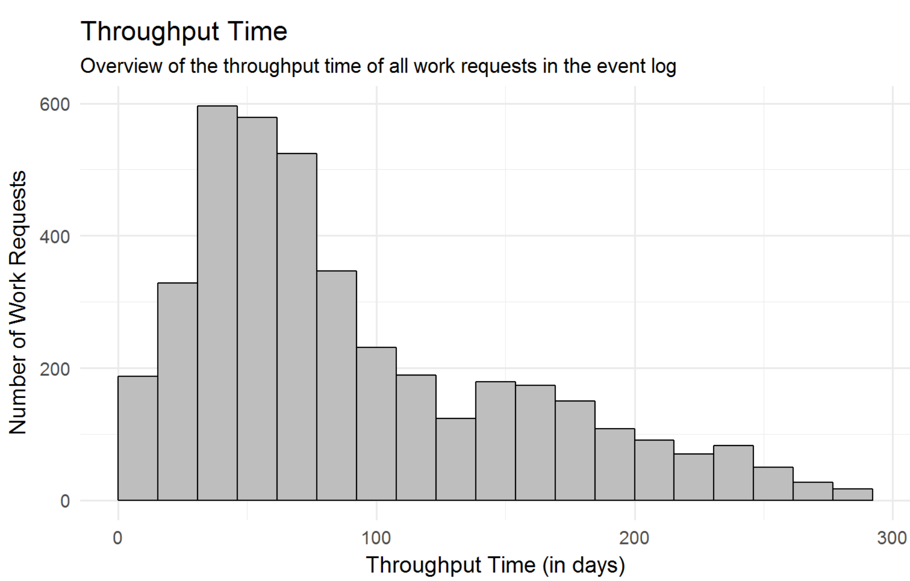

Figure 7.5: Overview of the throughput time of all work requests in the event log.

Pending cases. Besides these rather long *finished* work requests, which should be analysed more thoroughly by the company, the number of pending cases, which are in this case defined as work requests that are still open (not finished properly) at the date of data extraction, can be calculated. Data extraction took place at the 15th of January 2016, which is around nine months after the start of the work requests in our data selection. From the 5 176 work requests that are initiated in April 2015, 4 054 have been properly finished with the “Register aftercare” activity, implying that 1 122 work requests can be defined as pending cases. Table 7.3 contains a list of the most frequent process variants (traces) that are pending, accompanied with their frequency. A visual representation of the pending time of these work requests is given in Figure 7.6. This chart shows how long the pending work requests are already awaiting without any registered activity. In a large number of work requests there was no registered activity since their start in April 2015, which is more than 200 days. It can also be noticed that there is a large spread on the pending cases and that in most pending work requests the influence of the customer is very big. As can be seen in Table 7.3, the most frequent pending process variants end with an activity that requires customer input such as “Information customer”, “Analyse customer request”, “User agreement”, or “Offer sent”. This is presumably also the case for the long work requests that actually did finish correctly. Based on these analyses and findings, the company can conclude which types of work requests have the longest lead time and contain therefore a lot of waste. Consequently, they can define the value stream and improve the value flow, as was defined in the lean management philosophy.

Because there is only one timestamp present for each registered activity in the dataset, we are not able to calculate the *processing time* metric, or the actual time that activities are being executed. As an example, Figure 7.7 visualises the processing time of a fictional example on the resource-activity level. This example gives an overview of the time that each resource actually worked on each possible activity. By providing examples like this in a visual way, the company could be convinced of the benefit of collecting data more precisely in order to get a better insight in the activities and resources actually adding value to the process. Also the *waiting time* metric, which is an indicator of delays and thus waste according to the findings from literature [69, 118], can not be calculated with the data at hand.

Table 7.3: Process variants of the most frequent pending cases.

Pending cases process variant	**Frequency**
Creation Work request, Analyse customer request, Information customer	18.63 %
Creation Work request, Analyse customer request	12.75 %
Creation Work request, Analyse customer request, Request EAN, Offer sent, User agreement	5.53 %
Creation Work request, Analyse customer request, Offer sent, User agreement	4.46 %
Creation Work request, Analyse customer request, Technical visit, Offer sent, User agreement	4.01 %
Creation Work request, Analyse customer request, Technical visit, Offer sent	3.30 %
Creation Work request, Analyse customer request, Technical visit	2.94 %
Creation Work request, Analyse customer request, Information customer, Analyse customer request	1.69 %
Creation Work request, Analyse customer request, Study, Information customer	1.25 %
Creation Work request, Analyse customer request, Technical visit, Information customer	1.16 %

7.3.2 Structuredness metrics

- *Number of traces.* The first metric that was presented in order to get a notion of the variance in an event log is the number of traces or process variants. As was already shown in the general descriptions in Figure 7.3, the dataset at hand contains 4 054 (finished) work requests divided over 1 113 possible sequences of activities. This is a high number of process variants, i.e., a specific sequence of activities appears on average in 3.64 work requests in the event log at hand, implying that the process is very unstructured. The top 5 most common process variants is provided in Table 7.4. These process variants are rather similar, except for the fifth one, which does not contain "User agreement" and "Offer sent" which are two important activities in this process. However, according to the business people, this sequence of activities contains a special kind of work requests, implying it is not problematic that these two activities are missing. Nonetheless it can be stated that there is a lot of variance in the process, which can be further investigated with the other structuredness metrics, to gain insights in the underlying source of this variance.

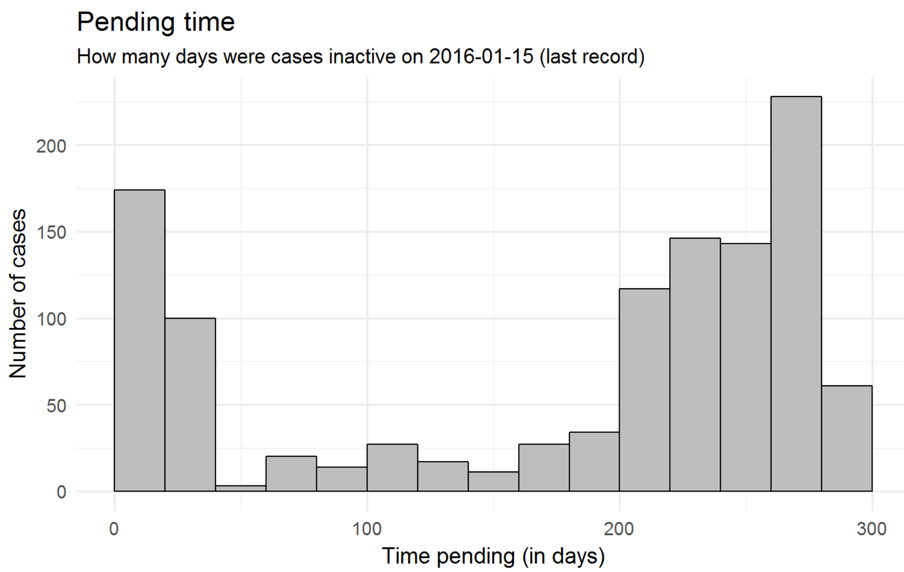

Figure 7.6: Pending time of work requests that did not finish before 2016-01-15.

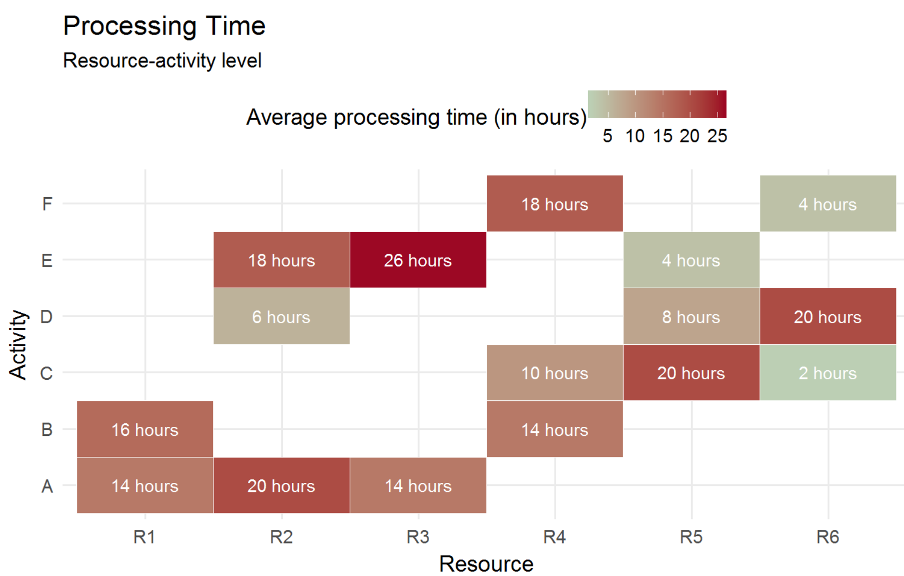

Figure 7.7: Processing time (resource-activity level) - fictional example.

- *Activity presence.* The activity presence metric shows the appearance of the

Table 7.4: Top 5 most common process variants in the event log.

Process variant	**Absolute frequency**	**Relative frequency**
Creation Work request, Analyse customer request, Request EAN, Offer sent, User agreement, Optimise file, Customer ready, Execution, Register aftercare, Physical document	601	14.82 %
Creation Work request, Analyse customer request, Offer sent, User agreement, Optimise file, Customer ready, Execution, Register aftercare	328	8.09 %
Creation Work request, Analyse customer request, Request EAN, Offer sent, User agreement, Optimise file, Customer ready, Execution, Register aftercare	289	7.13 %
Creation Work request, Analyse customer request, Technical visit, Offer sent, User agreement, Optimise file, Customer ready, Execution, Register aftercare	177	4.37 %
Creation Work request, Analyse customer request, Optimise file, Customer ready, Execution, Register aftercare	111	2.74 %

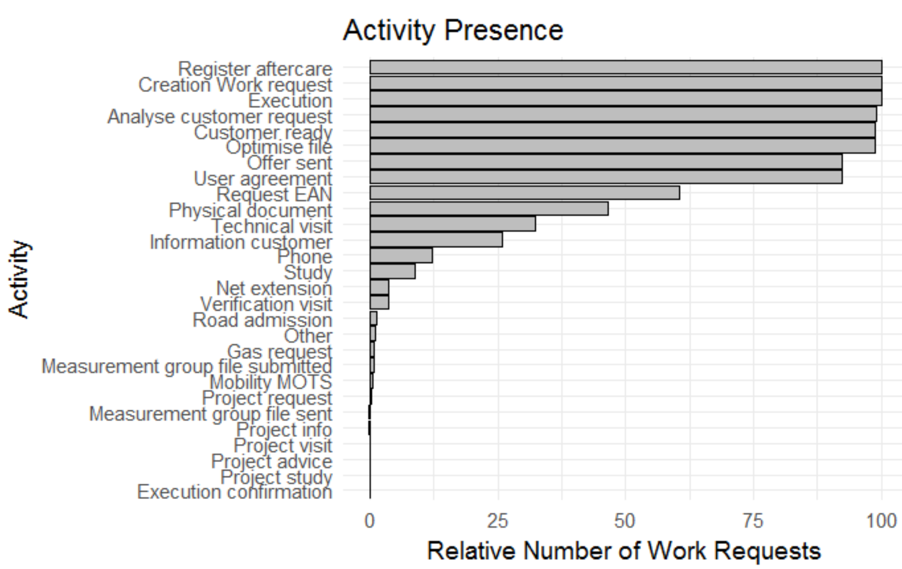

Figure 7.8: Overview of the activity presence in the event log.

different activities in cases, answering the question: "Are there work requests in which certain activities are skipped?". In Figure 7.8 we can see that there are work requests without the activities "User agreement" (7.82 %), "Offer sent" (7.75 %), "Optimise file" (1.38 %), "Customer ready" (1.13 %), "Analyse customer request" (0.91 %), and "Execution" (0.05 %), which are all activities that definitely should be executed before the aftercare takes place. As we filtered our dataset on work requests only including the activity "Register aftercare", these findings can be defined as certainly remarkable for the company. Especially work requests containing the activity "Net extension" should have a "User agreement" according to the business people, but this is always the case in the selection of data that was used. Work requests without the "Offer sent" activity appear to be less hazardous.

- *Start activities.* The first activity in each work request in the dataset at hand is "Creation Work request", which was added manually during data preparation to indicate the start of each work request. Figure 7.9 therefore shows one block indicating the same start activity for each work request. This metrics is therefore not adding any new insights for the case study company.

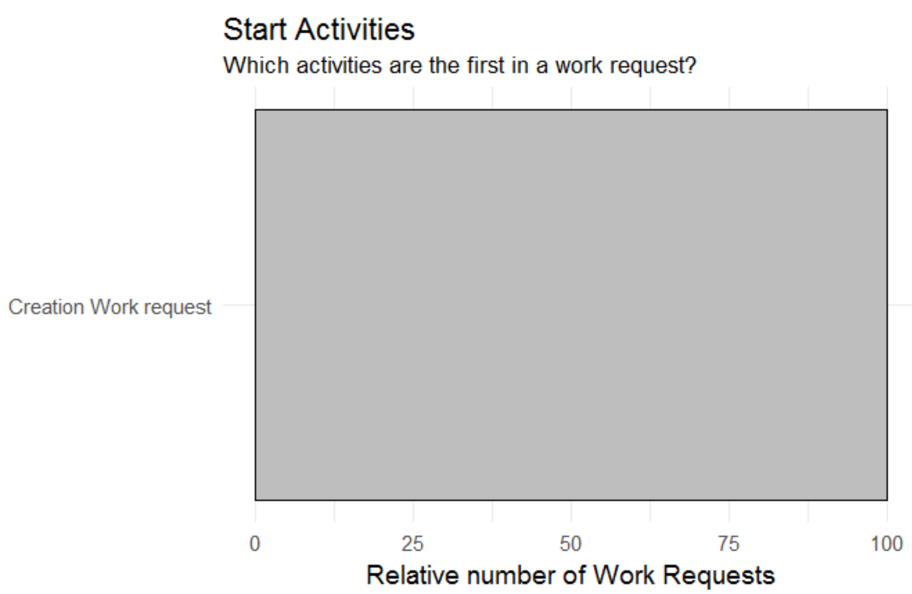

Figure 7.9: Overview of the start activities in the event log.

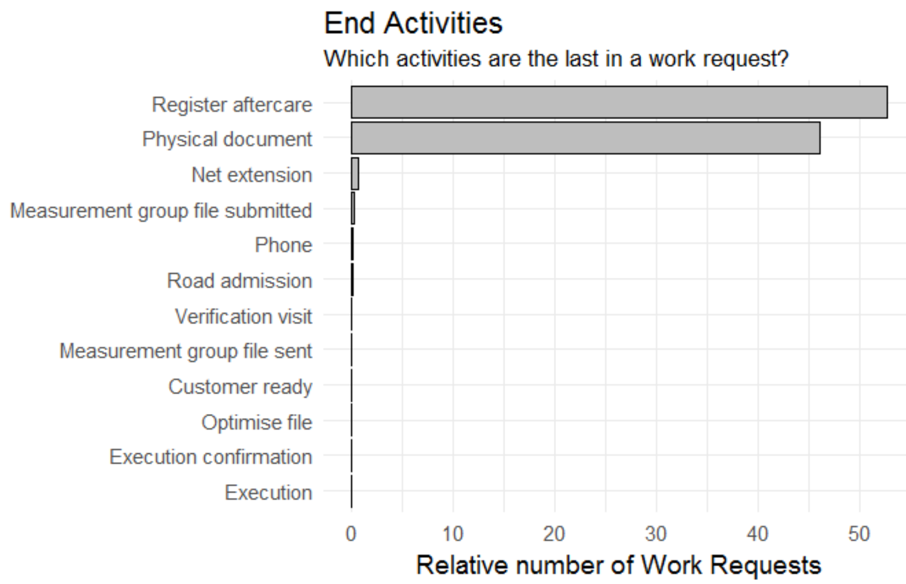

Figure 7.10: Overview of the end activities in the event log.

- *End activities.* Similar to the previous metric, the last activity in each work request can also be of interest to companies. The dataset at hand only contains work requests including the "Register aftercare" activity, implying that the end of most work requests is not surprising. The chart in Figure 7.10 shows which is the last registered activity for this selection. In 52.61 % of the work requests this is "Register aftercare" and in 46 % of the work requests this is "Physical document", which is another perfectly normal way to end the process. Consequently, 55 of the work requests (1.36 %), end with an activity other than "Register aftercare" or "Physical document", such as "Net extension" or "Phone", which are indications for the company to analyse these work requests more precisely, as they cause unwanted variance in the process. Because of the exact mapping of resources to activities in the dataset, this metric does not have any added value on the resource level or on the resource-activity level.

- *Case Length.* This metric shows the number of activities that have been executed in each distinct work request. It does not concern the number of distinct activities but the number of actual executions of activities. On average, ten actions or activity executions take place in a work request, with a standard deviation of 2. This is a rather normal length of a work request and does not alarm the business people. The results of this metric at the level of the entire event log are shown in Figure 7.11. The work request that has been dealt with the most contains 24 activity executions, which is more exceptional and requires the company's attention as it is probably containing a lot of excess or unnecessary processing, or waste.

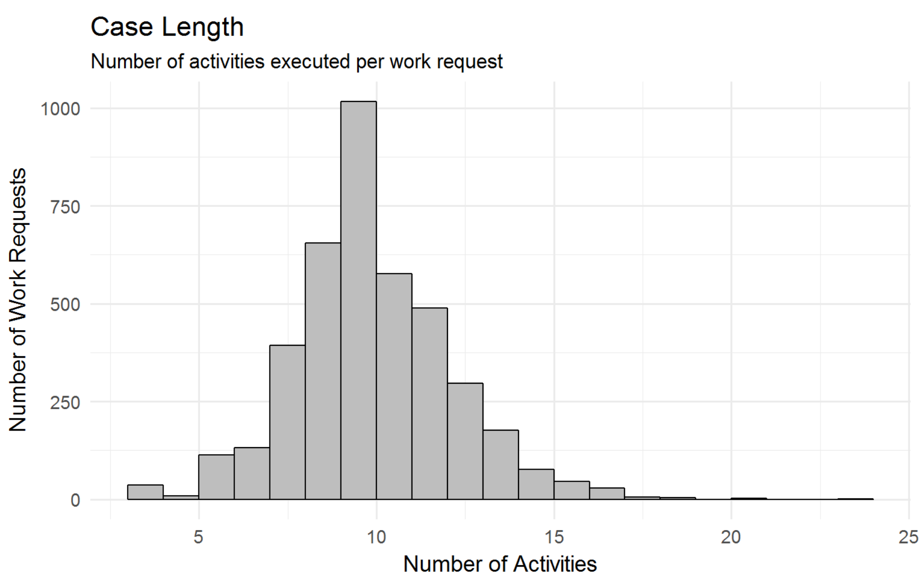

Figure 7.11: Overview of the case length in the event log.

- *Trace coverage.* Another structuredness metric measuring the variance in an event log is the trace coverage. This metric presents the minimum number of traces, or process variants, that is required to cover a certain percentage of the cases. For the dataset at hand, this metric provides the process variant coverage, representing the ratio between the number of process variants required to cover a part of, or the complete population. We can find here that 302 process variants or sequences of activities are required to cover 80 % of the complete population (event log), implying a high value of variance. Figure 7.12 shows the process variant coverage with the relative number of process variants, indicating that a high amount of process variants takes up most of the event log. To encompass the entire event log, 1 113 process variants were required, as was already shown in the general descriptions. Each process variant can be looked at individually on the trace level, together with its absolute and relative frequency. The most common sequence of activities appears 601 times in the event log, which is only 14.82 % of the entire event log. Table 7.4 provided the top five most common process variants in the event log, which do not differ a lot from each other.

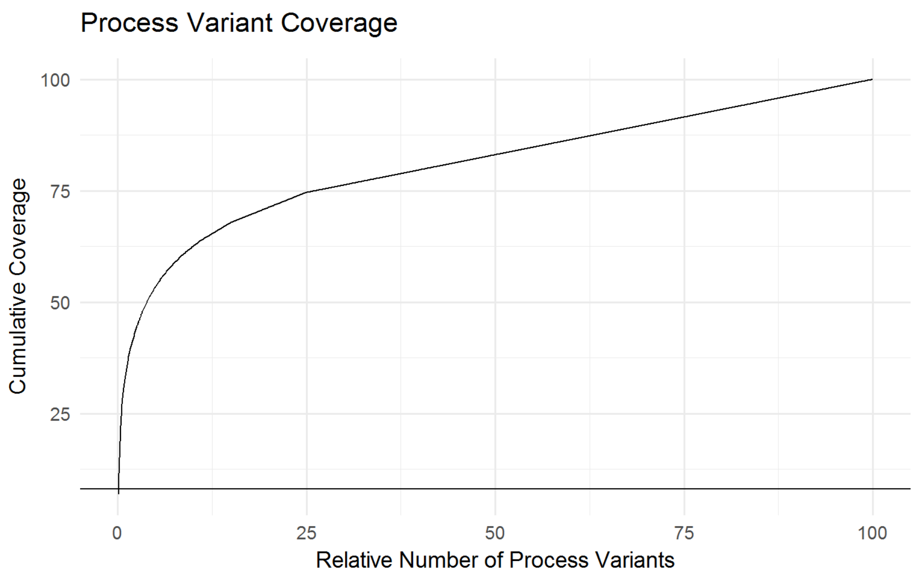

Figure 7.12: The trace coverage showing the number of process variants to cover 80 % of the entire event log.

- *Activity frequency.* Looking at the distribution of the distinct activities over the entire event log, the activity frequency metric can provide a company insights into the frequency of specific activities throughout the process. On the level of the entire event log, this metric shows that each distinct activity appears on average 9.8 times per case with a standard deviation of 1.67. On the level of the distinct cases, we can have a look at the activity frequency within each case. The highest number of distinct activities within one case is 17, which occurs in one case. The relative frequency for this case is 1, implying that each activity occurs only once in the case. More interestingly, at the activity level we can find which activities occur the most often in the entire event log. Table 7.5 provides an overview of the 10 most frequently occurring activities in the event log.

- *Self-loops.* For the rework metrics concerning self-loops and repetitions, a dis-

Table 7.5: Overview of the activity frequency for each activity in the event log.

activity	absolute activity frequency	relative activity frequency
Analyse customer request	5356	12.72 %
Customer ready	4353	10.34 %
Execution	4302	10.22 %
Register aftercare	4057	9.64 %
Creation Work request	4054	9.63 %
Optimise file	4019	9.55 %
Offert sent	3769	8.95 %
User agreement	3741	8.89 %
Request EAN	2454	5.83 %
Fysical document	1906	4.53 %

tinction was made between two types of rework in Chapter 5, i.e., redo and repeat rework. Redo stands for rework that is done for a second time by another resource. Repeat means that the same resource executes the activity again. Because the dataset at hand contains only roles of resources executing the activities, indicating that each specific activity is always executed by the same resource, redo metrics can be ignored here.

On the level of the event log, the number of self-loops metric shows how many self-loops occur in each work request. This metric shows that 8.14 % of the total number of work requests contain rework in the form of self-loops, and there are 347 self-loops in the total event log. Note that when for example two self-loops occur in a work request, this can be two times a self-loop of the same activity or two self-loops of two different activities. Figure 7.13 shows the number of self-loops compared to the relative number of work requests. Here we can see that 7.75 % of the total number of work requests contains only one self-loop, which is the majority of the total number of work requests containing one or more self-loops.

On the level of individual activities, the number of self-loops metric shows the number of times an activity occurs in a self-loop compared to the number of

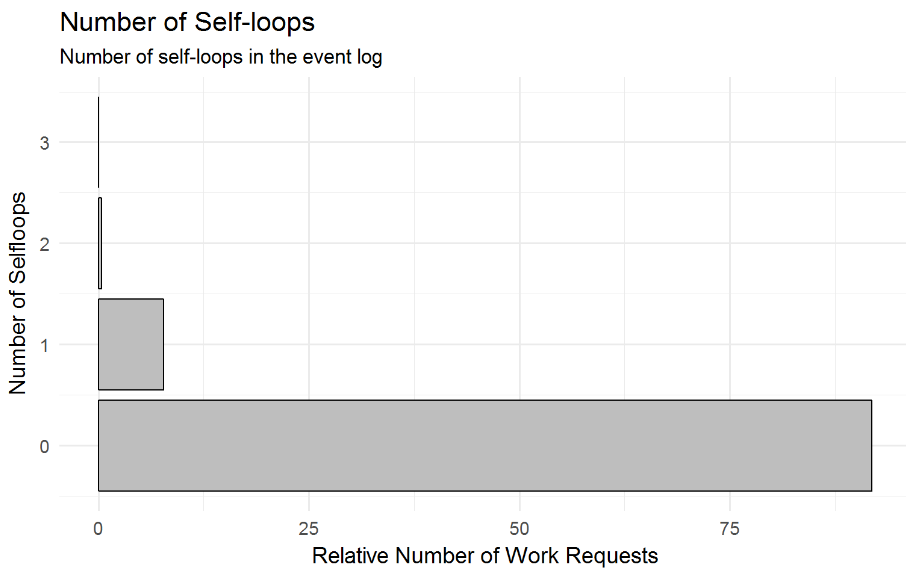

Figure 7.13: Overview of the number of self-loops in the entire event log, relative to the number of work requests within the event log.

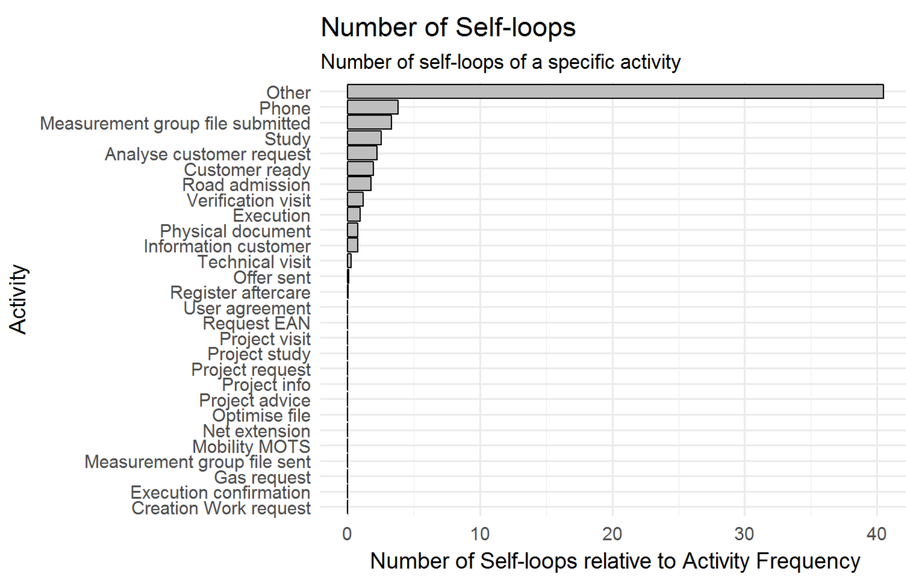

Figure 7.14: Overview of the number of self-loops in the event log on the activity level, relative to the total number of executions of each activity within the event log.

times this activity occurs in the entire event log. Figure 7.14 shows for example that in 2.22 % of the occurrences of the “Analyse customer request” activity, the activity occurs more than once immediately after each other, which is again an indication of waste and variability in the process. The activity “Phone” is higher in the chart, as it has a higher relative number of self-loops, although it occurs less often than the “Analyse customer request” activity. The category of activities with the name “Other” are activities that have been logged without a name, which can therefore be any other activity, explaining the high number of self-loops compared to the number of occurrences of this activity.

On the level of the resources, this metric provides an insight in the number of times a resource executes an activity multiple times immediately after each other for the same work request, compared to the number of times this resource executes an activity in the entire event log. However, because the dataset at hand only contains information about the roles for each activity, we cannot deduce information about activities that are executed by people or teams which were not planned or foreseen. However, this can be an interesting additional insight when the actual resources are added to the dataset. Figure 7.15 shows that “Team Net Development” occurs the most often in a self-loop, relatively, which is 1.84 % or 10 occurrences. “Work preparation” occurs 258 times in a self-loop, which accounts for 0.88 % of their total executions.

Next to the number of self-loops that occur in the process, it can also be of interest to have a look at the size of self-loops, which represents the number of times an activity is redone immediately again within the same work request. A self-loop which is executed three times after each other, has a size of two (two extra executions). Figure 7.16 shows that most work requests contain no self-loops and there are only three self-loops of size four, and one of size three. On the level of a specific activity, this can for example be interesting to see how often an activity that is in a self-loop is mostly repeated.

- *Repetitions.* Similar to the self-loops, the number and size of repetitions, i.e., activities that are repeated in the work request with at least one other activity in between, can also be interesting to have a look at. Figure 7.17 shows the number of repetitions compared to the relative number of work requests. In 36 % of the work requests an activity is repeated (not immediately after the previous execution), with a total of 1 766 repetitions in the entire event log. However, most work requests that contain a repetition, only contain one. Note that if two repetitions occur in a work request, this can concern two different

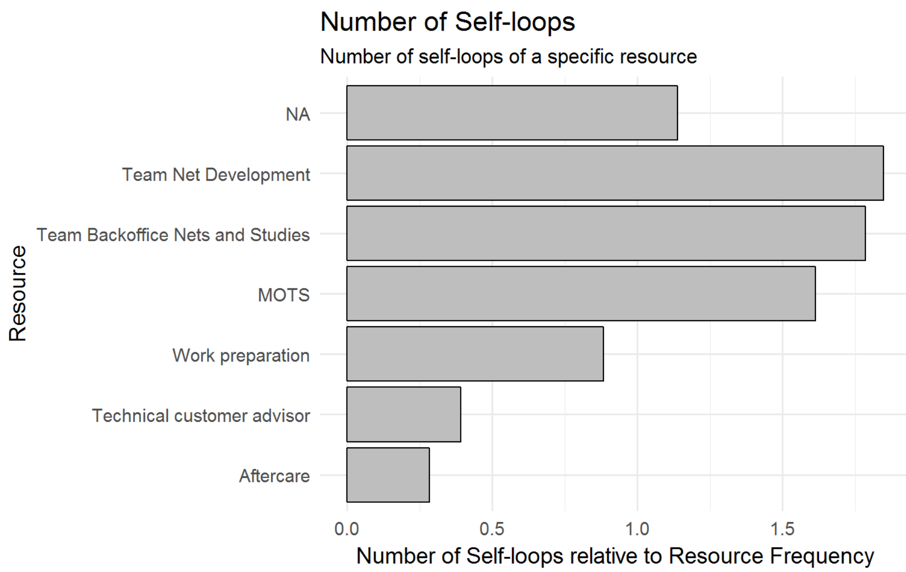

Figure 7.15: Overview of the number of self-loops in the event log on the resource level, relative to the total number of executions of each resource within the event log.

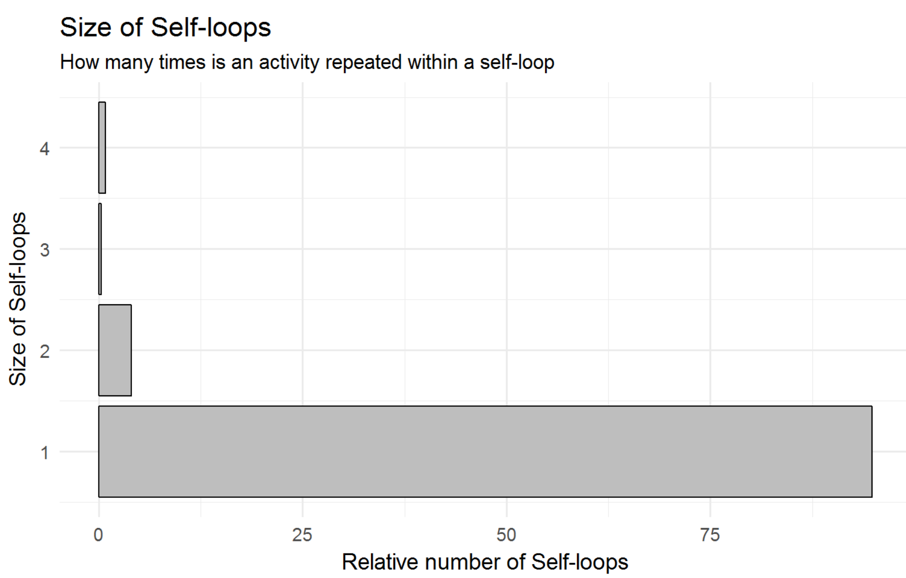

Figure 7.16: Overview of the size of self-loops in the entire event log.

activities or two times the same activity.

Figure 7.17: Overview of the number of repetitions in the entire event log.

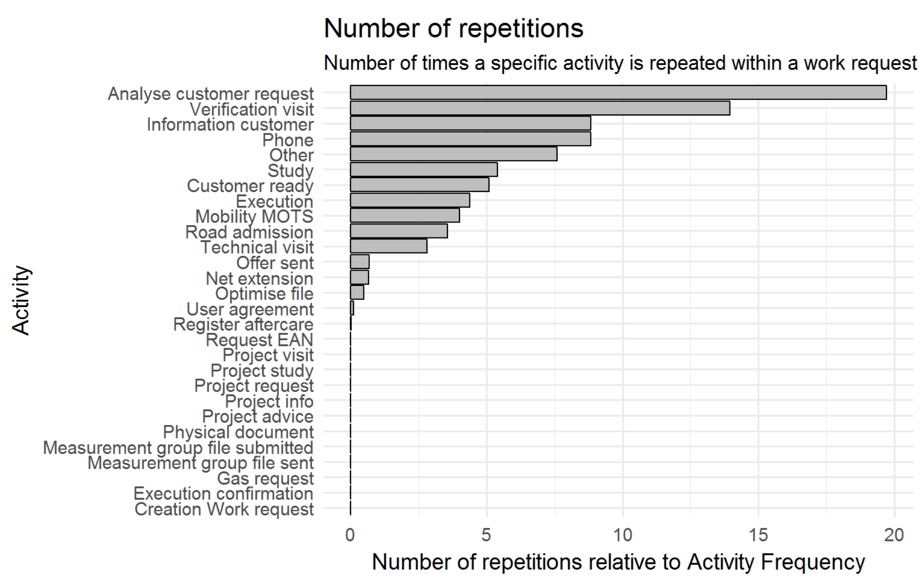

Figure 7.18: Overview of the number of repetitions in the event log on the activity level, relative to the total number of executions of each activity within the event log.

On the level of the individual activities, the number of repetitions metric shows how often an activity occurs in a repetition compared to the number of times this activity is executed in the entire event log. We can observe in Figure 7.18 that 13.95 % of the occurrences of "Verification visit" (24 occurrences in total) occurs more than once in one work request with at least one other activity in between. Other activities that are repeated later on in a work request are for example "Analyse customer request" (19.72 %), "Information customer" and "Phone" (both 8.84 %), "Customer ready" (5.10 %), and "Execution" (4.29 %). From the discussion with the business people we learned that some of these activities can actually be defined as waste. However, the "Verification visit" and "Analyse customer request" are absolutely required and cannot be ignored. These activities take up a lot of time, so executing them more than once within the same work request should be avoided in all cases.

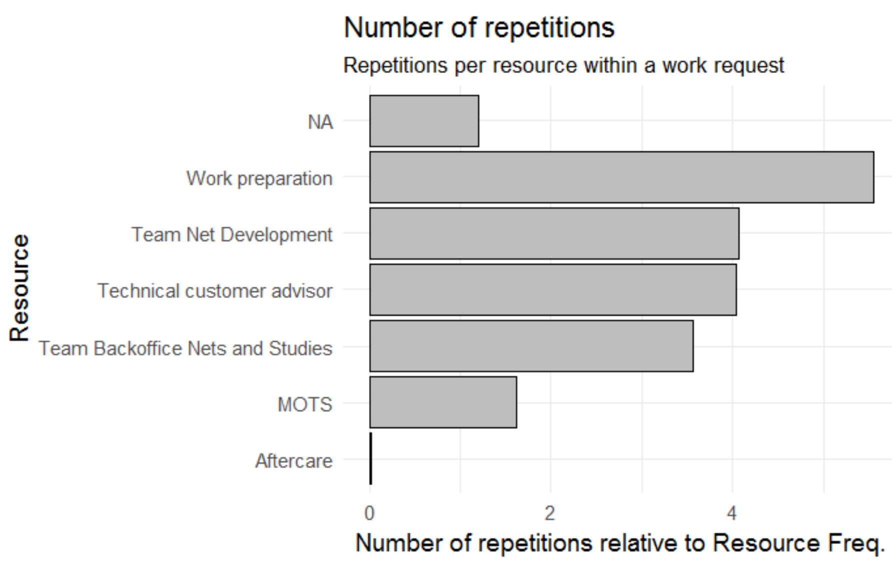

Figure 7.19: Overview of the number of repetitions in the event log on the resource level, relative to the total number of executions of each resource within the event log.

Next, on the level of the resources, this metric shows the number of times each resource executes an activity multiple times per work request with at least one other activity in between, compared to the number of times this resource executes an activity in the entire event log. As can be seen in Figure 7.19, the "Work preparation" team, for example, executes 1 621 times a repetition in a work request, which counts for 5.55 % of all their executions. However, because the resources are mapped one-on-one with their activities, little new insights will be found.

Finally, the size of the repetitions shows the number of times an activity is repeated in a work request. An activity that is executed three times in a work request, with other activities in between, has a size of two. Figure 7.20 shows that there are two work requests in which an activity is repeated up to five times (size 6). However, the vast majority of repetitions only contain one repetition of a certain activity.

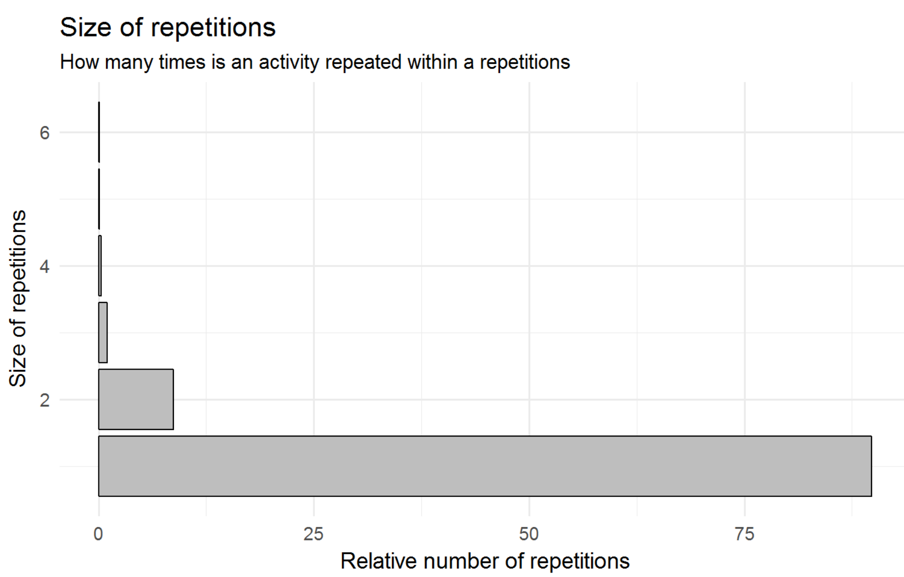

Figure 7.20: Overview of the size of repetitions in the event log.

7.3.2.1 Resource metrics

- *Resource frequency - workload.* This metrics answers the question "How many tasks does a resource execute?". The answer to this question provides us the number of actions per resource in the entire event log. Because the resources in the dataset at hand are roles that are mapped one-on-one with the activities, there are no remarkable findings, as can be seen in Figure 7.21. The people from "Work preparation" are responsible for 69 % of all actions in the event log. "Aftercare" is only responsible for 14 %. "NA" is linked to activities that have not been identified with a specific role in the dataset. In case the actual resources would be added to the dataset, also here process variation, waste, and non-value added work can be discovered.

On the level of each resource-activity combination, we can find how essential a resource is for the process or for a specific activity. The division of the resources over the different activities can be seen in Figure 7.22. However, given the data structure at hand, analysing this division is pointless because there is no overlap.

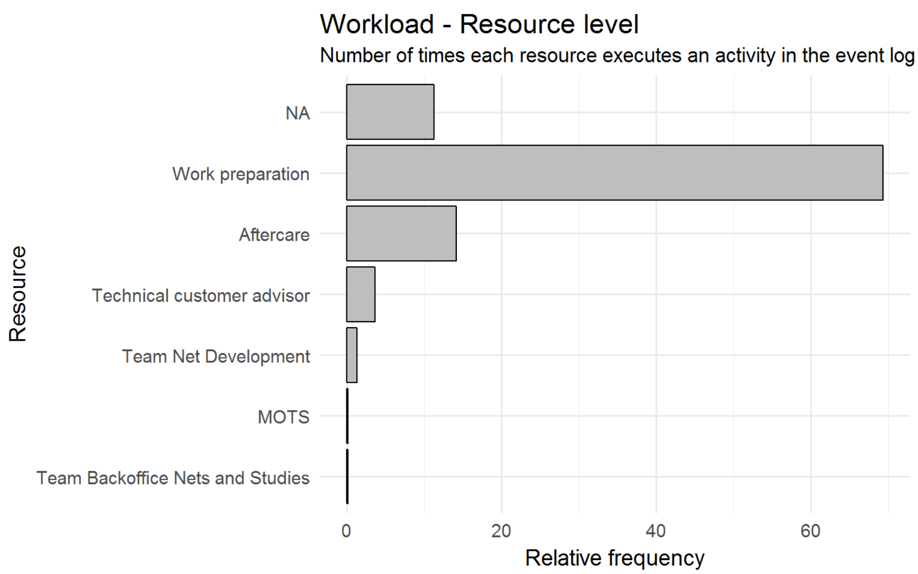

Figure 7.21: Overview of the resource frequency per resource in the entire event log.

- *Resource involvement.* This metric is a simple overview of which resources are the most involved in the process. On the level of specific resources, we find in Figure 7.23 that "Work preparation" and "Aftercare" are required in each work request. "Technical customer advisor" is required in 1 396 work requests (34 %), "Team Net development" in 405 (9.99 %), "Team Backoffice Nets and Studies" in 53 (1 %), and "MOTS" in only 29 (0.72 %) of the work requests.

  Similarly, this metric on the resource-activity level answers the question: "In how many work requests is each specific resource-activity combination involved the most?". However, each activity is assigned to a specific resource in this dataset, which can be seen in Figure 7.24. For example, in all work requests the "Execution" is performed by people from the "Work preparation" team. It

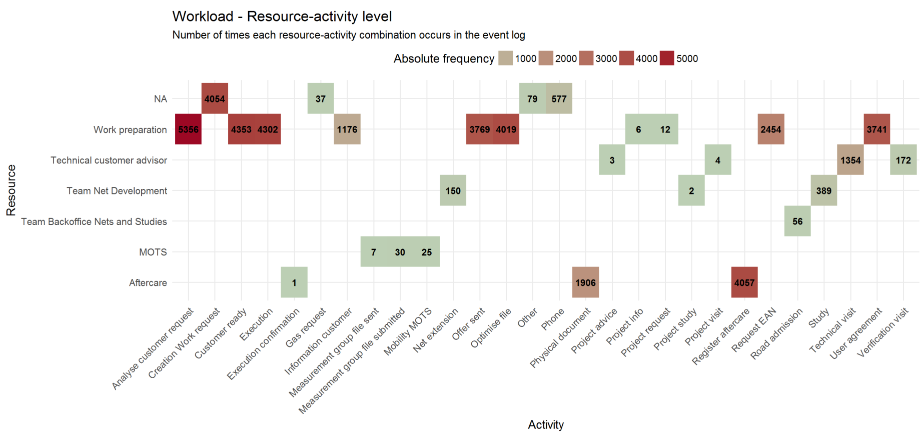

Figure 7.22: Overview of the resource frequency per resource-activity combination in the entire event log.

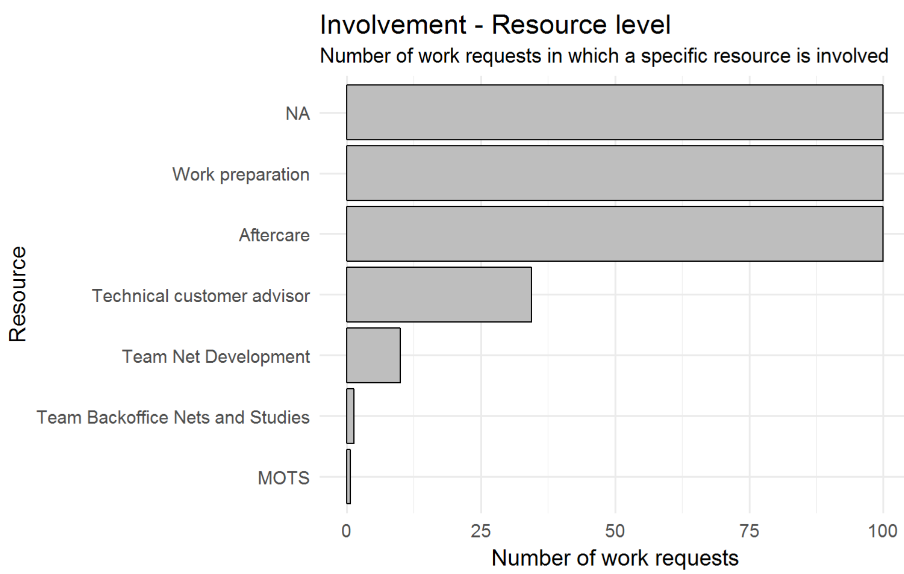

Figure 7.23: Overview of the resource involvement in the entire event log.

would be interesting to find resources that are involved in all cases, what could be very dangerous in case this person gets ill or leaves the company. On the other side, resources that are only involved in very few cases could be a cause of longer lead times or variation because this resource is not familiar with the activities in this process.

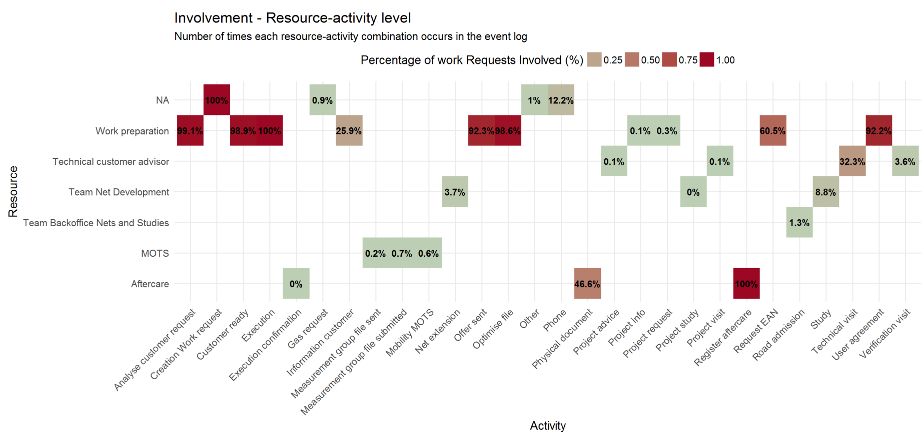

Figure 7.24: Overview of the resource involvement per resource-activity combination in the entire event log.

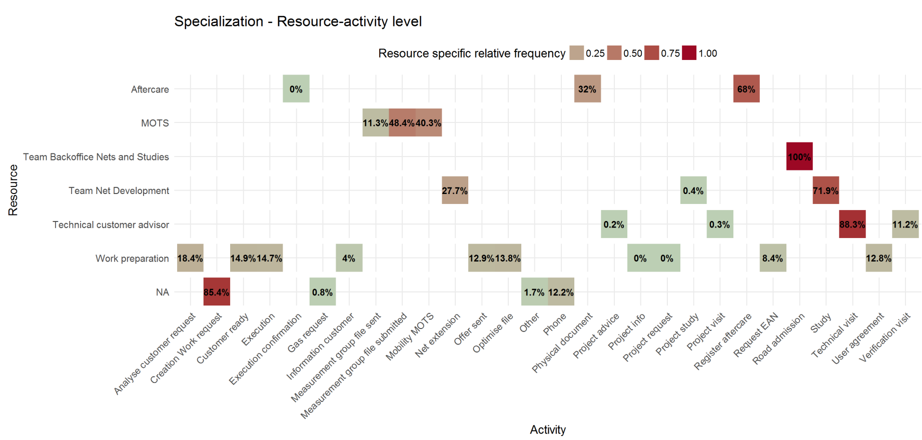

Figure 7.25: Overview of the specialisation of resources on the resource-activity level in the entire event log.

- *Resource specialisation.* Similar to the resource involvement metric, the resource specialisation metric shows which resources execute more different activities than others, which can be interesting for business concepts such as brain drain or team selection, and can again be an indication of longer lead times and variation in case resources are working on *all* activities in the process instead of specialising in one or some of them. Figure 7.25 shows that "Work preparation" executes different activities, while "Team Backoffice Nets and Studies" only performs the "Road admission". Of course, the interpretation of these findings are left to the domain experts in the company who can subsequently take actions in order to optimise the process.

7.4 Findings

Together with some useful filters that can be applied to the dataset in the dashboard that was created and the set of metrics presented in Chapter 5, the findings from the applied metrics were well received by the company.

Next to the application of the metrics to the event log, another request of the company concerning the metrics has been analysed. One of the regulations that the process under analysis needs to comply with is a criterion stating that the time between the agreement of the customer and the execution should not be any longer than 15 working days. To analyse if the company is meeting this criterion, and what can possibly be the cause in work requests that do not meet this criterion, the dataset was split up in two parts. All work requests containing both the activities "Customer agreement" and "Execution", which are 3 736 cases, are ordered based on the duration in time between these two activities. There are 2 239 work requests that did not meet the criterion, and 1 497 work requests that did. Figure 7.26 shows the throughput time of the work requests in both parts, implying that work requests take considerably more time when the criterion is not met, which is not surprising.

Next to this, the case length can also be of interest. Figure 7.27 shows that when the criterion is not met, slightly more activities are executed per work request. However, what can be of more interest to the company, is the sequence of the activities in the work requests that do not meet the criterion. Therefore, Table 7.6 shows the most frequent process variants for both parts of the dataset, together with their frequency in the dataset. It can be noticed that work requests that do not meet the criterion contain the activities "Request EAN" and "Physical document". After presenting these findings to the company, the domain experts confirmed that the work requests containing the activity "Request EAN" actually take much longer because of this activity, but it can not be classified as non-value-adding for the process so it is not waste that should be reduced.

Other findings based on the application of the metrics and the analyses above are:

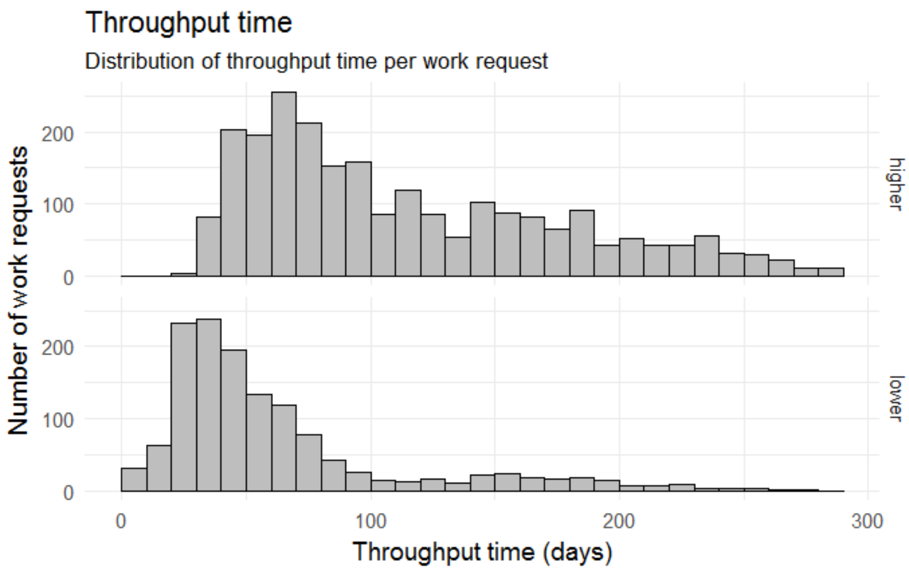

Figure 7.26: Overview of the throughput time (in days) for work requests meeting the criterion (lower) and work requests not meeting the criterion (higher).

- Most work requests have a duration of 40 to 50 days, with an average of 91 days and a standard deviation of 64 days, which is very high, implying a very high variation in time.
- Pending cases are mostly work requests that get hindered by customer dependent activities.
- As expected, every work request starts with the (artificially added) activity "Create work request".
- 55 work requests (1.36 %) end with an other activity than "Register aftercare" or "Physical document", which can be classified as variation.
- On average 10 activity executions take place in a work request, with a standard deviation of 2. However, the most *active* work request contains 24 executions.
- There are work requests without the activities "User agreement" (7.82 %), "Offer sent" (7.75 %), "Optimise file" (1.38 %), "Customer ready" (1.13 %),

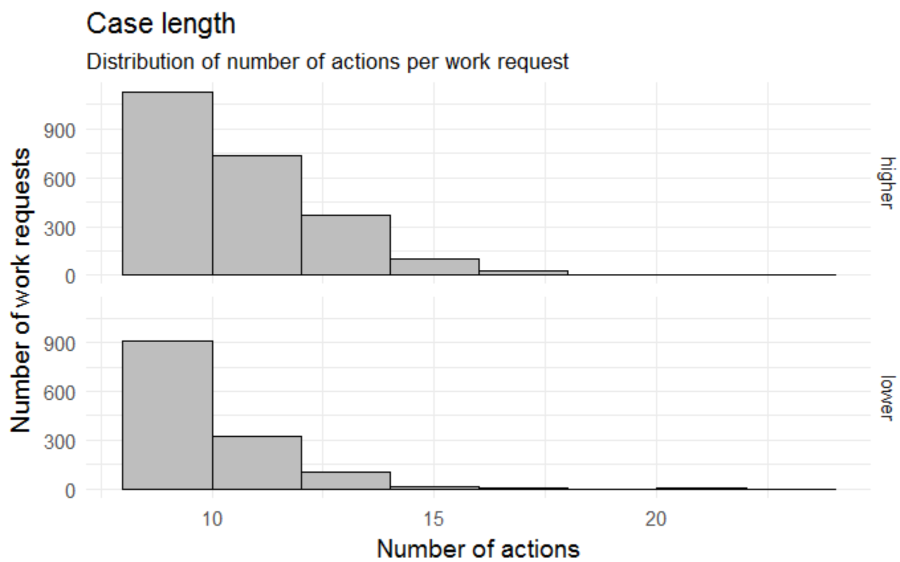

Figure 7.27: Overview of the case length for work requests meeting the criterion (lower) and work requests not meeting the criterion (higher).

"Analyse customer request" (0.91 %), and "Execution" (0.05 %), which are all activities that definitely should be executed before the *aftercare* takes place.

- There are 302 process variants or traces required to cover 80 % of the entire event log.
- The most frequent process variant occurs 601 times. This is only 14.82 % of the entire event log implying a high amount of variation in the process.
- 8.14 % of the work requests contain rework in the form of self-loops. There are 347 self-loops in the total event log.
- In 36 % of the work requests a repetition of an activity with at least one other activity in between occurs. There are 1 766 repetitions in the entire event log.
- Rework in the form of self-loops or repetitions both mostly occur for the activities "Analyse customer request", "Information customer", "Customer ready", and "Execution", which are all important and value-adding activities in the process at hand.

Table 7.6: Top 5 most common process variants for the criterion that has been met or not.

Criterion met	
Process variant	**Absolute frequency**
Create work request, Analyse customer request, Offer sent, User agreement, Customer ready, Optimise File, Execution, Register Aftercare	43
Create work request, Offer sent, Analyse customer request, User agreement, Customer ready, Optimise File, Execution, Register Aftercare	33
Create work request, Analyse customer request, Offer sent, User agreement, Optimise File, Customer ready, Execution, Register Aftercare	25
Create work request, Offer sent, Analyse customer request, User agreement, Optimise File, Customer ready, Execution, Register Aftercare	25
Create work request, Analyse customer request, Technical visit, Offer sent, User agreement, Optimise File, Customer ready, Execution, Register Aftercare	24

Criterion not met	
Process variant	**Absolute frequency**
Create work request, Analyse customer request, Offer sent, **Request EAN**, User agreement, Optimise File, Customer ready, Execution, Register Aftercare, **Physical document**	35
Create work request, Analyse customer request, **Request EAN**, Offer sent, User agreement, Optimise File, Customer ready, Execution, Register Aftercare, **Physical document**	35
Create work request, **Request EAN**, Analyse customer request, Offer sent, User agreement, Optimise File, Customer ready, Execution, Register Aftercare, **Physical document**	35
Create work request, Analyse customer request, **Request EAN**, Offer sent, User agreement, Optimise File, Customer ready, Execution, Register Aftercare	31
Create work request, Offer sent, Analyse customer request, **Request EAN**, User agreement, Optimise File, Customer ready, Execution, Register Aftercare, **Physical document**	31

- Because the dataset contains roles instead of actual resources, there are no new insights on the level of resources. For example, in all work requests the “Execution” activity is performed by the “Work preparation” team.

- The “Work preparation” team is responsible for 96 % of all executions in the event log.

- The “Work preparation” team is required in each work request, while “MOTS” is only required in 29 (0.72 %) work requests.

- “Work preparation” executes a lot of different activities, while “Team Backoffice Nets and Studies” only executes the “Road admission” activity. This finding, together with the previous two findings, can be indications of variation and waste within the process. In case the actual resources are added instead of the roles or the team executing the activities, this can be investigated more thoroughly.

Finally, Table 7.7 gives an overview of some questions posed by the case study company that were answered by applying the developed metrics. The metrics and questions are structured according to the most important operational excellence principles presented in Chapter 2 and Chapter 3.

7.5 Conclusion

To conclude, we can state that the presented metrics in Chapter 5 represent the behaviour that is present in the event log without the use of a process model, and thus without any influences from underlying algorithms or assumptions. Moreover, different groups can be easily and correctly compared and analysed, both over time or based on certain case attributes.

Besides that, this case shows that the applied metrics are supporting some important operational excellence principles as they are based on the reduction of waste and non-value-adding activities and the management of variation and structuredness, which were stated throughout this dissertation to be requirements to reach operational excellence [154]. First, it can be stated that the five lean management principles of waste, listed in Chapter 2, are supported by the presented metrics. For instance, metrics such as the throughput time and the ones concerning rework provide the company with an overview of which steps in the process and which features of products and services add value, and which can be identified as waste. Second, the value stream,

Table 7.7: Some questions posed by the case study company that are answered with the developed metrics.

Principles	**Metric and question**
Reduction of waste and removal of non-value-adding activities	Throughput time: how long do work requests take?
	Rework: How many repetitions or self-loops take place?
	Rework: How many times are activities repeated?
Reduction of process variation	Number of traces: How many process variants do we find in the event log? Which ones are the most frequent?
	Activity frequency: How often are certain activities executed?
	Trace length: how many actions are executed per work request?
	Start and end activities: Which activities are the start and end activities of work requests?
Reduction of resource variation	Workload: How many actions do resources perform in the complete process or in each work request?
	Resource involvement: In how many work requests are resources involved?
	Resource specialisation: How specialised are resources? Which activities do they perform?

or the sequence of activities actually adding value to the process, can be defined by the business people based on the findings from the metrics. Next to this, by removing the non-value-adding activities, such as repetitions, work in queue, or batch processing, interruptions in the process are reduced. Another lean management principle that has been covered by the presented metrics is 'pursue perfection and continuous improvement' as the application of the metrics is easily repeatable. The metrics also cover different concepts that were defined as waste according to Pepper and Spedding [118] and Hines and Rich [69], such as waiting, delays, or unnecessary motion. As the lean management tools are rather qualitative, the presented metrics are a useful quantitative and very objective addition to existing tools.

The metrics concerning the number of process variants, the rework metrics, and the activity presence are examples of metrics that support the Six Sigma philosophy, which focuses on the reduction of variation in business processes in order to minimise defects and errors [93]. But also the organisational metrics measuring the variation in resources focus on optimisation of the workload and resource specialisation by analysing where variation takes place. Looking at the DMAIC-cycle of Six Sigma, which is also presented in Chapter 2, the metrics can be placed in different steps of this cycle. We can, for example, identify improvement issues based on the findings from the application of the metrics, which is a task in the *define* step. In the *measure* step, an insight in the process is provided by the metrics, to get a notion of the as-is status. However, the *analysis* step is the most crucial stage because here the actual source of variation in the process should be sought and potential critical inputs should be identified. This is also covered by the presented metrics. However, the metrics provide an objective analysis of business processes and are not providing any interpretation for the business people. These interpretations, and the resulting improvement steps that should be taken to reduce variation and consequently improve the process, are steps the business people should take based on the provided analyses. Next to this, both lean management and Six Sigma place importance on repeatability and reproducibility of the systems to measure the operational excellence of processes [7].

For future use, some recommendations can be done to the case study company (and to other companies) to get even more benefit from the analyses. Firstly, the data logging should be optimised in order to make fully use of the developed metrics. The most important logging requirements are start and end timestamps for all activities in order to calculate bottleneck activities, actual processing time for each work request compared to inactivity, and waiting time. Next to this, actual resources should be tracked in order to calculate the specialisation of resources, their (in)efficiency, and the actual active time of resources compared to their inactivity. Secondly, the analysis of the case study above could be improved by introducing benchmarks against which the results can be evaluated. Thirdly, and related to the previous recommendation, finding a link between the business' key performance indicators (KPIs) and the presented metrics could be of interest, to analyse which metrics should be calculated to cover which KPIs. These can be interesting topics for future research.

Chapter 8

Conclusions and future recommendations

This dissertation focuses on the use of event log knowledge to support operational excellence in businesses. Given the limited existing work on the use of event log knowledge in the light of operational excellence, the first part of this dissertation contains a literature overview of existing techniques in the field of operational excellence and the interplay of operational excellence and process mining. Chapter 2 provides an overview of the evolution of quality management and the emergence of different methodologies in the field of operational excellence, while Chapter 3 supplements these findings with an introduction to the field of process mining and an outline of the match between process mining and operational excellence. From this foundation, the second part of the dissertation focuses on the development of methods that retrieve event log insights to support specific operational excellence concepts. In order to address the lack between process mining and operational excellence, this dissertation will introduce the concept of event log-based process metrics for which the requirements are identified in Chapter 4. The developed metrics are then presented in Chapter 5, which are developed on different levels of analysis. One of the operational excellence concepts that was stated to be an indication of waste in a business process is batch processing, which is further elaborated upon in Chapter 6. Next to this, the presented metrics are applied to a real-life case study in Chapter 7. And finally, this chapter summarises the main conclusions and provides recommendations for future research, as shown in Figure 8.1.

Throughout this dissertation, the principles and steps of the design science re-

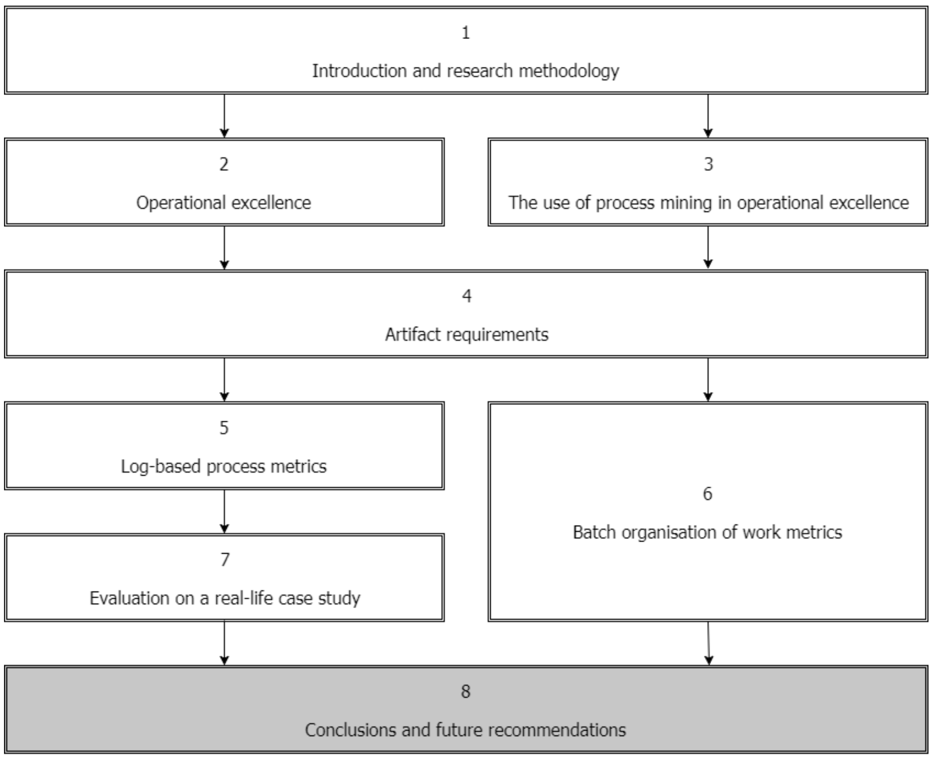

Figure 8.1: Outline of the thesis - Chapter 8.

search framework have been followed and applied. Therefore, the following section is structured according to these steps, which were discussed in Chapter 1.

8.1 Final conclusions

New technologies, fast changing customer expectations, and growing competition drive companies to continuously modify and improve their business processes. These processes are therefore constantly changing and different methodologies and philosophies have been developed to implement these business changes. Nowadays, companies require high-end information systems that execute the company processes in an efficient and effective way, processes which include decent control and planning processes, good operating systems, and a culture of continuous improvement to become operationally excellent. The concept of continuous improvement is related to operational excellence techniques such as lean management, Six Sigma, business process improvement (BPI), and business re-engineering, which can be situated in the total quality management (TQM) field. A combination of these elements is often recommended, as the individual programmes can be found to be insufficient for companies.

During the first step in a design science research project, which is the problem identification step, the researcher has to gain a clear and precise understanding of the problem at hand. The problem domain in this dissertation is therefore explored by conducting a literature review of operational excellence which aims to highlight its strengths and weaknesses. Based on a comparison analysis of the different philosophies in operational excellence, it can be stated that existing methods in this field are mainly based on qualitative paper-and-pencil approaches. It may be argued that these techniques are ignoring the actual underlying system and not collecting enough detail. Moreover, the principles of lean management are also found to be less useful in the volatile or fast changing environments in which companies, especially service companies, are operating nowadays. Six Sigma tools on the other hand, are in some cases based on quantitative data, while others are also based on qualitative methods such as interviews. The results of these methods and the decisions taken based on these results may therefore be rather subjective and depending on the person or team that has been performing the analyses. This leads to the fact that there is not one single strategy or roadmap for companies to improve the performance of their processes. A combination of multiple methodologies should be incorporated, and there is a need for more objective and data-based analyses to gain more insights in the operational excellence of companies. Moreover, literature on the application of these operational excellence techniques within service-oriented companies is limited, leading to these companies not yet being convinced of the use of operational excellence techniques within their companies.

In the field of business process improvement, claims are found that specific guidelines for business process optimisation are limited in literature and that it is not always clear how existing techniques are used to support the process of business process improvement. Process mining was therefore introduced, as it shows to be promising in the field of knowledge retrieval based on process data that is collected from process aware information systems in companies. However, it can be stated that most of the research on process mining is focused on discovering process models from event logs, and checking the conformance between the discovered model and the underlying event log. These models are learned from event logs with certain algorithms, based on parameters and assumptions, and are often manually manipulated with sliders and filters in visualisation tools. This may result in conclusions and improvement measures that are less reliable as they are based on possibly incorrect or incomplete process models that do not contain all information of the business process under analysis or that contain unobserved behaviour.

Based on these findings, a gap between operational excellence and process mining was identified and the need for an artifact that uses the knowledge from event logs to support business process improvement in the light of operational excellence was identified. In the requirement specification step of a design science research, an artifact that addresses the defined problem should be identified and described. The requirements for the artifact in this dissertation are assembled based on the literature review that was conducted on the principles of operational excellence and on the existing interplay of process mining and operational excellence on the one hand, and on the results found in the interviews that were conducted with business process experts on the other hand.

The requirements that have been created in this dissertation, based on the problem identification, are fourfold and are both functional and non-functional. Firstly, event log-based metrics should be developed, which provide an organisation insights into event log knowledge without the use of intermediate models. The first requirement that was developed states that the event log-based metrics should measure both the general aspects of an event log and the more specific measures concerning the operational excellence field, which were identified to be useful event log knowledge insights in operational excellence. Secondly, the metrics should measure only one dimension or level of analysis, in order to remain comprehensible. Therefore, each of the measures should be defined on different levels of analysis in order to get insights into the different degrees of granularity of a business process. Thirdly, all metrics should contain clear descriptions of the measure itself, the requirements for the event data, and the underlying calculation, which are found to be missing regularly in existing research. And finally, the metrics should be complemented with a suitable visual representation and the used terminology should be adapted to the correct level of interpretation in order to be understandable for business people.

Based on these requirements, event log-based process metrics are presented as an artifact, which give a company insights in the behaviour in an event log. The presented metrics are structured along two categories of process performance measures that should be focused on in a business process improvement project, which are time and structuredness. These dimensions are both shown to be direct indicators of different types of waste and other indicators of operational excellence in literature. For the other two categories of performance measurement of the Devil's Quadrangle, which are cost and quality, no metrics have been presented in this research. Structuredness is defined as the level of variation within the event log and concerns the level of variance in a process, the amount of rework, and the organisation of resources, which are the people or machines executing activities in the process. The presented metrics are calculated on one of the following levels of analysis: log, case, trace, activity, resource, or resource-activity, and also involve one or more of the aspects of operational excellence such as, among others, batch processing, waste reduction, rework, iterations, and the removal of non-value-adding activities. Finally, the metrics that are created in order to overcome the identified lack serve as a mean to objectively compare different event logs in terms of the different aspects of process performance.

These metrics provide business people with a picture of the present process behaviour and are all implemented in the R-package edeaR, making them easy to apply to any event log. Moreover, to make the results of the metrics more valuable and accessible for business people, a dashboard including visualisations of each metric has been developed and presented. Next to the visualisations, the dashboard contains filters to get more insights into the quality and different aspects of the business process. Comparisons with benchmarking values and analyses of the event data over time are other possible advantages of the presented metrics and filters within the metric dashboard.

The metrics concerning the time dimension measure the throughput time, the actual processing time, and the waiting time on different levels of analysis to get an insight in, for example, the duration of cases and activities. Based on these metrics, supplementary operational excellence concepts such as the number of pending cases and bottleneck indicators within a business process can be calculated.

Variability is defined as one of the key sources of waste within a business process, as it is causing a process to deviate from its desired behaviour. However, as variability or unstructured behaviour should not always be eliminated, it is important for a company to learn the type of unstructured behaviour that is occurring within the process under analysis. In order to get more insights in this, different classes of metrics concerning the structuredness of a process are developed, which are variance metrics, rework metrics, and organisational or resource-related metrics. The variance metrics include measures calculating the number and length of traces, the presence of start and end activities within the process, and the trace coverage of a business process. Rework metrics look at the number of times activities are repeated within a case, both immediately following each other, which are defined as self-loops, and with other activity executions in between, which are defined as repetitions. Here, a distinction is made between repeat and redo self-loops and repetitions, indicating the difference between activities that are repeated by the same resource, and activities that are redone by another resource. These rework metrics give companies an insight in which activities are often executed more than once, which resources are involved in the rework and how many times the activities are repeated when they occur in a self-loop or repetition. These are all possible indicators of waste within the process. The resource metrics that have been presented within this dissertation concern the different aspects of the process behaviour concerning the resources, which are defined as process participants, software systems, or equipment, which execute activities in the process. These metrics are related to the frequency of a resource executing activities, the involvement of a resource in the process, and the specialisation of the resources. Again, these metrics can be calculated on different levels of analysis and can support organisations in performing knowledge management, for instance when creating a knowledge map, or project management with applications such as resource levelling and resource allocation.

Getting insights in the behaviour of resources and the amount of "waste" they cause can be very interesting for companies who want to optimise their business processes. Related to this, are the metrics concerning the concept of batch processing, in which resources execute activities of two or more cases at the same or almost at the same time. To this end, firstly, three types of batch processing are defined and formalised, which are simultaneous, sequential, and concurrent batch processing. Using these definitions, the Batch Organisation of Work Identification algorithm (BOWI) is developed to gather knowledge on batch processing from event logs. When certain cases fulfil the conditions associated to a particular batch processing type, the algorithm groups them in a case set. These case sets are then used to calculate batch processing metrics such as the frequency of batches, the size of batches, and the duration and waiting time of activities in batches. These metrics and the BOWI-algorithm are evaluated on both artificial event logs and on two real-life datasets.

After the development of the artifacts according to the artifact requirements, the evaluation phase of the design science research framework includes the evaluation of the artifacts concerning these requirements. From the list of possible evaluation techniques in the field of design science, this thesis uses expert evaluation and illustrative scenarios to demonstrate the applicability of the presented log-based process metrics and their added value in the light of operational excellence. All metrics are applied to both an artificial event log and a real-life event log of a Belgian utilities company. From the application to the real-life case study, it can be concluded that the presented metrics provide the case study company with a clear picture of the present process behaviour, without any influences from underlying algorithms or assumptions. Moreover, it can be stated from this analysis that different groups of event logs can be more easily and correctly compared and analysed, both over time and based on certain case attributes such as the geographical region or the building. Besides that, the case study shows that the applied metrics are supporting some important operational excellence principles as they are based on the reduction of waste and non-value-adding activities and the management of variation and structuredness, which were stated to be requirements to reach operational excellence. Firstly, the different lean management principles of waste are supported by the presented metrics. For instance, metrics such as the throughput time and the ones concerning rework provide the company with an overview of which steps in the process and which features of products and services add value, and which can be identified as waste. Moreover, the value stream, or the sequence of activities actually adding value to the process, can consequently be defined by the business people, based on the findings from the metrics. Next to this, by removing the non-value-adding activities, such as repetitions, work in queue, or batch processing, interruptions in the process are reduced. Another lean management principle that has been covered by the presented metrics is 'pursue perfection and continuous improvement' as the application of the metrics is easily repeatable. As most existing lean management tools are rather qualitative, the presented metrics are a useful quantitative and objective addition to existing tools.

The metrics concerning the number of process variants, the rework metrics, and the activity presence are examples of metrics that support the six sigma philosophy, by focusing on the reduction of variability in business processes in order to minimise defects and errors. But also the organisational metrics measuring the variation in resources focus on optimisation of the workload and resource specialisation by analysing where variation takes place. Also different steps within the DMAIC-cycle of Six Sigma are covered by the presented metrics, with the most added value in the analysis step, where the amount of variation and the underlying source of variation is analysed with the different metrics.

However, the presented metrics are only methods to analyse the behaviour within event logs. They provide an objective analysis of business processes and are not providing any interpretation for the business people. These interpretations, and the resulting improvement steps that should be taken to reduce variation and consequently improve the process, are steps the business people should take based on the provided analyses in this dissertation.

8.2 Future research

Throughout this dissertation, a number of recommendations for future research have been made. An overview of the most important ones is provided here.

With respect to one of the metrics of the time dimension, future work is required to investigate the waiting time on the level of a specific activity. Waiting time is identified as one of the most important causes of waste and variation within a business process, and can be calculated with the presented metrics on the level of the entire event log, the case, the trace, and the specific resources. In addition to this, it can be interesting to narrow down to the waiting time of a specific activity, which is the time between the arrival of the activity in the trace and the start of the execution of this activity. However, current research states that information on the concurrency of activities, and thus a process model, is required to calculate the waiting time on the level of an activity, while this dissertation focuses on analyses without the need of a process model. Future research on the calculation of the waiting time of activities without the need of a process model is therefore recommended.

Another interesting metric that can be added to the provided list is a measure that identifies the consensus sequence within an event log, which is the trace or process variant from which other traces differ the least. Comparing other traces to this consensus sequence would provide companies with better insights in their business processes as it shows how much reality differs from the largest common denominator.

Regarding the process metrics that have been presented, some challenges and different perspectives can provide an even better indication of the process behaviour observed in an event log. For example, indicators or metrics should not be considered to be independent from each other and the results of one metric can be the input of or complement other metrics. Next to this, the analysis of the metrics and the results of the metrics over time is not discussed in this dissertation. Specific guidelines or a roadmap that should be taken to get insights in the evolution of certain aspects over time can be interesting additions. Moreover, all developed metrics assume the presence of both start and complete timestamps, and resource information for each task that is executed within a process. However, this assumption does not always hold in real-life event logs such as the case study that is analysed in Chapter 7, where only one timestamp was present for each activity. In this respect, further research on start time estimation is recommended, in order to calculate bottleneck activities, actual processing time for each work request compared to inactivity, and idle time between activities. Next to this, actual resources should be tracked in order to calculate the specialisation of resources, their (in)efficiency, and the actual active time of resources compared to their inactivity. Related to this matter, the process mining community should continue to build partnerships with industry to influence how process execution data is recorded. This way, more real-life event logs containing both start and complete timestamps and accurate resource information might become available.

In relation to the metrics that have been developed concerning the concept of batch processing, future work can extend the BOWI-algorithm to retrieve even more versatile batch processing insights from an event log. Firstly, while the BOWI-algorithm currently focuses on gathering batch processing knowledge on the level of a single activity, this perspective can be broadened by considering multiple consecutive activities. Secondly, insights on the logic behind batch formation can be added as an analysis dimension. While BOWI currently aims to identify which cases are batched, it can be useful to identify the reasoning behind batching behaviour through the identification of batch activation rules.

Finally, based on the evaluation performed in Chapter 7 and the future research ideas stated above, recommendations can be done to the case study company, and consequently to other companies, to get even more benefit from the analyses that can be done with the current set of log-based process metrics presented in this dissertation. The added value of the presented metrics could be improved by introducing specific benchmarks against which the results can be evaluated. And related to this, finding a link between the business' KPIs and the presented metrics could be of interest, to analyse which metrics should be calculated to cover which KPIs.

Appendix A

Interview questions

The goal of this interview is to analyse the problems and obstacles that one encounters when visualising and optimising a business process. The insights that arose from these interviews are presented in Chapter 4. The following questions are asked to each interview respondent.

Start of the interview

- Is it ok if this interview is recorded?

Information about the respondent

- Name?
- Company?
- Function?
- How long do you have experience in this function?

The first set of questions will be rather broad, in order to let you tell your own story. After that, more specific questions will be asked concerning different related topics.

Information about the business processes in your company

- To what extent are you involved in the execution of business processes? Can you explain what your responsibility is towards the business processes in your company?

- Can you describe one or some of the processes that you work with? What is the purpose of them?
- Is it possible to receive an overview of the process(es) that you work with? And the documentation of the(se) process(es) (if it exists).

In case there is no specific process with which you work, the following questions can be answered about other business processes in your company.

- Are the processes implemented or automated in an information system? Or are they executed manually?
- Are the processes documented? Is there a process model or process design present?

Information about the process

- What is, according to you, a good business process?
- Do you have a good notice of how good the process is running, based on the information that is available in your company?
- Which information adds to this or is missing to answer this question? And on which level of granularity is this information captured?
- Do you think there are categories in which the performance of a process can be subdivided?
- Are you involved in the evaluation of the performance of the process? If yes, in which categories of process performance?
- Are you actively improving the process? If yes, in which categories of process performance?
- Do you think there are categories of process performance that are more important than other categories?
- Do you think there are different levels of analysis on which the performance of a process can be measured?

The following questions go deeper into the different categories of process performance.

Costs

- What do you think are indicators or measures to define the costs of a certain process or a certain activity?

- Are there people already involved in defining and analysing the costs of a process or activity?
- Are you satisfied about the costs of the process or of certain activities in the process? What would you do to decrease the costs of the process or of a certain activity within the process?

Quality

- What do you think are indicators or measures to define the quality of a certain process or a certain activity?
- Are there people already involved in defining and analysing the quality of a process or activity?
- Are you satisfied about the quality of the process or of certain activities in the process? What would you do to increase the quality of the process or of a certain activity within the process?

Time

- What do you think are indicators or measures to define the time dimension of a certain process or a certain activity?
- Are there people already involved in defining and analysing the time dimension of a process or activity?
- Are you satisfied about the timing of the process or of certain activities in the process? What would you do to improve the timing of the process or of a certain activity within the process?

Structuredness

- What do you think are indicators or measures to define the structuredness of a certain process or a certain activity?
- Are there people already involved in defining and analysing the structuredness of a process or activity?
- Are you satisfied about the structuredness of the process or of certain activities in the process? What would you do to improve the structuredness of the process or of a certain activity within the process?

Value-adding activities

- What do you think are indicators or measures to define if certain activities in a process (do not) add value to the process?
- What would you do or what do you do with activities that do not add value to the process?
- What is already being done to map or follow the process?

Problems with business processes

- Do you experience any problems with the business process or with a certain part of the process?
- Would you change the implementation or the method of execution of the activities, if this is possible?

Finalisation

- Do you have any other remarks to add after this interview?
- Are there people in your company that are responsible for the optimisation of the business process? Or do you know people that I can ask the same questions?
- If I would have any other question afterwards, or if something needs to be clarified, can I contact you again via e-mail or phone?

Appendix B

Overview of the log-based process metrics

Table B.1: Log-based process metrics of the time dimension.

Metric	Level of analysis	Description
Metric class: Duration		
Throughput time	Log	The summary statistics of the throughput time of all cases in the entire event log.
Throughput time	Case	The total throughput time per case.
Throughput time	Trace	The summary statistics of the throughput time per trace.
Metric class: Actual processing time or active time		
Processing time	Log	The summary statistics of the actual processing time of all cases in the entire event log.
Processing time	Case	The total processing time per case (the sum of the processing time of all activities in the case).
Processing time	Trace	The summary statistics of the actual processing time of a specific trace.

Table continued on the next page

Table B.1: Log-based process metrics of the time dimension (continued).

Metric	**Level of analysis**	**Description**
Processing time	Activity	The summary statistics of the duration of each specific activity in the entire event log.
Processing time	Resource	The summary statistics of the processing time per case for each specific resource in the entire event log.
Processing time	Resource-activity	The summary statistics of the processing time per case for each specific resource-activity combination in the entire event log.
Metric class: Waiting time		
Waiting time	Log	The summary statistics of the waiting time of all cases in the entire event log.
Waiting time	Case	The total waiting time per case.
Waiting time	Trace	The summary statistics of the waiting time of a specific trace.
Waiting time	Resource	The total amount of waiting time for each specific resource in the entire event log.

Table B.2: Log-based process metrics of the structuredness dimension.

Metric	**Level of analysis**	**Description**
Variance metrics		
Number of traces	Log	The absolute number of traces in the log and the average trace coverage of the log.
Trace length	Log	The summary statistics of the number of activities in each trace.

Table continued on the next page

Table B.2: Log-based process metrics of the structuredness dimension (continued).

Metric	Level of analysis	Description
Trace length	Case	The total number of activity executions per case.
Trace length	Trace	The absolute and relative frequency of activities in a specific trace (compared to the average trace length of the top 80).
Trace coverage	Log	The absolute and relative number of traces that cover a certain percentage (default: 80 %) of the log.
Trace coverage	Case	The absolute and relative number of the coverage of the corresponding trace per case.
Trace coverage	Trace	The absolute and relative frequency of each trace together with the cumulative sum.
Activity presence	Activity	The absolute and relative number of cases where each specific activity is present.
Start activities	Log	The absolute and relative number of distinct activities that are the first activity of a case.
Start activities	Case	The activity that occurs the first per case.
Start activities	Activity	The absolute and relative number of cases that start with a specific activity.
Start activities	Resource	The absolute and relative number of times that a resource executes the start activity of a case.
Start activities	Resource-activity	The absolute and relative number of times that each resource-activity combination includes the activity that is executed first in a case.
End activities	Log	The absolute and relative number of distinct activities that are the last activity of a case.
End activities	Case	The activity that occurs the last per case.

Table continued on the next page

Table B.2: Log-based process metrics of the structuredness dimension (continued).

Metric	Level of analysis	Description
End activities	Activity	The absolute and relative number of cases that end with a specific activity.
End activities	Resource	The absolute and relative number of times that a resource executes the end activity of a case.
End activities	Resource-activity	The absolute and relative number of times that each resource-activity combination includes the last activity of a case.
Rework metrics		
Activity frequency	Log	The summary statistics of the number of times a distinct activity occurs in a case over the entire event log.
Activity frequency	Case	The absolute and relative number of distinct activities in each case.
Activity frequency	Trace	The absolute and relative number of distinct activities in each specific trace.
Activity frequency	Activity	The absolute and relative number of times each activity is performed in a case.
Number of self-loops	Log	The summary statistics of the number of repeat and redo self-loops within a trace.
Number of self-loops	Case	The absolute and relative number of repeat and redo self-loops within each specific case.
Number of self-loops	Activity	The absolute and relative number of repeat and redo self-loops for each specific activity.
Number of self-loops	Resource	The absolute and relative number of repeat and redo self-loops executed by each resource in the entire event log.

Table continued on the next page

Table B.2: Log-based process metrics of the structuredness dimension (continued).

Metric	**Level of analysis**	**Description**
Number of self-loops	Resource-activity	The absolute and relative number of repeat and redo self-loops executed by each resource-activity combination in the entire event log.
Size of self-loops	Log	The summary statistics of the size of repeat and redo self-loops in the entire event log (excluding activities without self-loops).
Size of self-loops	Case	The summary statistics of the size of repeat and redo self-loops in a specific case (excluding activities without self-loops).
Size of self-loops	Activity	The summary statistics of the size of self-loops of a specific activity (excluding activities without self-loops).
Size of self-loops	Resource	The summary statistics of the size of self-loops executed by each resource in the entire event log.
Size of self-loops	Resource-activity	The summary statistics of the size of self-loops executed per resource-activity combination.
Number of repetitions	Log	The summary statistics of the number of repeat and redo repetitions of an activity in the entire event log.
Number of repetitions	Case	The absolute and relative number of times an activity is repeated or redone within each case.
Number of repetitions	Activity	The absolute and relative number of times a specific activity is repeated or redone within a case.
Number of repetitions	Resource	The absolute and relative number of times a repeat or redo repetition of an activity occurs per resource.

Table continued on the next page

Table B.2: Log-based process metrics of the structuredness dimension (continued).

Metric	**Level of analysis**	**Description**
Number of repetitions	Resource-activity	The absolute and relative number of times a repeat or redo repetition of an activity occurs per resource-activity combination.
Size of repetitions	Log	The summary statistics of the size of repeat and redo repetitions in the entire event log (excluding activities within a self-loop).
Size of repetitions	Case	The summary statistics of the size of repeat and redo repetitions in each specific case (excluding activities within a self-loop).
Size of repetitions	Activity	The summary statistics of the size of repeat and redo repetitions of each specific activity (excluding activities within a self-loop).
Size of repetitions	Resource	The summary statistics of the size of repeat and redo repetitions executed by each resource in the entire event log (excluding activities within a self-loop).
Size of repetitions	Resource-activity	The summary statistics of the size of repeat and redo repetitions executed per resource-activity combination (excluding activities within a self-loop).
Resource metrics		
Resource frequency	Log	The summary statistics of the number of times a resource executes an activity in the entire event log.
Resource frequency	Case	The summary statistics of the number of times a resource executes an activity in each case.
Resource frequency	Activity	The summary statistics of the number of times a resource executes each activity.

Table continued on the next page

Table B.2: Log-based process metrics of the structuredness dimension (continued).

Metric	**Level of analysis**	**Description**
Resource frequency	Resource	The absolute and relative number of times each resource executes an activity in the entire event log.
Resource frequency	Resource-activity	The absolute and relative number of times each resource-activity combination occurs in the entire event log. Two different relative numbers are provided, one from a resource perspective and one from an activity perspective.
Resource involvement	Case	The absolute and relative number of distinct resources executing activities in each case.
Resource involvement	Resource	The absolute and relative number of cases in which each resource is involved.
Resource involvement	Resource-activity	The absolute and relative number of cases in which each resource-activity combination is involved.
Resource specialisation	Log	The summary statistics of the number of distinct activities executed per resource on the level of the entire event log.
Resource specialisation	Case	The summary statistics of the number of distinct activities executed per resource on the level of each case.
Resource specialisation	Activity	The absolute and relative number of distinct resources that execute each activity in the entire event log.
Resource specialisation	Resource	The absolute and relative number of distinct activities that each resource executes in the entire event log.

Appendix C

Batch organisation of work identification algorithm

Algorithm 1 BOWI-algorithm: pseudocode

Input: *eventLog*: an event log (list of complex objects representing events), *controlFlowNotion*: knowledge on the prior activity that is executed for a case to support (when required) arrival event imputation, *tolerances*: time tolerances for sequential batch processing

Output: a: activity log with batching information (list of complex objects representing activity instances)

1: *eventLog* ← ADDARRIVALEVENTS(*eventLog*, *controlFlowNotion*)
▷*imputes (when required) arrival events using knowledge on the prior activity executed for a case*
2: a ← CONVERTTOACTIVITYLOG(*eventLog*)
▷*creates activity instances by mapping corresponding events*
3: a ← SORTACTIVITYLOG(a)
▷*sort rows in activity log based on variables in following order: activity, resource, start timestamp and complete timestamp*
4: a ← REMOVEIMMEDIATEREWORK(a)
▷*removes immediate rework from activity log*
5: *batchNumber* ← 1
▷*initialise value - instances in a batch will have the same batchNumber*

```
a[1].batchNr ← batchNumber  ▷initialise batchNumber value for first instance in activity log
firstCaseStart ← a[1].start
  ▷initialise value representing the start timestamp of the first case of a potential batch
tol ← getTolerance(tolerances, a[1].activity, a[1].resource)
  ▷determines sequential batch proc.time tolerance for particular resource-activity combination
n ← numberOfRows(a)                                   ▷number of rows in activity log
for i = 2 to n do
    currentActivity ← a[i].activity                  ▷activity of instance under analysis
    priorActivity ← a[i − 1].activity                 ▷activity of prior instance in a
    currentResource ← a[i].resource
    priorResource ← a[i − 1].resource
    currentArrival ← a[i].arrival
    currentStart ← a[i].start
    priorStart ← a[i − 1].start
    currentComplete ← a[i].complete
    priorComplete ← a[i − 1].complete
    priorBatchType ← a[i − 1].batchType     ▷batch type to which the prior case belongs
    if currentActivity == priorActivity and
        currentResource == priorResource then
        if currentStart == priorStart and                ▷simultaneous batch processing
            currentComplete == priorComplete and
            priorBatchType is empty or simultaneous  then
            a[i].batchNumber ← batchNumber
            a[i].batchType ← simultaneous
            if a[i − 1].batchType is empty then
                a[i − 1].batchType ← simultaneous
            end if
        else if currentStart ≥ priorStart and              ▷concurrent batch processing
            currentStart < priorComplete and
            currentComplete ≠ priorComplete and
            priorBatchType is empty or concurrent  then
            a[i].batchNumber ← batchNumber
            a[i].batchType ← concurrent
            if a[i − 1].batchType is empty then
                a[i − 1].batchType ← concurrent
            end if
```

```
        else if currentStart ≥ priorComplete and          ▷sequential batch processing
            currentStart ≤ priorComplete + tol and
            currentArrival ≤ firstCaseStart and
            !resourceActive(a, currentResource, priorComplete, currentStart) and
            priorBatchType is empty or sequential then
            a[i].batchNumber ← batchNumber
            a[i].batchType ← sequential
            if a[i − 1].batchType is empty then
                a[i − 1].batchType ← sequential
            end if
        else                                               ▷start a new batch
            batchNumber ← batchNumber + 1
            a[i].batchNumber ← batchNumber
            firstCaseStart ← currentStart
        end if
    else        ▷subsequent instances belong to different resource-activity combination
        batchNumber ← batchNumber + 1
        a[i].batchNumber ← batchNumber
        firstCaseStart ← currentStart
        tol ← getTolerance(tolerances, currentActivity, currentResource)
  ▷adjust tolerance
    end if
end for
return a                          ▷returns activity log enriched with batching information
```

Bibliography

[1] Aboelmaged, M. G., 2010. Six Sigma quality: a structured review and implications for future research. International Journal of Quality & Reliability Management 27 (3), 268–317.

[2] Adesola, S., Baines, T., 2005. Developing and evaluating a methodology for business process improvement. Business Process Management Journal 11 (1), 37–46.

[3] Alturki, A., Bandara, W., Gable, G. G., 2012. Design science research and the core of information systems. Lecture Notes in Computer Science 7286, 309–327.

[4] Alturki, A., Gable, G. G., Bandara, W., 2011. A design science research roadmap. Lecture Notes in Computer Science 6629, 107–123.

[5] Alturki, A., Gable, G. G., Bandara, W., 2013. The design science research roadmap: in progress evaluation. Proceedings of the 2013 Pacific Asia Conference on Information Systems.

[6] Andersson, R., Eriksson, H., Torstensson, H., 2006. Similarities and differences between TQM, six sigma and lean. The TQM Magazine 18 (3), 282–296.

[7] Antony, J., 2006. Six Sigma: a strategy for supporting innovation in pursuit of business excellence. International Journal of Technology Management 37 (1-2), 8–12.

[8] Antony, J., 2006. Six sigma for service processes. Business Process Management Journal 12 (2), 234–248.

[9] Antony, J., Escamilla, J. L., Caine, P., 2003. Lean sigma production and supply chain management. Manufacturing Engineer 82 (2), 40–42.

[10] Anupindi, R., Chopra, S., Deshmukh, S. D., Van Mieghem, J. A., Zemel, E., 1999. Managing business process flows. Prentice Hall.

[11] Arbulu, R., Tommelein, I., Walsh, K., Hershauer, J., 2003. Value stream analysis of a re-engineered construction supply chain. Building Research & Information 31 (2), 161–171.

[12] Arnheiter, E. D., Maleyeff, J., 2005. The integration of lean management and Six Sigma. The TQM magazine 17 (1), 5–18.

[13] Ayanso, A., Lertwachara, K., Vachon, F., 2011. Design and behavioral science research in premier IS journals: evidence from database management research. Lecture Notes in Computer Science 6629, 138–152.

[14] Baskerville, R., 2008. What design science is not. European Journal of Information Systems 17 (5), 441–443.

[15] Baskerville, R., Pries-Heje, J., Venable, J. R., 2009. Soft design science methodology. Proceedings of the 4th international conference on design science research in information systems and technology.

[16] Becker, J., Beverungen, D., Knackstedt, R., Matzner, M., Mueller, O., Pöppelbuss, J., 2009. A framework for design research in the service science discipline. Proceedings of the 2009 Americas Conference on Information Systems.

[17] Bhagwat, R., Sharma, M. K., 2007. Performance measurement of supply chain management: A balanced scorecard approach. Computers & Industrial Engineering 53 (1), 43–62.

[18] Bigelow, M., 2002. How to achieve operational excellence. Quality Progress 35 (10), 70.

[19] Braun, R., M. B., Wendler, H., Esswein, W., 2015. Proposal for requirements driven design science research. Lecture Notes in Computer Science 9073, 135–151.

[20] Briggs, R. O., Schwabe, G., 2011. On expanding the scope of design science in IS research. Lecture Notes in Computer Science 6629, 92–106.

[21] Brucker, P., 2007. Scheduling algorithms. Vol. 3. Springer.

[22] Buijs, J. C. A. M., 2014. Flexible Evolutionary Algorithms for Mining Structured Process Models. Ph.D. thesis, Eindhoven Unversity of Technology.

[23] Chang, W., Cheng, J., Allaire, J., Xie, Y., McPherson, J., 2017. shiny: Web Application Framework for R. R package version 1.0.5.

[24] Charmaz, K., Belgrave, L. L., 2007. Grounded theory. Wiley Online Library.

[25] Cohen, J., 1988. Statistical power analysis for the behavioral sciences. Routledge.

[26] Cohen, J., 1992. A power primer. Psychological Bulletin 112, 155–159.

[27] Conboy, K., Gleasure, R., Cullina, E., 2015. Agile Design Science Research. Lecture Notes in Computer Science 9073, 168–180.

[28] Conger, S., 2010. Six sigma and business process management. In: Handbook on Business Process Management 1. Springer, pp. 127–146.

[29] Creemers, M., Jans, M., 2016. Social mining as a knowledge management solution. CEUR Workshop Proceedings 1612, 57–64.

[30] Crosby, P. B., 1979. Quality is free: The art of making quality certain. Vol. 94. McGraw-Hill.

[31] Cross, N., 2001. Designerly ways of knowing: design discipline versus design science. Design issues 17 (3), 49–55.

[32] Cua, K. O., McKone, K. E., Schroeder, R. G., 2001. Relationships between implementation of TQM, JIT, and TPM and manufacturing performance. Journal of Operations Management 19 (6), 675–694.

[33] Dahlgaard, S. M. P., 1999. The evolution patterns of quality management: some reflections on the quality movement. Total quality management 10 (4-5), 473–480.

[34] Dahlgaard, S. M. P., Andersson, R., Eriksson, H., Torstensson, H., 2006. Similarities and differences between TQM, six sigma and lean. The TQM magazine 18 (3), 282–296.

[35] Dahlgaard, S. M. P., Dahlgaard, J. J., Dahlgaard-Park, S., 2006. Lean production, six sigma quality, TQM and company culture. The TQM magazine 18 (3), 263–281.

[36] De Leoni, M., Maggi, F. M., van der Aalst, W. M. P., 2012. Aligning Event Logs and Declarative Process Models for Conformance Checking. Lecture Notes in Computer Science 7481, 82–97.

[37] de Medeiros, A. K. A., 2006. Genetic process mining. Ph.D. thesis, Eindhoven University of Technology.

[38] de Medeiros, A. K. A., van Dongen, B. F., van der Aalst, W. M. P., Weijters, A. J. M. M., 2004. Process mining: Extending the alpha-algorithm to mine short loops. Tech. Rep. BETA Working Paper Series 113.

[39] de Medeiros, A. K. A., Weijters, A. J., van der Aalst, W. M. P., 2007. Genetic process mining: an experimental evaluation. Data Mining and Knowledge Discovery 14 (2), 245–304.

[40] De Toni, A., Tonchia, S., 2001. Performance measurement systems. Models, characteristics and measures. International Journal of Operations & Production Management 21 (1/2), 46–71.

[41] De Weerdt, J., De Backer, M., Vanthienen, J., Baesens, B., 2011. A critical evaluation study of model-log metrics in process discovery. Business Process Management Workshops 66, 158–169.

[42] De Weerdt, J., De Backer, M., Vanthienen, J., Baesens, B., 2012. A multi-dimensional quality assessment of state-of-the-art process discovery algorithms using real-life event logs. Information Systems 37 (7), 654–676.

[43] Deming, W. E., 1993. The new economics for industry, government, education. MIT Press.

[44] Dobson, G., Tilson, D., Tilson, V., 2015. Optimizing the timing and number of batches for compounded sterile products in an in-hospital pharmacy. Decision Support Systems 76, 53–62.

[45] Dooley, K., 2000. The paradigms of quality: evolution and revolution in the history of the discipline. Advances in the management of organizational quality 5, 1–28.

[46] Drohomeretski, E., Gouvea da Costa, S. E., Pinheiro de Lima, E., Garbuio, P. A. d. R., 2014. Lean, Six Sigma and Lean Six Sigma: an analysis based on operations strategy. International Journal of Production Research 52 (3), 804–824.

[47] Dumas, M., La Rosa, M., Mendling, J., Reijers, H. A., et al., 2013. Fundamentals of business process management. Springer.

[48] Dumas, M., van der Aalst, W. M. P., Ter Hofstede, A., 2005. Process-aware information systems: bridging people and software through process technology. John Wiley & Sons.

[49] Dwivedi, N., Purao, S., Straub, D. W., 2014. Knowledge Contributions in Design Science Research: A Meta-Analysis. Lecture Notes in Computer Science 8463, 115–131.

[50] Eisenhardt, K. M., 1989. Building theories from case study research. Academy of management review 14 (4), 532–550.

[51] Emiliani, M. L., 1998. Lean behaviors. Management Decision 36 (9), 615–631.

[52] Folino, F., Greco, G., Guzzo, A., Pontieri, L., 2009. Discovering expressive process models from noised log data. Proceedings of the 2009 international database engineering & applications symposium, 162–172.

[53] Frei, F. X., 2006. Breaking the trade-off between efficiency and service. Harvard business review 84 (11).

[54] Fursule, N. V., Bansod, S., Fursule, S. N., 2012. Understanding the benefits and limitations of six sigma methodology. International journal of scientific and research publications 2 (1), 1–9.

[55] Garvin, D. A., 1988. Managing quality: The strategic and competitive edge. Simon and Schuster.

[56] Gill, T. G., Hevner, A. R., 2013. A fitness-utility model for design science research. ACM Transactions on Management Information Systems 4 (2).

[57] Goedertier, S., Martens, D., Vanthienen, J., Baesens, B., 2009. Robust process discovery with artificial negative events. The Journal of Machine Learning Research 10, 1305–1340.

[58] Goldkuhl, G., Lind, M., 2010. A multi-grounded design research process. Lecture Notes in Computer Science 6105, 45–60.

[59] Goldratt, E. M., 1990. Theory of constraints. North River Press.

[60] Goldratt, E. M., Cox, J., 1984. The Goal. North River Press.

[61] Gregor, S., Hevner, A. R., 2013. Positioning and presenting design science research for maximum impact. MIS quarterly 37, 337–355.

[62] Griesberger, P., Leist, S., Zellner, G., 2011. Analysis of techniques for business process improvement. Proceedings of the 19th European Conference on Information Systems.

[63] Günther, C. W., van der Aalst, W. M. P., 2007. Fuzzy mining-adaptive process simplification based on multi-perspective metrics. Lecture Notes in Business Information Processing 43, 328–343.

[64] Günther, C. W., Verbeek, H. M. W., 2014. Xes-standard definition. Eindhoven University of Technology, Tech. Rep.

[65] Harry, M. J., 1998. Six sigma: a breakthrough strategy for profitability. Quality progress 31 (5), 60–64.

[66] Heckl, D., Moormann, J., 2010. Process performance management. In: Handbook on Business Process Management 2. Springer, pp. 115–135.

[67] Henn, S., Koch, S., Wäscher, G., 2012. Order batching in order picking warehouses: a survey of solution approaches. Springer.

[68] Hevner, A., March, S. T., Park, J., Ram, S., 2004. Design science in information systems research. MIS quarterly 28 (1), 75–105.

[69] Hines, P., Rich, N., 1997. The seven value stream mapping tools. International Journal of Operations & Production Management 17 (1), 46–64.

[70] Huang, Z., Xudong, L., Huilong, D., 2012. Resource behavior measure and application in business process management. Expert Systems with Applications 39 (7), 6458–6468.

[71] Iivari, J., 2015. Distinguishing and contrasting two strategies for design science research. European Journal of Information Systems 24 (1), 107–115.

[72] Jacobs, F. R., Chase, R. B., Chase, R., 2010. Operations and supply chain management. McGraw-Hill/Irwin.

[73] Janssenswillen, G., 2017. bupaR: Business Process Analysis in R. R package version 0.3.2.

[74] Janssenswillen, G., 2017. edeaR: Exploratory and Descriptive Event-Based Data Analysis. R package version 0.7.2.

[75] Janssenswillen, G., Depaire, B., Swennen, M., Jans, M., Vanhoof, K., 2018. bupar: A framework for reproducible business process analysis. Knowledge-Based Systems (submitted).

[76] Janssenswillen, G., Jouck, T., Creemers, M., Depaire, B., 2016. Measuring the quality of models with respect to the underlying system: An empirical study. Lecture Notes in Computer Science 9850, 73–89.

[77] Janssenswillen, G., Swennen, M., Depaire, B., Jans, M., Vanhoof, K., 2015. Enabling event-data analysis in R: Demonstration. CEUR Workshop Proceedings 1527, 189–198.

[78] Johannesson, P., Perjons, E., 2014. An introduction to design science. Springer.

[79] Johannsen, F., Leist, S., Zellner, G., 2011. Six sigma as a business process management method in services: analysis of the key application problems. Information Systems and e-Business Management 9 (3), 307–332.

[80] Jugdev, K., Mathur, G., 2012. Classifying project management resources by complexity and leverage. International Journal of Managing Projects in Business 5 (1), 105–124.

[81] Kerzner, H., 2013. Project Management: A Systems Approach to Planning, Scheduling, and Controlling. John Wiley & Sons.

[82] Kis, I., Bachhofner, S., Di Ciccio, C., Mendling, J., 2017. Towards a data-driven framework for measuring process performance. Lecture Notes in Business Information Processing 287, 3–18.

[83] Klefsjö, B., Wiklund, H., Edgeman, R. L., 2001. Six Sigma seen as a methodology for total quality management. Measuring Business Excellence 5 (1), 31–35.

[84] Koning, H., Verver, J. P., Heuvel, J., Bisgaard, S., Does, R. J., 2006. Lean six sigma in healthcare. Journal for Healthcare Quality 28 (2), 4–11.

[85] Kotonya, G., Sommerville, I., 1998. Requirements engineering: processes and techniques. Wiley.

[86] Kumar, M., Antony, J., Singh, R. K., Tiwari, M. K., Perry, D., 2006. Implementing the Lean Sigma framework in an Indian SME: a case study. Production Planning and Control 17 (4), 407–423.

[87] Kwak, Y. H., Anbari, F. T., 2006. Benefits, obstacles, and future of six sigma approach. Technovation 26 (5), 708–715.

[88] Leemans, S., Fahland, D., van der Aalst, W. M. P., 2015. Using life cycle information in process discovery. Lecture Notes in Business Information Processing 256, 204–217.

[89] Leemans, S. J., Fahland, D., van der Aalst, W. M. P., 2014. Discovering block-structured process models from incomplete event logs. Lecture Notes in Computer Science 7927, 91–110.

[90] Liu, J., Hu, J., 2007. Dynamic batch processing in workflows: model and implementation. Future Generation Computer Systems 23 (3), 338–347.

[91] Liu, J., Wen, Y., Li, T., Zhang, X., 2011. A data-operation model based on partial vector space for batch processing in workflow. Concurrency and Computation: Practice and Experience 23 (16), 1936–1950.

[92] Mabin, V. J., Balderstone, S. J., 2003. The performance of the theory of constraints methodology: analysis and discussion of successful TOC applications. International Journal of Operations & Production Management 23 (6), 568–595.

[93] Maleyeff, J., Arnheiter, E. A., Venkateswaran, V., 2012. The continuing evolution of Lean Six Sigma. The TQM Journal 24 (6), 542–555.

[94] March, S. T., Smith, G. F., 1995. Design and natural science research on information technology. Decision support systems 15 (4), 251–266.

[95] March, S. T., Storey, V. C., 2008. Design science in the information systems discipline: an introduction to the special issue on design science research. MIS Quarterly 32 (4), 725–730.

[96] Martin, N., Bax, F., Depaire, B., Caris, A., 2016. Retrieving resource availability insights from event logs. Proceedings of the 2016 IEEE International Conference on Enterprise Distributed Object Computing, 69–78.

[97] Martin, N., Depaire, B., Caris, A., 2016. The use of process mining in business process simulation model construction: structuring the field. Business & Information Systems Engineering 58 (1), 73–87.

[98] Martin, N., Swennen, M., Depaire, B., Jans, M., Caris, A., Vanhoof, K., 2015. Batch processing: definition and event log identification. CEUR Workshop Proceedings 1527, 137–140.

[99] Martin, N., Swennen, M., Depaire, B., Jans, M., Caris, A., Vanhoof, K., 2017. Retrieving batch organisation of work insights from event logs. Decision Support Systems 100, 119–128.

[100] McBride, D., 2016. The process of research in psychology. Sage.

[101] McKay, P., Marshall, J., 2005. A Review of Design Science in Information Systems. Proceedings of the 16th Australasian Conference on Information Systems and Technology.

[102] McManus, H., Millard, R., 2002. Value stream analysis and mapping for product development. Proceedings of the 23rd International Council of the Aeronautical Sciences.

[103] Melton, T., 2005. The benefits of lean manufacturing: what lean thinking has to offer the process industries. Chemical engineering research and design 83 (6), 662–673.

[104] Mettler, T., Eurich, M., Winter, R., 2014. On the use of experiments in design science research: a proposition of an evaluation framework. Communications of the Association for Information Systems 34 (1), 223–240.

[105] Mullarkey, M. T., Hevner, A. R., 2015. Entering Action Design Research. Lecture Notes in Computer Science 9073, 121–134.

[106] Munoz-Gama, J., 2014. Conformance checking and diagnosis in process mining. Ph.D. thesis, Universitat Politècnica de Catalunya, Barcelona.

[107] Murthy, P., 2005. Production and operations management. New Age International Publishers, New Delhi.

[108] Nakatumba, J. J., 2013. Resource-aware business process management: analysis and support. Ph.D. thesis, Eindhoven University of Technology.

[109] Näslund, D., 2008. Lean, six sigma and lean sigma: fads or real process improvement methods? Business Process Management Journal 14 (3), 269–287.

[110] Nave, D., 2002. How to compare six sigma, lean and the theory of constraints. Quality Progress 35 (3).

[111] Neely, A., Gregory, M., Platts, K., 1995. Performance measurement system design: A literature review and research agenda. International Journal of Operations & Production Management 15 (4), 80–116.

[112] Niehaves, B., 2007. On episemological diversity in design science: New vistas for a design-oriented IS research? Proceedings of the 28th Annual International Conference on Information Systems.

[113] Nuseibeh, B., Easterbrook, S., 2000. Requirements engineering: a roadmap. Proceedings of the Conference on the Future of Software Engineering, 35–46.

[114] Offermann, P., Levina, O., Schönherr, M., Bub, U., 2009. Outline of a design science research process. Proceedings of the 4th International Conference on Design Science Research in Information Systems and Technology.

[115] Ostrowski, L., Helfert, M., Xie, S., 2012. A conceptual framework to construct an artefact for meta-abstract design knowledge in design science research. Proceedings of the 2012 Hawaii International Conference on System Science, 4074–4081.

[116] Peffers, K., Rothenberger, M., Tuunanen, T., Vaezi, R., 2012. Design science research evaluation. Lecture Notes in Computer Science 7286, 398–410.

[117] Peffers, K., Tuunanen, T., Rothenberger, M. A., Chatterjee, S., 2007. A design science research methodology for information systems research. Journal of management information systems 24 (3), 45–77.

[118] Pepper, M. P. J., Spedding, T. A., 2010. The evolution of lean Six Sigma. International Journal of Quality & Reliability Management 27 (2), 138–155.

[119] Pika, A., Wynn, M. T., Fidge, C. J., ter Hofstede, A. H. M., Leyer, M., van der Aalst, W. M. P., 2014. An extensible framework for analysing resource behaviour using event logs. Lecture Notes in Computer Science 8484, 564–579.

[120] Pinedo, M. L., 2012. Scheduling: theory, algorithms, and systems. Springer.

[121] Pufahl, L., Bazhenova, E., Weske, M., 2014. Evaluating the performance of a batch activity in process models. Lecture Notes in Business Information Processing 202, 277–290.

[122] Pufahl, L., Herzberg, N., Meyer, A., Weske, M., 2014. Flexible batch configuration in business processes based on events. Lecture Notes in Computer Science 8831, 63–78.

[123] Pufahl, L., Meyer, A., Weske, M., 2014. Batch regions: process instance synchronization based on data. Proceedings of the 2014 IEEE International Enterprise Distributed Object Computing Conference, 150–159.

[124] Pufahl, L., Weske, M., 2013. Batch activities in process modeling and execution. Lecture Notes in Computer Science 8274, 283–297.

[125] Pufahl, L., Weske, M., 2016. Batch processing across multiple business processes based on object life cycles. Lecture Notes in Business Information Processing 255, 195–208.

[126] Rajamanoharan, I. D., Collier, P., Wright, B., 2011. Positioning Six Sigma within the quality management agenda: a conceptual framework. International Journal of Quality and Innovation 1 (3), 197–216.

[127] Recker, J., Mendling, J., 2016. The state of the art of business process management research as published in the bpm conference. Business & Information Systems Engineering 58 (1), 55–72.

[128] Reijers, H. A., Mansar, S. L., 2005. Best practices in business process redesign: an overview and qualitative evaluation of successful redesign heuristics. Omega 33 (4), 283–306.

[129] Ribeiro, J., Carmona, J., Misir, M., Sebag, M., 2014. A Recommender System for Process Discovery. Proceedings of the 12th International Conference on Business Process Management, 67–83.

[130] Rossi, M., Sein, M. K., 2003. Design research workshop: a proactive research approach. 26th Information Systems Research Seminar, 9–12.

[131] Rother, M., Shook, J., 1999. Learning To See: Value Stream Mapping to Add Value and Eliminate Muda. Lean Enterprise Institute.

[132] Rozinat, A., de Medeiros, A. K. A., Günther, C. W., Weijters, A. J. M. M., van der Aalst, W. M. P., 2008. The need for a process mining evaluation framework in research and practice. Vol. 4928. pp. 84–89.

[133] Rozinat, A., van der Aalst, W. M. P., 2006. Decision mining in business processes. Business process management 4102.

[134] Rozinat, A., van der Aalst, W. M. P., 2006. Decision mining in ProM. Vol. 4102.

[135] Rozinat, A., van der Aalst, W. M. P., 2008. Conformance checking of processes based on monitoring real behavior. Information Systems 33 (1), 64–95.

[136] Salah, S., Rahim, A., Carretero, J. A., 2010. The integration of Six Sigma and lean management. International Journal of Lean Six Sigma 1 (3), 249–274.

[137] Sale, M. L., Sale, R. S., 2013. Theory of Constraints as Related to Improved Business Unit Performance. Journal of Accounting and Finance 13 (1), 108.

[138] Satyal, S., Weber, I., Paik, H.-y., Di Ciccio, C., Mendling, J., 2017. AB-BPM: Performance-driven instance routing for business process improvement. Proceedings of the 15th International Conference on Business Process Management, 113–129.

[139] Schragenheim, E., Ronen, B., 1990. Drum-buffer-rope shop floor control. Production and Inventory Management Journal 31 (3), 18–22.

[140] Schroeder, R. G., Linderman, K., Liedtke, C., Choo, A. S., 2008. Six sigma: Definition and underlying theory. Journal of Operations Management 26 (4), 536–554.

[141] Sein, M., Henfridsson, O., Purao, S., Rossi, M., Lindgren, R., 2011. Action design research. MIS Quarterly 35 (1), 37–56.

[142] Simon, H. A., 1996. The sciences of the artificial. MIT press.

[143] Song, M., Günther, C. W., van der Aalst, W. M. P., 2009. Trace clustering in process mining. Lecture Notes in Business Information Processing 17, 109–120.

[144] Song, M., van der Aalst, W. M. P., 2008. Towards comprehensive support for organizational mining. Decision Support Systems 46 (1), 300–317.

[145] Sugimori, Y., Kusunoki, K., Cho, F., Uchikawa, S., 1977. Toyota production system and kanban system materialization of just-in-time and respect-for-human system. The International Journal of Production Research 15 (6), 553–564.

[146] Suri, R., 1998. Quick response manufacturing: a companywide approach to reducing lead times. CRC Press.

[147] Suriadi, S., Wynn, M. T., Xu, J., van der Aalst, W. M. P., ter Hofstede, A. H., 2017. Discovering work prioritisation patterns from event logs. Decision Support Systems 100, 77–92.

[148] Swennen, M., Janssenswillen, G., Jans, M., Depaire, B., Vanhoof, K., 2015. Capturing process behavior with log-based process metrics. CEUR Workshop Proceedings 1527, 141–144.

[149] Swennen, M., Martin, N., Janssenswillen, G., Jans, M. J., Depaire, B., Caris, A., Vanhoof, K., 2016. Capturing resource behaviour from event logs. CEUR Workshop Proceedings 1757, 130–134.

[150] Telsang, M., 2005. Production management. S. Chand & Company Ltd.

[151] Tsironis, L. K., Psychogios, A. G., 2016. Road towards Lean Six Sigma in service industry: a multi-factor integrated framework. Business Process Management Journal 22 (4), 812–834.

[152] Tuckman, A., 1995. Ideology, quality and TQM. Routledge.

[153] Vaishnavi, V. K., Kuechler, W., 2007. Design science research methods and patterns: innovating information and communication technology. CRC Press.

[154] van Assen, M. F., 2011. Operational excellence for services (position paper).

[155] van der Aalst, W. M. P., 2011. Process mining: discovery, conformance and enhancement of business processes. Springer.

[156] van der Aalst, W. M. P., 2013. Business process management: A comprehensive survey. ISRN Software Engineering 2013, 1–37.

[157] van der Aalst, W. M. P., 2013. Mediating between modeled and observed behavior: the quest for the “right” process. IEEE International Conference on Research Challenges in Information Science, 31–43.

[158] van der Aalst, W. M. P., 2016. Process Mining: Data Science in Action. Springer.

[159] van der Aalst, W. M. P., 2016. Process mining: data science in action. Springer.

[160] van der Aalst, W. M. P., Adriansyah, A., de Medeiros, A. K. A., Arcieri, F., Baier, T., Blickle, T., Bose, J. C., Brand, P. V. D., Brandtjen, R., Buijs, J., Burattin, A., Carmona, J., Castellanos, M., Claes, J., Cook, J., Costantini, N., Curbera, F., Damiani, E., de Leoni, M., Delias, P., van Dongen, B. F., Dumas, M., Dustdar, S., Fahland, D., Ferreira, D. R., Gaaloul, W., van Geffen, F., Goel, S., Günther, C., Guzzo, A., Harmon, P., Hofstede, A. T., Hoogland, J., Ingvaldsen, J. E., Kato, K., Kuhn, R., Kumar, A., Rosa, M. L., Maggi, F.,

Malerba, D., Mans, R. S., Manuel, A., McCreesh, M., Mello, P., Mendling, J., Montali, M., Motahari-Nezhad, H. R., zur Muehlen, M., Munoz-Gama, J., Pontieri, L., Ribeiro, J., Rozinat, A., Prez, H. S., Prez, R. S., Sepúlveda, M., Sinur, J., Soffer, P., Song, M., Sperduti, A., Stilo, G., Stoel, C., Swenson, K., Talamo, M., Tan, W., Turner, C., Vanthienen, J., Varvaressos, G., Verbeek, E., Verdonk, M., Vigo, R., Wang, J., Weber, B., Weidlich, M., Weijters, T., Wen, L., Westergaard, M., Wynn, M., 2011. Process mining manifesto. Lecture Notes in Business Information Processing 99, 169–194.

[161] van der Aalst, W. M. P., Adriansyah, A., van Dongen, B., 2012. Replaying history on process models for conformance checking and performance analysis. Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery 2 (2), 182–192.

[162] van der Aalst, W. M. P., Buijs, J., Van Dongen, B., 2012. Towards improving the representational bias of process mining. Lecture Notes in Business Information Processing 116, 39–54.

[163] van der Aalst, W. M. P., Nakatumba, J., Rozinat, A., Russell, N., 2010. Business process simulation. Springer.

[164] van der Aalst, W. M. P., Rubin, V., Verbeek, H. M. W., van Dongen, B. F., Kindler, E., Günther, C. W., 2010. Process mining: a two-step approach to balance between underfitting and overfitting. Software & Systems Modeling 9 (1), 87–111.

[165] van der Aalst, W. M. P., Schonenberg, M. H., Song, M., 2011. Time prediction based on process mining. Information Systems 36 (2), 450–475.

[166] van der Aalst, W. M. P., Weijters, T., Maruster, L., 2004. Workflow mining: Discovering process models from event logs. IEEE Transactions on Knowledge and Data Engineering 16 (9), 1128–1142.

[167] van Geffen, F., Niks, R., 2013. Accelerate DMAIC using Process Mining. Proceedings of the 11th International Conference on Business Process Management (BPI Challenge).

[168] Van Vliet, H., 1993. Software engineering: principles and practice. Vol. 3. Wiley.

[169] vanden Broucke, S. K. L. M., 2014. Advances in Process Mining. Ph.D. thesis, Katholieke Universiteit Leuven.

[170] vanden Broucke, S. K. L. M., De Weerdt, J., Vanthienen, J., Baesens, B., 2013. A Comprehensive Benchmarking Framework (CoBeFra) for conformance analysis between procedural process models and event logs in ProM. Proceedings of the 2013 IEEE Symposium on Computational Intelligence and Data Mining, 254–261.

[171] Venable, J. R., 2006. A framework for design science research activities. Proceedings of the 2006 Information Resource Management Association Conference, 21–24.

[172] Venable, J. R., 2006. The role of theory and theorising in design science research. Proceedings of the 1st International Conference on Design Science in Information Systems and Technology, 1–18.

[173] Venable, J. R., 2009. Identifying and addressing stakeholder interests in design science research: an analysis using critical systems heuristics. In: Information Systems - Creativity and Innovation in Small and Medium-Sized Enterprises. Springer, pp. 93–112.

[174] Venable, J. R., Pries-Heje, J., Baskerville, R., 2012. A comprehensive framework for evaluation in design science research. Lecture Notes in Computer Science 7286, 423–438.

[175] Weijters, A. J. M. M., van der Aalst, W. M. P., 2003. Rediscovering workflow models from event-based data using little thumb. Integrated Computer-Aided Engineering 10 (2), 151–162.

[176] Weijters, A. J. M. M., van der Aalst, W. M. P., De Medeiros, A. K. A., 2006. Process mining with the heuristics miner-algorithm. Tech. Rep. BETA Working Paper Series 166, 1–34.

[177] Wen, L., van der Aalst, W. M. P., Wang, J., Sun, J., 2007. Mining process models with non-free-choice constructs. Data Mining and Knowledge Discovery 15 (2), 145–180.

[178] Wen, Y., Chen, Z., Liu, J., Chen, J., 2013. Mining batch processing workflow models from event logs. Concurrency and Computation: Practice and Experience 25 (13), 1928–1942.

[179] Wieringa, R. J., 1996. Requirements engineering: frameworks for understanding. John Wiley & Sons.

[180] Wieringa, R. J., 2009. Design science as nested problem solving. Proceedings of the 4th International Conference on Design Science Research in Information Systems and Technology.

[181] Winter, R., 2008. Design science research in Europe. European Journal of Information Systems 17 (5), 470–475.

[182] Womack, J. P., Jones, D. T., 2005. Lean Consumption. Harvard Business Review 83 (3).

[183] Worley, J. M., Doolen, T. L., 2006. The role of communication and management support in a lean manufacturing implementation. Management Decision 44 (2), 228–245.

[184] Wu, K., 2014. Taxonomy of batch queueing models in manufacturing systems. European Journal of Operational Research 237 (1), 129–135.

[185] Yang, C. C., 2010. Six Sigma and Total Quality Management. Sciyo.

[186] Zellner, G., 2011. A structured evaluation of business process improvement approaches. Business Process Management Journal 17 (2), 203–237.

Samenvatting

Organisaties bestaan tegenwoordig uit een groot aantal processen die soms sterk met elkaar verweven zijn. Hierdoor is het moeilijk om een goed overzicht te krijgen van de verschillende gegevensstromen, activiteiten en actoren in het proces. Bovendien evolueren nieuwe technologieën en klantenverwachtingen sneller dan ooit, waardoor bedrijven ernaar streven om hun processen en werkwijzen voortdurend aan te passen en te verbeteren. Om mee te gaan met veranderende omgevingen en situaties moeten bedrijfsprocessen continu gemonitord worden en moeten bedrijven procesanalysemethoden en verbeterteams inzetten. Het concept van continue verbetering is gerelateerd aan methodologieën zoals lean management, Six Sigma, business process improvement (BPI) en total quality management (TQM).

Lean management is vooral gericht op vermindering van verspilling, waaronder elemementen zoals overproductie, wachttijden, overbodige bewerkingen, vertragingen of batchverwerking vallen. De vermindering van deze verspilling kan worden geoperationaliseerd door de identificatie en analyse van de waardenstroom van een bedrijf, welke aangeeft dat enkel activiteiten uitgevoerd worden die werkelijk waarde toevoegen voor de klant van het bedrijf. Six Sigma daarentegen is meer gericht op de kwaliteit van de bedrijfsprocessen en heeft als doel het minimaliseren van het aantal fouten en defecten in het proces, om zo variantie te voorkomen en de algehele kwaliteit van de organisatie te verbeteren. De bekendste verbeteringsmethode binnen Six Sigma is de DMAIC-cyclus, welke staat voor define, measure, analyse, improve en control. Naast lean management en Six Sigma, is de theory of constraints een andere bekende verbeteringsmethodologie, die meer gericht is op het detecteren en verwijderen van beperkingen of knelpunten in bedrijfsprocessen. Hoewel deze methoden en technieken al vele jaren worden toegepast in het bedrijfsleven, zijn ze minder bruikbaar in de steeds sneller veranderende omgevingen waarin dienstenbedrijven actief zijn. Veel van deze operational excellence-technieken zijn ook gebaseerd op voornamelijk kwalitatieve en “papier-en-pen”-benaderingen, wat impliceert dat de resultaten van deze methoden, en de beslissingen die genomen worden op basis van deze resultaten, subjectief kunnen zijn en afhankelijk van de persoon die de analyses uitvoert. Bovendien zijn de meeste studies in de literatuur gericht op de toepassing van methodologieën zoals lean management en Six Sigma in productieomgevingen en zijn dienstgerichte ondernemingen tot nu toe minder overtuigd van het gebruik van deze methodologieën binnen hun organisatie, meestal omwille van de meer veranderlijke omgeving en omstandigheden.

Vanuit dit opzicht wordt process mining gezien als een potentieel hulpmiddel op het gebied van operational excellence, omdat het zich richt op de analyse van bedrijfsprocessen om inzicht te krijgen in de activiteitenstromen van een bedrijf en deze overeenkomstig te verbeteren. Process mining verwijst naar het vergaren van kennis uit gegevens over de uitvoer van processen, die worden opgeslagen in zogenaamde event logs. Het omvat hoofdzakelijk drie types, nl. (i) *discovery*, of het ontdekken van procesmodellen uit event logs, (ii) *compliance checking*, of het opvolgen en vergelijken van de ontdekte procesmodellen met de onderliggende event logs en (iii) *enhancement*, of het verbeteren van de processen. Event logs zijn het startpunt van een process mining-project en zijn samengesteld uit gegevens die verzameld worden door informatiesystemen zoals enterprise resource planning systems. Elke rij in een event log beschrijft een bepaalde gebeurtenis in het proces, zoals de start van het inboeken van een bepaalde factuur door een bepaalde medewerker, of het moment waarop een bepaalde taak wordt afgehandeld voor een specifieke klant. Een breed gamma aan algoritmes werd reeds ontwikkeld om procesmodellen te creëren op basis van event logs maar deze algoritmes zijn meestal gebaseerd op parameters en veronderstellingen die door de procesanalist gekozen worden om de analyse te vereenvoudigen. Aangezien de modellen die gebouwd worden aan de hand van deze algoritmes daardoor mogelijk te specifiek of te algemeen zijn om het werkelijke gedrag in een bedrijfsproces te weerspiegelen, omvat de definitie van process mining in dit proefschrift ook het ophalen van kennis uit event logs zonder de nood aan een tussenliggend procesmodel.

Gezien het potentieel van process mining in het domein van operational excellence is verder onderzoek rond dit topic vereist. Bestaande onderzoeksinspanningen lijken bovendien beperkt en het is in de huidige literatuur ook niet altijd duidelijk hoe bestaande technieken gebruikt worden om de analyse en optimalisatie van bedrijfsprocesprestaties te ondersteunen.

Bestaande operational excellence-technieken vereisen dus meer data-gebaseerde analyses om meer objectieve en effectieve beslissingen te kunnen nemen. Process mining is een veelbelovend domein om deze technieken te ondersteunen, maar is nog te vaak gericht op het creëren van procesmodellen. Daarom behandelt dit proefschrift twee doelstellingen om na te gaan hoe process mining kan toegepast worden op bedrijfsprocessen om bestaande operational excellence-technieken te ondersteunen. De eerste doelstelling omvat het onderzoek van het huidige probleem om zo een overzicht te creëren van de vereisten van het artefact dat nodig is om het probleem op te lossen. Hiertoe wordt eerst een literatuurstudie uitgevoerd van zowel operational excellence als van bestaande onderzoeksinspanningen van process mining in het domein van operational excellence. De bevindingen uit dit literatuuroverzicht worden vervolgens aangevuld met de bevindingen uit een lijst van interviews met bedrijfsexperten om na te gaan of de vereisten en behoeften bevestigd worden door ervaringsdeskundigen.

Op basis van de vereisten van het gewenste artefact wordt de tweede doelstelling van dit proefschrift uitgevoerd, waarin het gewenste artefact daadwerkelijk ontwikkeld wordt. Uit het probleemonderzoek op basis van literatuur en bedrijfsexperten blijkt dat er nood is aan een lijst van metrieken die rechtstreeks gebruik maken van de gegevens in een event log, zonder tussenliggend procesmodel. De lijst van inzichten die uit een event log geleerd kunnen worden, ter ondersteuning van het verbeteren van de prestaties van een bedrijfsproces, omvat elementen zoals de identificatie van verspilling, herhaling, batchverwerking, variantie en activiteiten die geen waarde toevoegen aan het proces. De metrieken die ontwikkeld worden in dit proefschrift spelen in op deze elementen en geven bijgevolg inzicht in de prestaties van een bedrijfsproces. Bovendien blijkt uit het onderzoek dat de gewenste metrieken op verschillende analyseniveaus nodig zijn, en dat niet enkel de metriek duidelijk en onbetwistbaar omschreven moet worden, maar ook de achterliggende berekeningen en de vereisten van de onderliggende data. Ten slotte benadrukken de bedrijfsexperten dat een geschikte visuele ondersteuning van de metrieken een meerwaarde zou betekenen voor de analyse van de processen.

Vertrekkende van deze vereisten, wordt een lijst van op event log data-gebaseerde metrieken voorgesteld, welke een bedrijf inzicht geven in het gedrag in een event log. De voorgestelde metrieken zijn gestructureerd op basis van twee categorieën van procesprestatiemaatstaven waarop gefocust moet worden in een bedrijfsprocesverbeteringsproject. Deze categorieën zijn tijd en structuur. De metrieken kunnen berekend worden op verschillende analyseniveaus, zoals het logniveau, caseniveau, traceniveau, activiteitniveau, resourceniveau en op het niveau van specifieke resource-activiteitcombinaties. Bovendien spelen ze in op verschillende aspecten van operational excellence zoals batchverwerking, vermindering van verspilling, herhaling en het verwijderen van activiteiten die geen waarde toevoegen. Ten slotte kunnen de voorgestelde metrieken ook op een objectieve manier ingezet worden om verschillende event logs met elkaar te vergelijken op vlak van verschillende aspecten van procesprestaties.

De metrieken bezorgen procesanalisten dus een beeld van het huidige procesgedrag en zijn bovendien allemaal geïmplementeerd in het R-pakket edeaR, waardoor ze eenvoudig toepasbaar zijn op elke event log. Om de metrieken waardevoller en toegankelijker te maken voor mensen uit het bedrijfsleven, is bovendien een dashboard met visualisaties van elke metriek ontwikkeld en voorgesteld. Naast de visualisaties bevat het dashboard ook filters om meer specifieke analyses uit te voeren en zo meer inzicht te krijgen in verschillende aspecten van de bedrijfsprocessen.

De metrieken van de tijdsdimensie meten de doorlooptijd, de daadwerkelijke verwerkingstijd en de wachttijd op verschillende analyseniveaus om inzicht te krijgen in bijvoorbeeld de duur van de cases en activiteiten. Op basis van deze statistieken kunnen aanvullende operational excellence-concepten zoals het aantal openstaande cases en knelpunten in een bedrijfsproces worden berekend.

Variabiliteit is gedefinieerd als een van de belangrijkste oorzaken van verspilling binnen een bedrijfsproces, aangezien het ervoor zorgt dat een proces afwijkt van het gewenste gedrag. Aangezien variabiliteit of ongestructureerd gedrag echter niet altijd geëlimineerd moet worden, is het belangrijk dat een organisatie leert welke soorten van ongestructureerd gedrag voorkomen in het proces dat men analyseert. Om hierin meer inzicht te krijgen, worden verschillende klassen van metrieken ontwikkeld met betrekking tot de structuur van bedrijfsprocessen, nl. variantie-, herhalings-, en organisatorische- of resourcegerelateerde metrieken. De metrieken omtrent variantie omvatten metingen voor het berekenen van onder andere het aantal en de lengte van traces, de aanwezigheid van start- en eindactiviteiten in het proces en de trace coverage in een bedrijfsproces. Herhalingsmetrieken berekenen het aantal keer dat bepaalde activiteiten herhaald worden binnen eenzelfde case, door dezelfde of door een andere resource. Er wordt ook een onderscheid gemaakt tussen self-loops waarbij de activiteiten die herhaald worden meteen op elkaar volgen, en repetitions, waarbij de uitvoering van een andere activiteit plaatsvindt tussen de voorkomens van de activiteit die herhaald wordt binnen een case. Deze metrieken omtrent herhalingen geven organisaties een inzicht in welke activiteiten herhaaldelijk worden uitgevoerd, welke resources betrokken zijn bij deze herhalingen en hoe vaak de activiteiten herhaald worden wanneer ze zich voordoen in een self-loop of repetition. De organisatorische metrieken die in dit proefschrift ontwikkeld worden, hebben betrekking op de verschillende aspecten van het procesgedrag met betrekking tot de resources, die worden gedefinieerd als de werknemers die in het proces werken, of de softwaresystemen of machines die activiteiten uitvoeren. Deze metrieken berekenen de frequentie waarop resources activiteiten uitvoeren binnen het proces, de betrokkenheid van de resources in het proces en de specialisatie van de resources. Opnieuw kunnen deze metrieken op verschillende analyseniveaus uitgevoerd worden en kunnen ze organisaties ondersteunen om bijvoorbeeld een overzicht te krijgen van de kennisdeling binnen het proces of voor het toewijzen van werknemers aan taken.

Inzicht krijgen in het gedrag van resources en de hoeveelheid tijd die ze "verspillen" binnen een proces kan erg interessant zijn voor bedrijven die hun bedrijfsprocessen wensen te optimaliseren. Gerelateerd hieraan zijn de metrieken omtrent batchverwerking, waarbij resources activiteiten uitvoeren van twee of meerdere cases op hetzelfde of bijna hetzelfde moment. Hiertoe worden ten eerste drie types van batchverwerking gedefinieerd en geformaliseerd in dit proefschrift. Op basis van deze definities wordt vervolgens een algoritme ontwikkeld om kennis over batchverwerking te verzamelen uit event logs. Indien bepaalde cases voldoen aan de voorwaarden van een bepaald batchverwerkingstype, groepeert het algoritme ze in een set. Deze sets van cases worden vervolgens gebruikt om de grootte van de batch te berekenen en om batchverwerkingsmetrieken te berekenen, zoals de duur en de wachttijd van activiteiten in een batch. Deze metrieken en het algoritme worden geëvalueerd op zowel artificiële als op event logs van reële bedrijfsprocessen.

Na het ontwikkelen van de metrieken volgens de vooropgestelde vereisten, omvat de evaluatiefase de evaluatie van de metrieken met betrekking tot deze vereisten. Alle metrieken worden toegepast op zowel een artificiële event log als op een event log van een reëel bedrijfsproces van een Belgisch nutsbedrijf. Uit deze toepassing kan geconcludeerd worden dat de voorgestelde metrieken een duidelijk beeld geven van het huidige procesgedrag, zonder de nood aan of de invloed van onderliggende algoritmes of assumpties. Bovendien kan uit deze analyse afgeleid worden dat verschillende groepen van event logs gemakkelijker en correcter vergeleken en geanalyseerd kunnen worden, zowel over de tijd heen als op basis van bepaalde caseattributen zoals de regio of de vestiging van de organisatie. Daarnaast laat de case study ook zien dat de toegepaste metrieken enkele belangrijke operational excellence-principes ondersteunen. Eerst en vooral worden verschillende principes van lean management omtrent verspillingvermindering ondersteund met de metrieken. Metrieken zoals de doorlooptijd of metrieken met betrekking tot herhaling geven de organisatie inzicht in welke stappen in het proces en welke eigenschappen van producten of diensten waarde toevoegen, en welke geïdentificeerd kunnen worden als verspilling. Daarnaast kunnen onderbrekingen of knelpunten in het proces verminderd worden door de activiteiten die geen waarde toevoegen, zoals herhalingen of batchverwerking, te verwijderen. Een ander principe van lean management dat ondersteund wordt, is het streven naar perfectie en continue verbetering, aangezien de toepassing van de metrieken eenvoudig herhaalbaar is door de organisatie. Omdat de meeste bestaande lean managementtechnieken eerder kwalitatief zijn, zijn de gepresenteerde metrieken een waardevolle kwantitatieve en objectieve aanvulling op de bestaande technieken.

De metrieken met betrekking tot het aantal procesvarianten of traces en de herhalingsmetrieken zijn voorbeelden van metrieken die de Six Sigma-filosofie ondersteunen, door de focus op de vermindering van variabiliteit in bedrijfsprocessen waardoor gebreken en fouten geminimaliseerd worden. Ook de resourcemetrieken die de variantie van resources meten, zijn gericht op de optimalisatie van de werklast en de specialisatie van resources door te analyseren waar variantie plaatsvindt.

De voorgestelde metrieken zijn echter slechts methoden om het gedrag in event logs te analyseren. Ze bieden een objectieve analyse van bedrijfsprocessen, maar ze leveren niet de bijhorende interpretatie voor de procesanalisten. Deze interpretaties, en de resulterende verbeterstappen die genomen kunnen worden om de variantie en verspilling te verminderen en zodoende het proces te verbeteren, zijn stappen die de organisaties zelf moeten nemen op basis van de geleverde analyses in dit proefschrift.