Increasing accessibility and reproducibility of process mining research in healthcare

Janssen, R.J.P.

Award date:
2011

Eindhoven, August 2011

Increasing accessibility and reproducibility of process mining research in healthcare

By Robert Janssen

BSc Medical Natural Sciences — Vrije Universiteit Amsterdam (2008)
Student identity number 0631809

in partial fulfilment of the requirements for the degree of

Master of Science
in Innovation Management

Supervisors:
dr. P.M.E. Van Gorp, TU/e, IS
dr. A.J.M.M. Weijters, TU/e, IS

TUE. School of Industrial Engineering.
Series Master Theses Innovation Management

Subject headings: process mining, ProM, healthcare, reproducibility, pattern language, SHARE, virtual machine, accessibility

Abstract

The goal of this thesis was to develop a set of best practices for process mining research in healthcare. Process mining in healthcare has been studied by several researchers already, but these studies suffer from disappointing results, which may be partly to blame to the low accessibility and reproducibility of the process mining research methodologies. Therefore, in this thesis we aimed to synthesize previous process mining research, by means of a multiple case study and expert interviews, into one new process mining methodology for healthcare that would show increased accessibility, reproducibility and results that are better, or at least comparable, to present process mining research. Pattern language was identified as the most suitable method to report the new methodology and as a result, 22 patterns that cover a variety of process mining research steps have been developed. Subsequently, 11 patterns were selected as best practices as they showed the most promising results and highest accessibility. In addition, we have introduced the concepts of virtual machines and screencasts to the field of process mining as means to increase accessibility and reproducibility of the pattern language. The results of the patterns have been validated by experts at two Dutch hospitals and showed that the process mining pattern language is indeed able to produce process models which are of similar or superior quality compared to process models created with conventional methodologies.

Preface

The report you are about to read is the result of my graduation project that was carried out at the Eindhoven University of Technology as the last hurdle to the degree of Master of Science in Innovation Management. A graduation project always requires much time and effort, both from the student and the persons supporting him. Therefore, I would like to mention and thank a few persons to acknowledge their support to me and my project.

First, I would like to thank my supervisors at the Eindhoven University of Technology who have provided their guidance throughout the whole project. Pieter, I thank you for your patience, support and suggestions during the last two years. We started with a completely different idea in mind, but I feel that we eventually made the right decisions. Your enthusiasm towards the project always provided me with new motivation whenever it was low. I also thank Ton, who has provided me with precious feedback during many stages of this project and contributed valuable knowledge on some parts of the research more specifically.

Secondly, I would like to thank all the people who have contributed to this thesis in one way or another. During this project I required the help of many people and I thank them for their time and effort to help a student achieve his goals. Special thanks go out to Jenna who has been kind enough to provide me with good feedback near the completion of the project.

Thirdly, my appreciation goes out to my fellow students with whom I have spent countless hours on the many projects we had to complete over the last few years. Together we were able to get some comic relief during the countless coffee breaks, but eventually we managed to achieve many excellent results.

Fourthly, my thanks go out to all my friends at H.I.D. Quare for all the laughter and fun they have provided me with over the last four years whenever I required distraction from studying. We have really experienced some memorable events and I hope to drop by in the near and far future to create some more.

Last, but definitely the most important, I thank my parents Richard and Marijke, and my brother Tim. They have supported me during all stages of my life and always were there for me when I needed them. Their motivation over the last few months really kept me on track and helped me to finish this project. Without them I would have definitely not made this far.

Robert Janssen

Eindhoven, August 2011

Executive summary

As the global need for healthcare keeps rising there is a continuous pressure for lower waiting times, lower costs, increased throughput and increased overall performance; now more than ever is there a need for methodologies to make the complex healthcare processes more efficient.

Previous research has identified several potential tools that could contribute to increased healthcare performance. One of these methods is process mining (i.e. the automated construction of process models from an event log using specialized software tools), which has already been studied by several scholars in a healthcare setting. However, this research is limited to several pilot studies only and the results are far from optimal. Therefore, preceding stages of this thesis served to analyse the current state of process mining in healthcare. Despite the continuous efforts of previous researchers, several problems have been identified with regard to their methodology and results.

For one, the fitness and usability of the resulting process models generally is low, which may be partly to blame to the superficial treatment of the process mining algorithms. Second, due to the overload of data pre-processing decisions, complex process mining algorithms and corresponding complex and decentralized literature, accessibility for non-experienced users is low. Third, even experienced users can get lost in the large number of process mining possibilities, relying too much on what is known and has already been used in previous studies, thereby losing the potentially powerful combinations of process mining algorithms that have not yet been explored by previous researchers. Fourth, reproducibility of process mining research is low due to the lack of detailed described methodologies, which inhibits future researchers to use the existing knowledge on process mining and forces them to reinvent the wheel over and over.

Despite the promising contributions of process mining to healthcare process research, these problems have not yet been solved or addressed in any research. A detailed and accessible process mining methodology is lacking, leaving major gaps for improvement of the current proposed, however incomplete, methodologies. A possible solution to the current problems could be a new detailed, reproducible and accessible methodology for process mining in healthcare. Therefore, during this thesis we aimed to create a set of process mining best practices that would communicate the new improved methodology. As a result, we stated the following problem definition that was used as the foundation for this thesis:

Develop a set of accessible and reproducible process mining best practices, based on both prior research and expert knowledge, which lead to results that are similar or better than the original research.

The solution design of this thesis is based on several steps. First, a preceding literature review served as a starting point for the identification of problems in process mining research in healthcare. The cases that were identified during these stages were subsequently used for a detailed multiple case study in this thesis. The goal of the case study was to investigate the current state of process mining methodologies in healthcare and for six cases such a methodology was constructed. With the addition of process mining experts' knowledge (i.e. developers of the process mining software) that was gathered

through a series of interviews, a “global” process mining methodology was synthesized using both the experts’ knowledge and the results of the case studies.

A search for methods to report the new methodology and best practices led to the selection of pattern language as a solution (a pattern is a short and reusable solution to a specific but common problem). The “global” process mining methodology of present process mining research served as the basis for the development of the new process mining pattern language. The different steps in the methodology are represented by individual patterns. These patterns have been developed on the same datasets that have been used during previous process mining research, which allowed us to compare the results of the current and new methodologies. As a result, a set of 22 process mining patterns and a pattern network (i.e. a directed network of all patterns) was created, dealing with the complete process mining methodology ranging from the data collection stages to actual mining stages with the algorithms in the ProM framework (i.e. specialized software that contains numerous process mining algorithms).

During the development of the patterns, it became apparent that there were several differences between the alternative patterns for a number of research steps. As a result, we selected the patterns with the highest potential for accessibility and improved results as candidate best practices. A set of best practices would increase accessibility as it allows researchers to focus only on the most important patterns. By testing the set of candidates on three datasets we confirmed that the applying the set of best practices of process mining indeed leads to results that exceed the quality of the results that are obtained by process mining without a similar approach. In addition, a virtual machine i.e. a completely isolated guest system platform that supports the execution of a certain operating system and software) was created to allow reviewers to test the process mining patterns themselves.

In addition, to test the patterns and to increase reproducibility, a set of virtual machines (were created in the SHARE environment, in which we aimed to replicate the results of previous process mining studies with the newly developed patterns. As a result, we were able to confirm that the patterns indeed can create results that are similar to the original process mining methodologies.

Furthermore, as a means to increase accessibility and reproducibility, we introduced the concept of screencasts (i.e. digital video recording of a computer screen output) to communicate process mining research. Screencasts allow for a simple and quick overview of the methodology and produced results. To communicate the pattern language and best practices, screencasts have been created that illustrate the use of several patterns. In addition, screencasts have also been used during the replication of process mining results. This approach shows the potential of the combination of virtual machines, pattern language and screencasts as a way to report process mining methodology and accompanying results.

We have created a dedicated website[1] which serves as a centralized repository for all process mining related knowledge. Currently, it contains the complete set of patterns, best practices, screencasts and virtual machines.

[1] Refer to http://sites.google.com/site/prompatternlanguage/home (retrieved 8 August 2011).

To increase the contribution of the patterns, we aimed to validate the results of the patterns and best practices in two case studies where the involvement of process mining experts secured the validity of the results. First, we aimed to reproduce the results of previous process mining researchers with the use of the newly developed patterns. Second, we applied the best practices of process mining in order to achieve process models that would exceed the quality of the original research, thereby demonstrating the added value of the patterns and best practices. For both cases we found that the patterns are indeed able to create results that are similar to present process mining methodologies. In addition, the second case showed that the best practices lead to a significant increase in usability of the process models, thereby strengthening their added value for process mining research in healthcare. Moreover, the process experts recognized the need for an accessible methodology, strengthening our initial claims.

As a result, this thesis contributes to process mining research in several ways. First, the process mining patterns that have been developed could increase the accessibility and understandability of process mining in two ways: 1) the patterns are a collection of the most important knowledge for each step in the process analysis and 2) the pattern network and best practices serve as a guide through the complex and extended maze of process mining algorithms and decisions. By creating a central repository of process mining patterns, it should be easy for any researcher to quickly identify the possibilities that should be considered. Secondly, the best practices of process mining should allow non-experienced process mining users to quickly analyze their data through the pre-specified use of a series of patterns. Compared to previous process mining guides, this thesis presents a much more detailed, complete and accessible methodology. Third, to increase the reproducibility of process mining research one can simply refer to the patterns that were used during the process analysis. By only referring to the patterns and explaining certain deviations and choices, one could increase the ability for readers to understand and reproduce alleged results. Fourth, in this thesis we have explored the use of virtual machines and screencasts as a means of reporting process mining results. A combination of these methods proved to be a potentially powerful combination that should be considered by future process mining researchers.

In addition, several limitations of this research have to be stated and they could serve as directions for future research. First, the most important limitation is the validity of the claims in this thesis. Despite our efforts to increase accessibility of process mining research and the genuine interest of experts in the field of healthcare, it has yet to be proven that the patterns indeed lead to increased understanding of process mining for non-experts. Especially, it should be confirmed that non-experts are able to use the best practices to create results that are of superior quality compared to process models created with conventional mining methodologies. Furthermore, the claimed reproducibility of the process mining patterns has yet to be validated. It needs to be confirmed that the patterns indeed allow researchers to present their methodology in a more convenient way and whether this allows reviewers to easily understand the research. In addition, the added value of virtual machines and screencasts need to be confirmed as well. Secondly, the current repository of process mining patterns is limited. There are still dozens of unexplored process mining algorithms and possibilities. Therefore, future research could extend and improve the current pattern language. Lastly, since process mining is not nearly limited to healthcare, future research should focus on the external validity of the pattern language and test the methodology in industry settings other than healthcare.

Table of contents

List of figures

List of tables

1. Introduction

As the global need for healthcare keeps rising there is a continuous pressure for lower waiting times, lower costs, increased throughput and increased overall performance; now more than ever is there a need for methodologies to make the complex healthcare processes more efficient. However, a potential and promising method for the discovery and extension of business processes, called process mining [Aalst *et al.*, 2007a], has yet to gain solid ground in healthcare. The goal of process mining is to give the process owners more insight into the actual events that occur in the real process in the form of a process model and present the performance of the corresponding process and its resources. During preliminary research [Janssen, 2010 & 2011][2], the use of process mining in healthcare has been investigated and this led to the identification of numerous problems that obstructed process mining from reaching its full potential in healthcare. As a result, this master's thesis was conducted to analyze and solve the current problems of process mining in a healthcare environment.

This chapter will serve as an introduction to the present problems for process mining in healthcare and will elaborate on the according problem definition that has served as the starting point for this thesis.

1.1 Present state of process mining research in healthcare

Process mining in healthcare has been studied by several researchers and graduate students [Gupta, 2007; Mans *et al*., 2008 & 2008; Riemers, 2009; Ramos, 2009; Zanden, 2010]. However, most research can be classified as pilot studies only, executed by researchers and students from the Eindhoven University of Technology, and a detailed and clear description of the decisions, settings and possibilities of the process mining methodology is lacking. As a result, during preceding stages of this project [Janssen, 2010 & 2011] several problems with regard to the current process mining research in healthcare have been identified.

For one, new researchers may be misled by the fact that the process mining results of prior research is only of limited use and produces process models with a low fitness measure. The cause of this problem may be partly due to the fact that default parameter settings have been used in most studies and powerful data pre-processing steps have not been considered.

Secondly, the accessibility of process mining is low, due to the overload in literature, compared to the descriptions on how to use the complex algorithms. In addition, few studies are available on the important pre-processing steps of the datasets and the limited information that is available is scattered throughout the internet. Therefore, for present and especially new users of process mining it can be a daunting task to gather a complete overview of the many process mining possibilities and even more difficult to decide what algorithms are important and how to use them. As a result, process mining may remain a niche research area, limited to and understood by only business process management specialists, or even worse, the closed circle of process mining developers around Eindhoven University of Technology and a few of its students.

[2] Refer to http://sites.google.com/site/prompatternlanguage/ (retrieved 8 August 2011) for the documents of the preceding studies.

Thirdly, as the case studies in the preceding research show, even experienced ProM users at the Eindhoven University of Technology can get lost in the large and still increasing number of both mining algorithms and pre-processing options, thereby focussing too much on what they already know, excluding the possibilities of other useful algorithms and the strength of a combining them.

Fourthly, the reproducibility of the current process mining research is low as a result of the poorly documented process mining methodologies. Subsequently, this decreases the possibility for future researchers to verify the alleged results. Even worse, it also has a negative effect on the possibility for future researchers to apply previously developed methodologies in their process mining research. As a result, researchers may get lost in an infinite loop and re-inventing the wheel for process mining over and over, excluding interesting new possibilities.

1.2 Problem definition

As can be learned from section 1.1, there are numerous problems for process mining in healthcare causing disappointing results and inhibiting process mining from reaching its full potential. In short, these problems can be translated into two focal areas: 1) the *accessibility* of process mining is low due to an information overload in scientific literature and information deficiency with regard to process mining methodology and 2) the *reproducibility* of current process mining research is low due to poorly documented methodologies that are used by researchers.

As a means to overcome the previously stated problems, we considered that several researchers have tested the use of process mining in a healthcare setting. However, very few have taken much of the available knowledge on process mining methodologies into account. Most researchers have based their research methodology on one or two preceding studies only. Therefore, we decided to consider a broader view for development of a new process mining methodology, taking into account multiple studies that were available on this topic so far, to obtain a more complete overview of the process mining possibilities in healthcare. In addition, as means to overcome the pitfalls of information overload and tunnel vision, the addition of state-of-the-art process mining knowledge by several process mining experts was considered. By combining this knowledge we aimed to create a new methodology that would lead to results that exceeded the present available process mining methodologies. To increase the accessibility and reproducibility of the research and the methodology, we recognized the need for an alternative method of presenting the research, results and methodology. In addition, as another means to increase the added value of the methodology, development and testing would be performed on multiple datasets, whereas traditional process mining research is limited to one or two datasets at most, thereby increasing the validity of the results.

However, considering the large number of available algorithms and pre-processing steps, a new complete and detailed process mining methodology would still cause information overload. Therefore, as a means to increase accessibility and omitting the need to go through a large amount of information, we decided to develop a set of best practices for process mining in healthcare, illustrating only the most important algorithms, pre-processing steps and subsequent choices and answers which are involved. As a result, current and new process mining users would have easy access to the methodology and could

easily apply it to their own research without having to spend many hours collecting information and establishing a sound methodology themselves.

In addition, one of the recognized problems in process mining research is the low reproducibility of its methodology and results, which is the consequence of the poorly documented process mining methodologies in current process mining research. This inhibits future researchers to use previously developed process mining methodologies in their own research, forcing them to spend much time on redundant tasks. As a result, another significant problem which needed to be addressed in the thesis project was the reproducibility of the newly developed process mining methodology. It was apparent that the communication of the methodology needs to be clear and concise, allowing future researchers to use the new methodology in any setting without the need to search for additional information.

We have to consider the fact that process mining is a new field of research which undergoes constant development and changes. Therefore, perhaps there have not been many opportunities for the development of a methodology as is proposed in this research. Now that the potential of process mining is recognized and the accompanying points for improvement have been identified, we have stated the following problem definition that served as the basis for this thesis project:

Develop a set of accessible and reproducible process mining best practices, based on both prior research and expert knowledge, which lead to results that are similar or better than the original research.

1.3 Thesis outline

This thesis aims to provide a full description of the systematic methodology that was used to create a solution for the problem definition. Therefore, much information is included on the development and validation of the best practices and the research that preceded it. **Chapter 2** details on the preliminary knowledge required for a proper understanding of the methodology and techniques that are used in this thesis and will introduce the reader into subjects as such business process management, process mining (in healthcare) and ProM. In **chapter 3**, the research methodology that was followed during the complete thesis project is presented. **Chapter 4** discusses the multiple case study and interviews, the foundation for the development of the new process mining methodology and best practices. In **chapter 5**, pattern language, virtual machines and screencasts are introduced as a method to report the new methodology. **Chapter 6** discusses the development of the process mining pattern language and accompanying best practices. In **chapter 7**, previous research results are reproduced to demonstrate the use of the patterns. **Chapter 8** presents the results of the validation of the best practices. Finally, **chapter 9** will provide the conclusions, limitations and recommendations for future research.

To avoid the pitfall of information overload for users outside the scientific community, such as healthcare quality managers and process managers, it is recommended to read only chapters 2 and 6 (more specifically sections 6.2 and 6.4) for a complete understanding of the best practices for process mining in healthcare. For future process mining researchers, chapter 5 should be considered as an addition since it discusses the options to increase the reproducibility of their research. The remaining chapters serve as a means to communicate the development and validation of the results of this thesis and should be of interest to advanced process mining researchers and developers.

2. Preliminaries

This chapter serves as a brief introduction into the fundamental principles that are used in this thesis and reading it is therefore especially recommended for readers not familiar with the following topics: business process management, process mining, ProM and process mining in healthcare. The background knowledge on these topics will greatly aid understanding of the remainder of this thesis. Additionally, this chapter increases the added value of our results by placing the subjects of this thesis into a business perspective.

2.1 Business process management

Business processes started receiving significant attention by major companies during the early 1990's. Davenport, who was one of the first persons to recognize business process reengineering and one of the foremost authors in this area, defines a business process as [Davenport, 1993]: "*a structured, measured set of activities designed to produce a specific output for a particular customer or market. It implies a strong emphasis on how work is done within an organization, in contrast to a product focus's emphasis on what. A process is thus a specific ordering of work activities across time and space, with a beginning and an end, and clearly defined inputs and outputs: a structure for action. ... Taking a process approach implies adopting the customer's point of view. Processes are the structure by which an organization does what is necessary to produce value for its customers."* The results of a business process can vary, for example it can take the form of tangible products such as salt, an intangible product (i.e. service) such as teaching, or something more in the middle of the tangibility spectrum such as fast-food outlets (which provides the service of preparing the food, i.e. the customer receives both service and a physical product) [Wilson *et al.*, 2008]. Either way, business processes are present for any company along the tangibility spectrum.

Business process management (BPM) was introduced by Aalst *et al.* [2003] as: "*a systematic, structured approach to analyze, improve, control, and manage processes with the aim of improving the quality of products and services."* Several empirical studies have investigated the importance of the effect of business process management and demonstrated that BPM has positive effects on organizational performance, esprit de corps and financial performance [McCormack, 2001; Kohlbacher, 2010]. To aid companies in their BPM, Business Process Management Systems (BPMS) were introduced and these are defined by Aalst *et al.* [2003] as: *"an information system, which is a generic software system, driven by explicit process designs to enact and manage operational business processes".*

Before one can start with the optimization of a business process, it is imperative to gain a complete understanding of the actual outlook of the process and its performance in the as-is situation. Therefore, to make the analysis and development of processes visual and more accessible, Aalst *et al*. [2003] introduced the business process management lifecycle. This lifecycle was extended by Aalst *et al*. [2007c] to the state that is depicted in figure 2.1.

In the BPM lifecycle, the *design phase* uses input from the diagnosis phase to identify points for improvement (which is, of course, only possible when one is interested in improving an existing process), such as bottlenecks, and the output is transferred to the configuration part of the lifecycle. The resulting process definition contains the following elements: the process structure, resource structure, allocation logic and interfaces. During the *configuration phase*, designers focus on the detailed specifications of the selected design and shift the emphasis from the performance of the process (design phase) to the realization of the corresponding system. In the *execution phase*, the configured workflow from the configuration phase becomes operational by transferring the process definition to the workflow engine. The execution phase is monitored by the *control phase*. In this phase, the BPMSs monitor individual cases to provide feedback on their progress in the process. Moreover, aggregation of individual case data enables to analyze the global process performance in terms of performance indicators such as throughput time. This information can both be used for the execution phase as well as the diagnosis phase. To illustrate, for the former case it can be useful to temporarily add additional resources to parts of the process where bottlenecks are identified. In the latter case, during the *diagnosis phase*, aggregated data is used to reveal weaknesses in the process and provide ideas for a redesign of the process. The diagnosis phase is also the main domain for process mining, which will be discussed in the next section.

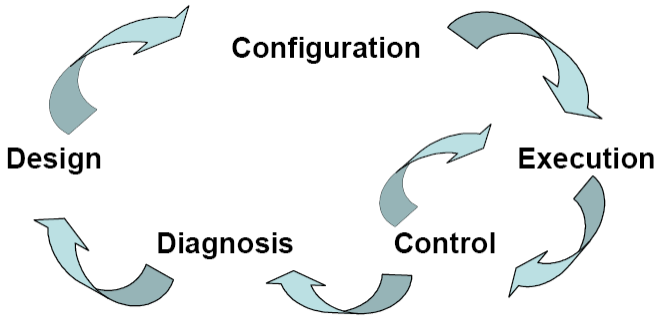


Figure 2.1: Business process management lifecycle [Aalst et al., 2007b].

2.2 Process mining

Process mining is a technique that is used mostly during the diagnosis phase of the BPM lifecycle, where weaknesses in the process are identified and ideas for redesign of the process are raised. The basic principle of process mining is *"to discover, monitor and improve real processes (i.e., not assumed processes) by extracting knowledge from event logs"* [Mans *et al.*, 2009]. In other words, process mining is the automated construction of process models that explain the observed behaviour in an event log that is recorded by an information system [Aalst & Weijters, 2005; Aalst *et al.*, 2007a]. Event logs are typically recorded by information systems such as ERP (Enterprise Resource Planning) and CRM (Customer Relationship Management) [Aalst *et al.*, 2007a; Aalst *et al.*, 2009], also known as process aware information systems [Mans *et al*., 2009]. These logs usually contain information on the different events that take place in a business environment. Each *event* refers to an *activity* (a well defined step in the process), each event refers to a *case* (i.e. process instance), each event refers to an *originator* (the person or department executing or initiating the activity), and finally each event can contain a *timestamp* which is used for the ordering of events [Aalst *et al.*, 2007a]. Using a variety of available mining algorithms (such as the such as the *α-algorithm* [Aalst *et al.*, 2004] and the *HeuristicsMiner* [Weijters *et al.*, 2006], available in the ProM framework[3] which is described in section 2.3), researchers

[3] Refer to http://www.processmining.org (retrieved 8 August 2011) for addition information on process mining research and the ProM framework.

can mine process models from the event log in different formats such as Petri nets (.cpn[4]) [Murata, 1989], HeuristicNets [Weijters *et al.*, 2006] or Event-Driven Process Chains (EPC) [Kindler, 2006].

In the field of process mining there are three perspectives which one can discriminate between: the process perspective, the organizational perspective and the case perspective [Aalst & Weijters, 2005; Aalst *et al.*, 2007a; Aalst *et al.*, 2009]. The *process perspective* focuses on the control flow, i.e. the ordering of activities. The goal is to find a process model which represents the actual process as good as possible (i.e. a high fit) and return this to the user in the form of for instance a Petri net. The *organizational perspective* focuses on the people (or any other resource) that execute the activities in the process, called the *originators*. Interesting for this perspective is what activities are executed by an originator, how these originators are classified in terms of roles and/or organizational units and how these originators are related to each other (e.g. handover of work). The *case perspective* focuses on the properties of cases in the event log. Each case has a certain path through the business process, undergoing certain activities that are executed by certain originators at a certain time. Additionally, the results for each perspective may refer to the logical and/or performance issues [Aalst & Weijters, 2005] (e.g. for the process perspective: *A* follows *B* and there is 35 minutes between the start of *A* and the end of *B* respectively).

Process mining can be used to investigate the three different perspectives of a business process. However, there is also variation in the use of the actual mining results [Aalst *et al.*, 2007c; Mans *et al.*, 2009; Rozinat, 2010]. For instance, when there is no a-priori process model present, we are dealing with the *discovery* type of process mining. In this method, a model is constructed based on an event log [Mans *et al.*, 2009; Aalst *et al.*, 2007c]. However, the discovery is not limited to process models only, as we have learned that also the organizational and the case perspective can be investigated. Additionally, *conformance* mining/checking deviates from the discovery type in that there is an a-priori model available. Mining results can be used to verify whether reality conforms to the a-priori model, i.e. is the process executed as the business believes it is and should be executed. Conformance checking can also be used to confirm that the mined model conforms to reality. Therefore, conformance checking works both ways. Finally, when a business is confident about the quality of a (discovered or existing) process model, *extension* techniques can be used to further develop the model by introducing additional information contained in the event log. To complete this introduction on process mining, figure 2.2 places process mining in a business environment.

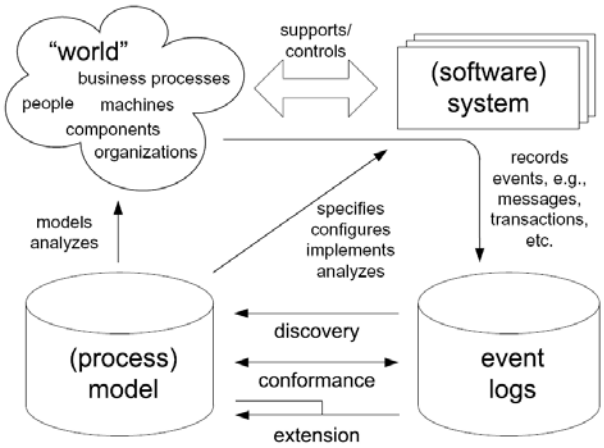


Figure 2.2: Overview showing three types of process mining: discovery, conformance and extension [Aalst et al., 2007c].

[4] Refer to http://cpntools.org/ (retrieved 8 August 2011) for additional information on Coloured Petri Nets and the CPN tools software for editing, simulating and analyzing Coloured Petri Nets.

As can be concluded from the previous description of process mining, there are several process mining types and perspectives that are interwoven. The results of process mining and the algorithms that are used to obtain them depend on the nature of the research. As an example, when one is interested in the discovery of a process model, different techniques should be used compared to when someone wants to check the performance of their originators. Regardless of one's interest in the event log, there is one common tool that combines all process mining possibilities, the ProM framework, which is discussed in the next section.

2.3 The ProM framework

In section 2.2 we discussed the wide variety of process mining perspectives and a subsequent large combination of process mining possibilities. This inherently results in a large number of different process mining algorithms, each dedicated to one or more of the process mining perspectives. The ProM framework is an open source process mining tool developed at the Eindhoven University of Technology, and combines a large number of different algorithms into a single pluggable environment [Aalst *et al.* 2007c; Aalst *et al*., 2009; Alves de Medeiros & Weijters, 2009]. In ProM, each independent algorithm can be called through the use of its corresponding plug-in. There are several types of plug-ins, each dedicated to a certain goal such as mining a process model or analyzing process performance.

The *mining* plug-ins use a mathematical algorithm to mine a process model from an event log [Aalst *et al.* 2007c]. For instance, the *HeuristicsMiner* (HM) or *α-algorithm* can be used to investigate the process perspective, as the result of these plug-ins is a process model. The HM produces a process model in a HeuristicsNet format and the *α*-algorithm produces a Petri net. There is a large variety of different plug-ins that can be used for process mining, each producing a process model in a certain format. Additionally, some mining plug-ins do not produce a process model. To illustrate, the *Social network miner* [Aalst, Reijers & Song, 2005; Song & Aalst, 2007a] can be used to investigate the organizational perspective, as it mines the relationships between the originators in the event log.

Analysis plug-ins can be used for the analysis of the underlying process model [Aalst *et al.*, 2007c]. For instance, the sojourn time, resource utilization and waiting times for an activity can be analyzed and monitored. Examples of plug-ins are the *Basic Performance Analysis* and *Dotted Chart Analysis* [Song & Aalst, 2007b]. Moreover, some analysis plug-ins can be used to cluster process instances based on similar process characteristics. Clustering results in smaller portions of the event log that contain process instances that are more similar to each other than compared to the remainder of the event log. Subsequently, these independent clusters can be used for additional analysis or process mining. *Sequence Clustering* [Veiga & Ferreira, 2009] and *Trace Clustering* [Song *et al*. 2009] are two of several algorithms that can be used for clustering data. Finally, some analysis plug-ins can be used to judge the produced process models by providing the so called fitness measure. In short, this is a representation of how well the process model fits the event log on which it was created [Gupta, 2007]. For instance, the ExtraBehaviourPunishment (EBP) fitness measures both the completeness and preciseness of the process model [Alves de Medeiros, 2006].

Common in any software tool is the ability to export the results. In ProM this can be achieved with the *export* plug-ins, which allow the user to save their process models or event logs. These files can again be imported by using the *import* plug-ins. In addition, the *conversion* plug-ins allow transformation of process models from format to another (e.g. from EPC to Petri net) [Aalst *et al.*, 2007c].

Furthermore, the ProM framework also possesses some powerful and important *filtering* possibilities and options to change the data in the event log [Mans *et al*., 2009] and many more plug-ins with different purposes are available [Aalst *et al.*, 2007c].

The ProM framework uses a standard Extensible Markup Language (XML) format, namely the Mining XML format (.MXML). Data that is used for process mining may originate from any business information system but has to be transformed to the .MXML format before it can be loaded into ProM [Aalst *et al.*, 2007a; Dongen & Aalst, 2005]. The ProM*import*[5] tool can be used to transform a variety of data formats into the .MXML format to be used in ProM [Aalst *et al.* 2007c]. The ProM*import* tool is a standalone tool which is not implemented in the actual ProM framework. Since 2010, it is also possible to create event logs using the commercial software tool *Nitro*[6]. In any case, an event log (figure 2.3) has a certain layout and several important elements that are required for process mining.

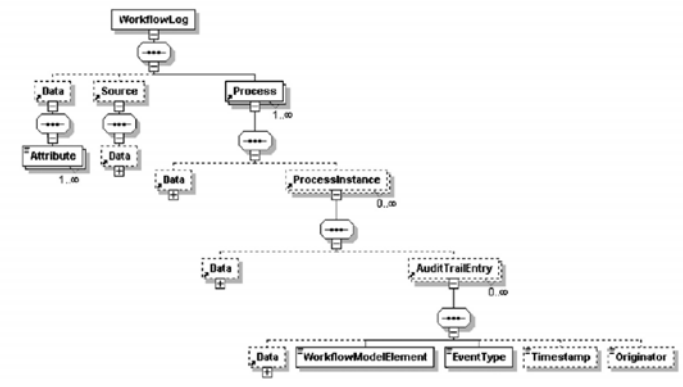


Figure 2.3: The MXML format that is used for process mining (XML schema) [Dongen & Aalst, 2005].

case id	activity id	originator	time stamp
case 1	activity A	John	9-3-2004:15.01
case 2	activity A	John	9-3-2004:15.12
case 3	activity A	Sue	9-3-2004:16.03
case 3	activity B	Carol	9-3-2004:16.07
case 1	activity B	Mike	9-3-2004:18.25
case 1	activity C	John	10-3-2004:9.23
case 2	activity C	Mike	10-3-2004:10.34
case 4	activity A	Sue	10-3-2004:10.35
case 2	activity B	John	10-3-2004:12.34
case 2	activity D	Pete	10-3-2004:12.50
case 5	activity A	Sue	10-3-2004:13.05
case 4	activity C	Carol	11-3-2004:10.12
case 1	activity D	Pete	11-3-2004:10.14
case 3	activity C	Sue	11-3-2004:10.44
case 3	activity D	Pete	11-3-2004:11.03
case 4	activity B	Sue	11-3-2004:11.18
case 5	activity E	Clare	11-3-2004:12.22
case 5	activity D	Clare	11-3-2004:14.34
case 4	activity D	Pete	11-3-2004:15.56

Figure 2.4: An example of an event log [Aalst & Weijters, 2005].

As we learn from figure 2.3, a workflow log has a specific tree structure. First, information on the software and/or information system that was used to record the activities/events in a process is specified in the *Source* element. The *Process* element represents one process and contains multiple *Process Instances* (PI) or cases. Each PI may hold multiple *AuditTrailEntries* (ATE) which refer to an event. Each ATE at least contains a *WorkflowElement* and *Event-type*. The former refers to an actual activity, a sub process or other routing element in the real process. The latter can be used to specify the type of event such as start or complete. Moreover, the ATE can contain a *Timestamp* and *Originator* elements. The timestamp is used to record the time at which an activity was executed and the originator element is used to indicate who (e.g. a person or department) has executed the activity. Finally, the *Data*

[5] Refer to http://www.promtools.org/promimport/ (retrieved 8 August 2011) for additional information on the ProM*import* tool.
[6] Refer to http://www.fluxicon.com/nitro/ (retrieved 8 August 2011) for additional information on the Nitro tool.

element can be used to specify additional information (e.g. when the case refers to a person this could be the age or gender) on all levels of elements in the .MXML format. All ATEs combined result in an event log such as can be found in figure 2.4. During this thesis it is important to consider that when we refer to an activity or event we imply the WorkflowElement (i.e. *activity id* in figure 2.4). In addition, when we refer to event classes we imply the collection of different events. To illustrate, in the event log in figure 2.4 there are four event classes, namely activity *A*, *B*, *C* and *D*. However, there are 19 events/activities in the event log, all representing one of the four event classes.

In addition to the many process mining possibilities that are offered in ProM, there are several so called "pre-processing" steps that need to be executed outside the confined algorithmic environment of the ProM framework in spreadsheets such as MS Excel [Ramos, 2009] or database managers such as MS Access [Mans *et al.*, 2008]. The goal of pre-processing is to prepare the data for the transformation into the .MXML format and analysis in ProM. These steps rely more on the intuition and process knowledge of the researcher instead of pre-developed algorithms. Unfortunately, despite the importance and potential of data pre-processing [Mans *et al.*, 2009], the focus of traditional process mining literature is mostly on the mining algorithms in ProM and does present much knowledge on the wide variety of pre-processing steps and possibilities.

2.4 Process mining in healthcare

According to Lenz & Reichert [2007], healthcare is characterized by its patient specific and frequent changing processes that require the cooperation of different organizational units and medical disciplines. Due to increasing medical costs and the competitive healthcare industry, healthcare institutions need to streamline their processes in order to increase process quality and decrease process cost [Mans *et al.*, 2009]. Therefore, Lenz & Reichert [2007] elaborate on the use of IT-systems and the importance of process awareness in healthcare. According to several researchers [Kohlbacher, 2010; Mans *et al.*, 2009; Lenz & Reichert, 2007], IT systems and process awareness yield a high potential for decreased costs and improved healthcare delivery. Additionally, these systems can record events in the form of an event log. Process mining software can subsequently use such data to analyze the process' performance and visualize the process in the form of process models. However, despite the potential of process mining and analysis it is still not widely applied in practice in the field of healthcare and its research has enjoyed little scientific attention. As a result, BPM in healthcare has not yet reached its full potential.

Despite the immensity of the healthcare industry, little scientific research has been devoted to the management of the BPM lifecycle through the means of process mining, regardless of its potential. During recent years however, more attention has been paid to process mining in the field of healthcare [Gupta, 2007; Mans *et al.*, 2008 & 2009; Riemers, 2009; Ramos, 2009; Zanden, 2010]. However, these studies can be classified as pilot studies only and have so far not generated huge response from the healthcare domain (potentially due to the disappointing results). Riemers [2009], Ramos [2009] and Zanden [2010] have started the development of a methodology for process analysis in healthcare. However, these studies have only contributed limited knowledge on process mining and a detailed description of their process mining methodology is lacking, leaving numerous gaps to be guessed by the

readers of their reports. Despite the best efforts of the previous process mining researchers, the discoveries that were made during the preceding stages of this thesis [Janssen, 2010 & 2011] show that there are several opportunities for improvement. To illustrate, up till now researchers have only applied a very limited number of process mining algorithms and pre-processing steps. Thereby, for instance, failing to recognize the potential of combining them. Furthermore, documentation of process mining research in healthcare is limited, which results in low reproducibility of the mining methodology and results. A combination of the problems indicates that there is a lack of a widely applicable, accessible and reproducible mining methodology.

To clarify, business processes in healthcare can be divided into clinical guidelines and clinical pathways [Lenz & Reichert, 2007]. The clinical guidelines contain domain specific knowledge, agreed upon by medical experts throughout the field. Clinical pathways are the site-specific adaptations of the clinical guidelines and contain more planning (e.g. resources and organizational issues) and detail. When referred to processes in healthcare in the context of this thesis, one can assume the clinical pathways are implied, as these are the actual processes that are logged by the hospital information system.

As we have learned from preceding process mining studies in healthcare, Diagnose Behandel Combinatie (Diagnose Treatment Combination) or simply DBC information is an important data object for process mining research in Dutch healthcare. According to DBC-onderhoud[7] (the organisation that is responsible for maintaining the DBC-system), a DBC can be defined as "*a predefined average packet of care (treatment) with a (sic) in most cases fixed price, which is applied when a specific diagnosis occurs.*" DBC-codes can therefore be used to indicate a patient's diagnosis and treatments. The form of a DBC in healthcare data is a numerical code that consists of four (or occasionally five) parts. For instance the code 5.11..1701.223 [Swolfs, 2010] refers to the *specialism* orthopaedics (5), the *care type* regular care (11), the *diagnosis* hip arthritis (317) and the *treatment* operation with clinical episodes. As will become obvious throughout this thesis, these codes can be used as input for the clustering of patients (e.g. clustering based on patients with a similar treatment DBC-code).

[7] Refer to http://www.dbconderhoud.nl/# (retrieved 8 August 2011) for additional information on the DBC-system in Dutch healthcare.

3. Research methodology

The goal of this thesis was to find a solution design for the problem definition that was stated during the preliminary research (chapter 1). Accordingly, this thesis was divided into several smaller individual parts that considered the different research steps that were required to accomplish this goal. The resulting research methodology that was followed during the thesis, including the preceding stages, is visualized in figure 3.1. In this chapter, the research methodology is described according to the different research steps that have been performed during this thesis project.

3.1 Preceding research

3.1.1. Literature review (chapters 1 and 2)

Process mining had been studied in a wide range of business industries [Aalst *et al.*, 2007b] already and has produced some promising results. However, in healthcare especially, general consensus is that process mining results could still be improved [Mans *et al.*, 2009; Riemers, 2009; Ramos, 2009]. As a result, a literature review [Janssen, 2010] was conducted in order to identify the gaps in process mining research in healthcare. Several case studies [Gupta, 2007; Mans *et al.,* 2008 & 2009; Riemers, 2009; Ramos, 2009; Zanden, 2010, Yang & Hwang, 2006; Shan *et al.*, 2008] and general process mining literature was analyzed and this resulted in the problem identification: *process mining research in healthcare suffers from a lack of a common detailed research methodology, low reproducibility and improvable results*. Subsequently, this problem identification served as input for the research proposal.

3.1.2. Research proposal (chapters 1 and 2)

During the research proposal [Janssen, 2011] the problem identification was given more detail as the case studies and process mining literature were analyzed more carefully. The final problem definition had been defined as: *develop a set of accessible and reproducible process mining best practices, based on both prior research and expert knowledge, which lead to results that are similar or better than the original research*. To solve this problem the research had been divided into several smaller parts: 1) investigate current process mining methodologies using a multiple case study and expert interviews, 2) find a method to report the best practices, 3) develop best practices and 4) validate the best practices.

A research plan was drafted to serve as a guide through the different research steps that had been defined. As a means to increase the quality of the proposed research project, several research quality criteria were identified and used during the development of the project plan. Aken *et al.* [2007] and Yin [2009] describe four types of research oriented criteria which a sound research project should meet:

1. *Construct validity* refers to the establishment of correct operational measures for the concepts that are being studied. For instance the use of multiple sources of evidence is considered as a good method to increase the construct validity. Therefore, we have introduced a multiple case study and several expert interviews as the sources of information for the new methodology.
2. The validity of the causal inferences that are made during scientific research is referred to as *internal validity*. To increase the internal validity of this thesis, multiple datasets were used to develop and test the new methodology.

3. As a means to increase the *external validity*, it is recommended to test the research results in a different setting (aiming to replicate the results). We have tested our solution in two new case studies and involved the knowledge of process experts to secure validity of the results.
4. Finally, *reproducibility or reliability* refers to the fact that results of the research can be replicated by a future researcher (i.e. the *same* case), leading to the same results as the original research. Therefore, clear documentation and explanation of the methodology and decisions (i.e. protocol) is of great importance. As was already discussed in chapter 1, reproducibility is a major problem for previous process mining research. Therefore, we aimed for a more detailed and complete documentation and developed additional tools for increased reproducibility.

Eventually, six cases were selected for the multiple case study [Gupta, 2007; Mans *et al.,* 2008 & 2009; Riemers, 2009; Ramos, 2009; Zanden, 2010] based on a set of selection criteria: 1) key subject headings include *process mining + ProM + healthcare*, 2) availability of data and 3) replication logic. Replication logic refers to the principle that either a case is selected because it produces similar results (i.e. literal replication), or it produces contrasting results but for predictable reasons (i.e. theoretical replication). The multiple case study was an iterative process and overlapped between the preceding stages and the actual thesis project.

To the best of our knowledge, an approach as was proposed during this stage is unique for process mining research in healthcare. Most studies only consider very few different process mining methodologies for their solution design and only one or two datasets to test their solution. By using multiple sources of information our goal was to create a more complete overview of the process mining possibilities in healthcare. In addition, by including multiple datasets during the development and testing of the best practices we aimed to create a more generalizable process mining methodology. With these factors combined we aimed for a distinctive study in a field that struggles to advance.

The products of this research proposal included: 1) a problem definition, 2) a sound project plan that included a description of the different research steps, time schedules, requirements and deliverables and 3) a definite case study selection, including a short description of their process mining methodology which would serve as the basis for the development of the new methodology.

3.2 Thesis project

3.2.3. Multiple case study (chapter 4)

The products of the research proposal served as the input for the start of the final thesis project. The first step was to get a detailed overview of the current process mining methodologies in healthcare. When there is only a limited amount of data available, a qualitative multiple case study is an excellent method to analyse current research results [Yin, 2009]. Therefore, the multiple case study method that is described by Yin [2009] was used a guideline for the analysis of the current process mining research.

First, the development of a theoretical framework is the basis of the multiple case study design and this was performed during the preceding stages of this thesis (chapters 1 and 2). Second, the selection of case studies must be carefully considered and should follow the so called *replication logic*. Third, a well

established data collection protocol is a major way to increase reliability/reproducibility of the research [Yin, 2009]. Therefore, before the analysis of the cases is started it should be clearly stated what, how and why something is investigated. For this research, the data collection protocol focused solely on the process mining methodology of previous research. According to Yin [2009], the selection of the cases and creation of the data collection protocol is an iterative process which started during the preceding stages of this thesis and was finalized during the thesis itself. During the so called pilot studies (first iterations of the analysis) both the selection criteria and the data collection protocol were established, two tasks that need to be completed before the actual multiple case study [Yin, 2009]. During the final analysis, no comparison between the cases is made yet, as they should all be judged according to the data collection protocol only.

After each case was analyzed an individual report was written. The six reports have been compared to each other during the cross case analysis. The goal of this phase was to identify the most important process mining steps of previous process mining studies in healthcare. This result would be used for the synthesis of a "global" process mining methodology during future steps in this thesis.

3.2.4. Expert interviews (chapter 4)

In addition to process mining literature and the results of the multiple case study, interviews have been conducted with process mining experts to gain additional knowledge and perspective on process mining and the ProM software tool. This group of experts consists of researchers that have contributed to the ProM software as developers (of plug-ins), published process mining research in scientific journals and conducted doctoral dissertations on process mining. According to Cassell & Symon [1994], a qualitative interview is a good method to understand the topic from the perspective of the interviewee and to understand how and why he/she has a certain perspective. For this research, a semi-structured approach was used as it uses a pre-determined set of themes and questions but allows deviating from these questions to focus on the most interesting parts of the interview [Saunders *et al.*, 2000]. During the interviews, specific questions with regard to process mining in healthcare were asked and theoretical problems were presented to analyze the experts' problem solving strategy. An additional advantage of interviews over literature is the possibility for the experts to combine process mining techniques and algorithms, whereas traditional literature on ProM plug-ins tends to focus on one single process mining step at a time, disregarding possible interaction between different plug-ins. The interviews allowed us to identify methodologies beyond the written literature on the use of the ProM plug-ins and possibilities. Subsequently, the result of this step was the recommendation of several important process mining steps by process mining experts.

3.2.5. Synthesis of process mining methodologies (chapter 4)

The synthesis of both the results of the multiple case study as well as the results of the expert interviews would serve as the basis for the development of the new process mining methodology. First, the most common and important steps were identified. Subsequently, these results have been synthesized into a single new "global" process mining methodology of the current process mining research. Visualization of this methodology allowed us to establish a global order of the many different steps.

3.2.6. Find method to report best practices (chapter 5)

One of the challenges of this thesis was to find a method to report the best practices. Such a method would have to meet several criteria: 1) the method should be easy to understand and accessible and 2) it should increase the reproducibility of process mining research. Eventually, we identified the concept of pattern language [Alexander *et al*., 1977] as an excellent method to report the new methodology because of its simplicity and reproducibility, as well as the possibility to create a pattern network that highlights the interdependencies between different the patterns. Using examples from several fields of research we developed a set of elements that are important for process mining patterns. The resulting process mining pattern language could serve to significant increase process mining accessibility and reproducibility. To facilitate the pattern network, a dedicated website was developed that allows interactive browsing through the patterns. However, to create additional value, two complementary methods were used.

First, virtual machines (VM) [Dittner & Rule, 2007] were identified as a potential method to increase reproducibility as the VMs facilitate the hosting of process mining software and data that was used during the original research. This allows peers to reproduce the alleged results without having to download and install additional software and data and provides the exact same research environment as was available to the original researcher. In this research, VMs were used to facilitate the testing of the process mining patterns and to serve as a proof of concept of process mining reproducibility.

Second, screencasts could serve as additional value for the presentation of the process mining patterns. Furthermore, it allows researchers to capture their analysis and peers to review their research by watching a simple video. In this thesis, screencasts have been used for both purposes.

3.2.7. Develop process mining patterns and best practices (chapter 6)

The “global” process mining methodology that was synthesized in chapter 4 served as the input for the development of the process mining patterns. With the addition of the knowledge that we have gathered throughout this thesis, a set of 22 process mining patterns was developed on three datasets. All patterns were created according to the pattern template that was established in chapter 5. Accordingly, the patterns were placed in a pattern network that visualizes their interdependencies. In addition, to deal with the alternative software tools for certain patterns we introduced the concept of hierarchy.

During the development of the complete set of patterns, some patterns proved to be simpler in their use and produced better results. Therefore, these patterns had been labelled as best practice. During the development we have kept record of the effect of the best practices on the resulting process model. As a result, we have showed that applying the sequence of best practices indeed leads to process models that exceed the quality of process models that are mined on raw or partially pre-processed event logs.

To facilitate testing of the process mining patterns and best practices we have created a virtual machine with the required software, datasets and the complete set of patterns. In addition, to increase the understandability of the patterns we have created several screencasts that show their application. As a method to increase accessibility we have developed a dedicated website that hosts both the complete

set of patterns, the best practices, links to the screencasts and the virtual machine and some additional information on the thesis project.

3.2.8. Reproducing research in VM with patterns (chapter 7)

By reproducing previous process mining results with the newly developed pattern language in a virtual machine we aimed at a twofold purpose. First, achieving similar or better results with the patterns, compared to conventional process mining methodologies, would strengthen our claim for the pattern language. Second, by creating VMs with screencasts and the pattern language we facilitate the replication of the work that has been performed in this thesis. This allows other researchers to be introduced to the patterns in an interactive way. Moreover, researchers can experience the combination of VMs, screencasts and patterns as a way to communicate process mining research.

3.2.9. Validate best practices (chapter 8)

During the previous research steps, process models have been judged by their layout and fitness measures. However, these measures do not state anything on their usability. Indeed, to judge the usability of the process models, some knowledge on the actual process is required. Therefore, in order to test the validity of the pattern language and best practices we have chosen to perform two additional case studies. During these case studies, involvement of process experts from the hospitals that donated their datasets ensured the necessary knowledge to validate the produced process models. These cases have been based on previous research, involving the same process experts, allowing them to compare the results of the original researcher to the results of the pattern language. Therefore, this stage had a twofold goal: 1) to compare the results of the original researcher to the reproduced results with the newly developed pattern language methodology and 2) to compare the process models created with the best practices to the models created by the original researchers. In the first case, we aimed to prove that it is possible to achieve very similar results or better compared to the original research. In the second case, we aimed that the newly developed pattern language would allow us to create process models which quality exceeds the quality of the process models by the original researchers. A positive validation of the former process models by healthcare process experts would prove that the newly developed patterns indeed create additional value for process mining research in healthcare.

3.2.10. Conclusions/limitations/recommendations (chapter 9)

As for any good scientific research, evaluation of the research, methodology and results is an absolute prerequisite [Aken *et al.*, 2007]. Therefore, we have stated a number of conclusions on the reproducibility, accessibility and improved results of the process mining patterns. In addition, it is important to recognize the limitations of the thesis with regard to these issues. Finally, recommendations for future research are specified that could possibly improve the results of this thesis, resolve the limitations or add to the validity and reliability of the results.

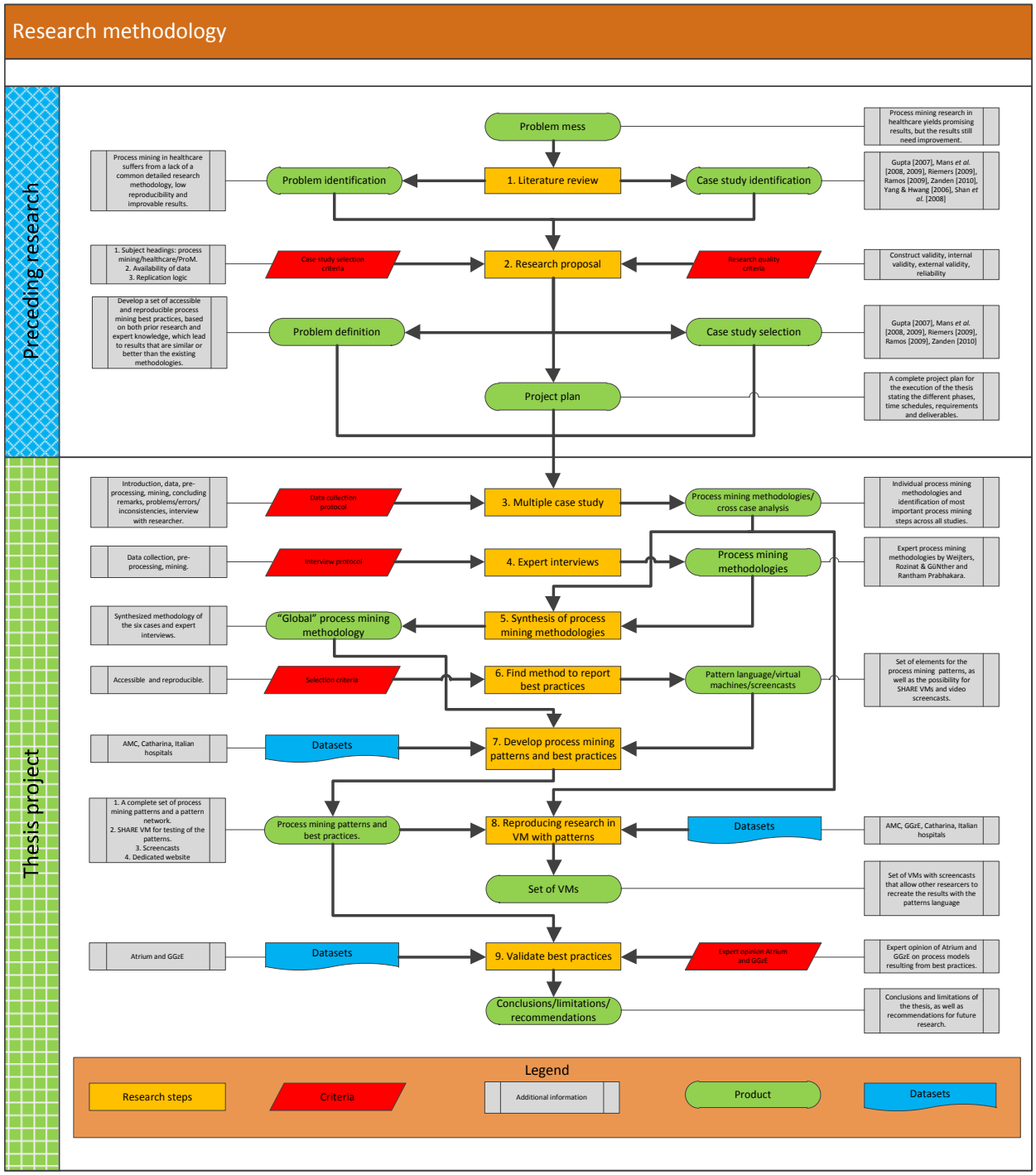


Figure 3.1: Visualization of the research methodology that was followed during this thesis project and its preceding stages. The orange squares represent research steps. The green ellipses symbolize the products of the research steps and the red parallelograms indicate certain criteria for a research step. Furthermore, the blue shapes represent the datasets that were used during this thesis project and the grey squares provide additional information on the products of the research for increased understandability of the methodology.

4. Multiple case study

In chapter 3 we discussed the possibility to learn from previous process mining studies and use them as a foundation for the development of a new process mining methodology. From literature [Yin, 2009], we learned that a multiple case study design is an excellent method to achieve that goal as it can be used to identify the present process mining methodologies, their results, associated problems and shortcomings. In addition, interviews with process mining experts were conducted to gather information on the most state-of-the-art process mining methodologies. According to Yin [2009], interviews as such are an excellent additional source of information in case study research.

From section 3.4 and figure 3.2 we learn that the first step in the multiple case study is to develop a theoretical framework. Preceding research [Janssen, 2010 & 2011] has already identified the current theory and stated the resulting problem definition. Furthermore, the theoretical background has already extensively been discussed in chapters 1 and 2. Therefore, the development of a theoretical framework is not repeated in this chapter. As a result, in section 4.1 we first discuss the selection of the case studies. Second, in section 4.2 we elaborate on the data collection protocol, which is an integral part of the analysis. Third, in section 4.3 the process mining methodology of the individual cases is discussed. Fourth, the cross case analysis is presented in section 4.4, which includes the evaluation, comparison and integration of the individual process mining methodologies. Fifth, the interviews with process mining experts and their results will be discussed in section 4.5. Finally, in section 4.6 we present the synthesized "global" process mining methodology of present process mining research in healthcare.

4.1 Case study selection

For the initial identification of candidate case studies, a search for papers was performed in several large scientific repositories such as Google Scholar, TU/e library and ISI Web of Knowledge. Key subject headings included *process mining*, *ProM* and *healthcare*. Furthermore, references in each candidate paper were checked for their usefulness as a case study for this thesis. As we are interested in the extension of process mining with ProM in a healthcare setting, it was important that the candidates' subjects included a combination of all three key headings. As a result, several studies that did not comply with the selection criteria had to be removed from the list of candidates. To illustrate, studies by Yang & Hwang [2006] and Shan *et al.* [2008] discuss the topic of process mining in a healthcare setting but they do not use the ProM framework and could therefore not be selected for the final multiple case study. Furthermore, research by Lassche [2010] was not selected because it did not provide significant new insights compared to the other studies (which according to Yin [2009] can also be considered a criterion for the selection of cases), as it only makes use of the method that was developed by Riemers [2009] and very briefly reports on the process mining methodology and results.

A second important criterion was the availability of the data that was used in the candidate case studies. These datasets would serve a threefold purpose: 1) to facilitate the development of the new process mining methodology, 2) to allow us to replicate the previous process mining research and 3) to validate the new methodology.

The availability of literature that met all selection criteria was very limited. However, during the period of October 2010 through February 2011 [Janssen 2010 & 2011], six studies originating from the Eindhoven University of Technology were labelled relevant for the final multiple case study: Gupta, 2007; Mans *et al.*, 2008; Mans *et al.*, 2009; Riemers, 2009; Ramos, 2009 and Zanden, 2010. There is a major difference between these six cases as the work by Mans *et al.* [2008 & 2009] was published in scientific journals and the other four cases are unpublished masters' theses. In addition, there is much correlation between the studies, and more specifically between the latter four. This is the result of successive execution of the studies, each using the methodologies that were developed in the preceding papers. To clarify, the work of Ramos and Zanden is mainly based on the method developed by Riemers, which in its turn was partially inspired by the preceding work of Mans *et al.* [2009]. However, each successive study did contribute to the process mining methodology and thereby qualifying for selection.

Finally, due to the fact that all studies find their origin at Eindhoven University of Technology, access to the datasets was considered possible and the data was secured during the preceding stages of this thesis. The six case studies have made use of the following datasets:

- Amsterdam Medisch Centrum (AMC): used by Mans *et al.* [2009] and Ramos [2009]
- Catharina Ziekenhuis Eindhoven: used by Gupta [2007]
- Italian hospitals in Lombardia: used by Gupta [2007] and Mans *et al.* [2008]
- Geestelijke Gezonheidszorg Eindhoven (GGzE): used by Zanden [2010]
- Atrium Medisch Centrum Parkstad Heerlen: used by Riemers [2009]

4.2 Data collection protocol

Constructing the data collection protocol is an important and iterative step in the multiple case study. According to Yin [2009], a well documented data collection protocol is *"…a major way of increasing reliability in case study research…*" The unstructured nature of some of the case studies and the differences between (writing) styles increased the difficulty of developing a general protocol. Furthermore, four of the selected cases are the result of a master's thesis project and consequently are much more detailed than the published work by *Mans et al.* [2008 & 2009] (as the former are allowed to use a considerable amount of space compared to published research).

Preceding research [Janssen, 2010 & 2011] served as a pilot to identify the variety of data, facts, methods and results that were reported case studies. After several iterations, it was possible to establish a solid data collection protocol that could be used during the final multiple case study. For the analysis in this thesis, we only focussed on the process mining part of the research and the steps leading towards it (i.e. data pre-processing). Additional use of software and methods are only mentioned and considered when they are of fundamental importance to the process mining methodology. Furthermore, we only focus on mining and clustering related plug-ins, not on performance analysis plug-ins. This was an explicit choice since actual mining algorithms are dependent on more parameters and considered more difficult. Additionally, during the preliminary analysis it became apparent that ProM was not the most suitable software for most performance analysis issues (as was concluded by Riemers [2009] and Ramos [2009]).

As was discussed before, during the analysis of the cases, no cross comparison shall be made and cases will only be reviewed according to the data collection protocol. For the sake of brevity we will only present the points for which data was collected and the reader is referred to Appendix A for the complete data collection protocol. The protocol that was used during the analysis consisted of the following elements: *Introduction into the research*, *information on the dataset*, *pre-processing methodology*, *process mining methodology*, *identified problems/inconsistencies/errors*, *concluding remarks* and *interview with the researcher*.

4.3 Case study reports

Due to the limited space that is available in this thesis and for the sake of brevity, the full analysis of the cases according to the data collection protocol is included in Appendix B. In addition, for each case we have created a process model that visualizes its complete process mining methodology, including both the pre-processing steps and the mining algorithms. These models are illustrated by figures C.1. to C.6 in Appendix C. The different steps in the process are used to indicate certain activities that the original researcher has performed. Examples include the collection of data, renaming of events and process mining with the HeuristicsMiner. Additionally, for some steps, important information has been added to the process model as well, such as what activities are renamed and aggregated. Together, this should provide the reader with an accessible and simple overview of the different methodologies that have been used in previous research.

4.4 Cross case analysis

The goal of the cross case analysis was to compare the process mining methodologies of the individual case studies and identify common and important steps. As we have learned, each researcher has used a different, but comparable methodology. For convenience we have divided the cross case analysis in a several parts:

- **Introduction into the research:** This briefly summarizes and compares the goals and results of the case studies.
- **Information on the data**: Highlights the most important aspects of the datasets that were used.
- **Pre-processing methodology:** This part focuses on all non-mining related data transformation.
- **Process mining methodology:** What mining algorithms have been used and how have they been used?
- **Identified problems/inconsistencies/errors:** What problems were encountered during the analysis of the reports?

4.4.1 Introduction into the research

The different case studies can clearly be categorized into two classes, published research and master students' thesis projects. The first category consists of two studies conducted by Mans *et al.* [2008 & 2009] and their research served as an exploratory research of the applicability of process mining in a healthcare setting. Due to the fact that there is little room for extensive details in published work (as was stated by Mans), the level of detail on the methodology and the results is low reducing the

reproducibility of both papers to a minimum. The second category includes the remaining four case studies [Gupta, 2007; Riemers, 2009; Ramos, 2009; Zanden, 2010] and they enjoyed more freedom and space for the description of details on the research methodology and results. Unfortunately, during the analysis we still identified a set of problems with regard to these topics that are common in all reports (refer to identified problems/inconsistencies/errors in section 4.4.5).

Three thesis reports (Riemers, Ramos and Zanden) are highly correlated with regard to their methodology. This is the result of the sequential execution of the individual research projects (in fact, all three students were mentored by the same professor at Eindhoven University of Technology). Riemers [2009] has adapted the CRISP-DM framework [Two Crows Corporation, 2005] for the application in a healthcare setting, changing various steps and adding the concept of process mining as a method to understand and visualize business processes[8]. Subsequently, this methodology was validated and extended by Ramos [2009] in a different healthcare setting and he introduced a few additional process mining techniques to the methodology. Finally, Zanden [2010] has tested the methodology in a mental healthcare setting. Despite the fact that these thesis reports serve as a guideline on how process analysis and improvement in a healthcare setting should be executed, the reproducibility is too low to be transferred to a different case without extensive additional research on the methodology. In addition, the process mining results have only generated moderate response from the healthcare field, which could partially be the result of the little use of the wide variety of different process mining algorithms and their corresponding parameter settings. In fact, little attention has been paid to any parameter settings in most studies and only very few algorithms have been considered all together, thereby losing the powerful potential of combining them. Furthermore, not all researchers have succeeded in completely profiting from the powerful potential of pre-processing.

Finally, the research by Gupta can be considered more isolated as it is not directly connected to the other cases. However, this case is characterized by the higher level of detail and well considered plug-ins and parameter settings. Despite its limitations, this report could serve as an example to other process mining research with regard to reporting the methodology.

4.4.2 Information on the data

Table 4.1 summarizes the characteristics of the datasets that have been used in the six case studies (for the sake of brevity, not all subsets are reported). As can be observed, some data characteristics show more variation than others. The number of process instances varies greatly, but can be reduced by clustering methods (such as can be observed for the AMC clusters). Furthermore, for the global event logs, the number of event classes is high. Healthcare datasets commonly contain a high number of event classes that are of no interest for the main behaviour in the process. Aggregation and renaming of events can reduce the number of event classes. An excellent example once again is the AMC case, going from 417 to 26 event classes with aggregation and renaming of events (i.e. AMC aggregated). Additionally, most datasets only contain information on the day an event has been executed and do not

[8] The focus of the framework that was developed by Riemers [2009] is on both process mining and visual analysis. As a result, the new process mining methodology that was developed in this thesis can be considered as an addition to and element of the process mining parts of the framework by Riemers.

carry a more detailed timestamp that indicates the exact time of execution. Furthermore, the number of originators is similar for most cases with the exception of the Catherina hospital. However, it should be noted that most datasets contain different levels of aggregation with regard to originators (e.g. there are more persons than departments). In addition, most datasets included DBC-code information, an important characteristics that can be (and has been) used for the clustering of process instances.

		Italy (treatments)	Italy (treatments & measurements)	AMC global	AMC aggregated	AMC aggregated M11	AMC aggregated M12	AMC aggregated M13	AMC aggregated M14	AMC aggregated M15	AMC aggregated M16	Atrium Mammacare	Atrium Bedreigde voet	GGzE (All_unique_activities level 3)	Catherina (Complications)	Catherina (ComplicationsTreatmentsExaminations)	Catherina (ComplicationsTreatmentsExaminationsMeasurements)	Catherina (Examinations)	Catherina (Treatments)
Dataset characteristics	DBC-codes	No	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	No	No	No	No	No
	# Hospitals	4	4	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
	# Process instances	369	374	682	682	134	8	296	93	12	139	3436	492	3512	576	2713	2765	2458	2711
	# Events	6.484	11.790	43.615	10.823	2.080	94	5.367	1.225	131	1.926	44.173	30.493	83.920	1.707	53.874	114.592	13.737	38.430
	# Event classes	95	102	417	26	17	14	24	17	10	19	216	192	61	183	615	619	177	255
	Timestamp	Day	Day	Day	Day	Day	Day	Day	Day	Day	Day	Day	Day	Day	Min	Min	Min	Min	Min
	# Originators	1	1	28	27	17	14	25	17	10	19	12	10	33	77	229	266	181	216
	# Event types	3	3	1	1	1	1	1	1	1	1	1	1	1	2	3	3	2	2

Table 4.1: Summary of the different datasets that have been used in the six case studies.

4.4.3 Pre-processing methodology

From most cases we have learned that pre-processing of the data is an integral part of process mining research. Creating the database, modify data and creating the event log are just a few of the pre-processing steps that we have identified. Some pre-processing steps only serve to make the eventual mining more convenient and increase the chance of understandable process models (e.g. clustering of data) and can therefore be classified as optional. Other pre-processing steps are essential and cannot be skipped (e.g. creating an event log). During the analysis of the pre-processing methodologies we have used the ETL-concept (extract, transform and load) to describe and aggregate the steps in the case studies.

Extract: First, the source of the data determines the difficulty of the extract phase. Data is often scattered throughout several information systems. However, Mans *et al*. [2009] showed that a hospital's billing system provides all necessary data in one single system, making it ideal for data extraction. The drawback of using these billing systems is the lack of detailed timestamps, as it is often only known on which day an activity was executed and it is not always possible to determine the exact order of activities that were executed on the same day. However, this does not necessarily mean that the resulting process model is incorrect and with additional information it could be possible to determine the exact order of events. In the former case, mining algorithms would place the activities in a parallel relationship and in practice many activities are indeed performed in parallel (according to Mans and Weijters).

Besides the billing systems, a variety of information systems can be used as a data source and the extract phase is highly dependent on the availability of such information systems. As a result, it is difficult to synthesize the previous research into a global approach for the extract phase and only recommendations on the type of systems that should be considered can be made.

Transform: During the transformation stage, possibly the most important steps of the complete process mining methodology are performed and each case study spends a significant amount of time and effort on the transformation of data and the construction of the database that is used for analysis in ProM.

Despite several available options for the creation of a database, mostly MS Access software was used during this part of the analysis (although this is not explicitly mentioned in all reports). Mans [20XX][9] has written a tutorial for the creation of a database in MS Access that can be transformed into the .MXML format with ProM*import*. The database has to be constructed according to a certain pattern in order for ProM*import* to be able to deal with the input (i.e. .mdb file) and create the desired output (i.e. .MXML file). The resulting database layout is based on the structure of the .MXML file (section 2.3) and contains four compulsory tables that need to be constructed. As a result, certain knowledge of the data and software (some Visual Basic scripts and knowledge are necessary to automatically fill the four tables) is required for this step. In addition, MS Excel can be considered an alternative for the pre-processing steps and was the preferred software by some researchers [Riemers, 2009; Ramos, 2009] as it made future pre-processing steps easier.

During the transformation phase of the database it is possible to manipulate the data by deleting or adding certain data objects to the file. For instance, creating artificial time stamps, aggregation and renaming of events, deleting incomplete care trajectories and removing insignificant data are a few of the pre-processing steps that we have identified. From previous research we have learned that these steps are essential in order to create event logs that produce sensible and understandable process models. Mining on global and raw event logs usually results in complex “spaghetti-like” process models that are not usable by the process experts.

The pre-processing techniques are dependent on the type and completeness of the data that was extracted (e.g. more data objects allow for more choices with regard to clustering). Moreover, sufficient knowledge of the actual process is required. For instance, healthcare professionals can indicate that certain activities are not of particular interest and can therefore be removed. Such process knowledge could also prevent the researcher from making mistakes, such as the removal of important events. In addition, healthcare professionals could indicate what parts of the process are of particular interest (e.g. they could be interested in patients with a specific DBC-code). Sufficient process knowledge could save time and effort and improve the resulting process models. Therefore, below we will elaborate on the specific pre-processing steps that have been used in the case studies.

First, as we have learned, many datasets contain information on events that are of little interest to the process experts. An excellent example is the common presence of large numbers of laboratory tasks

[9] Refer to http://www.win.tue.nl/processmining/_media/tutorial/msaccessdatabasetutorial_v2.zip (retrieved 8 August 2011) for additional information on the creation of the database in MS Access.

[Mans *et al*., 2009; Riemers, 2009; Ramos, 2009]. These events can make up for the larger part of the event log and are of no particular interest for the global behaviour in the event log. Therefore, it is recommended to aggregate such events and there are several possibilities to achieve that goal. Mans *et al*. [2009] used the ProM software, whereas Riemers [2009] and Ramos [2009] used MS Excel, and Zanden [2010] used MS Access.

In ProM, the first step is to rename the laboratory tasks using the *Remap Element Log Filter*. As an example, the researcher decides to rename all laboratory tasks to the department that executed the task (e.g. Algemeen Lab Klinische Chemie). Subsequently, the activities in the event log can be aggregated automatically in ProM using the *Repetitions-to-Activity filter*. The result is a smaller and more understandable event log.

There is no mention of any explicit methods which Riemers and Ramos used in MS Excel for the renaming and aggregation of events. However, both have indicated they used Excel because of the simplicity and freedom compared to ProM. In addition, similar pre-processing steps can be performed in MS Access [Zanden, 2010].

Second, as we learn from Riemers, Ramos and Zanden, DBC-code information plays a crucial role during pre-processing of the data. The presence of multiple DBC-codes is an indication of a heterogeneous dataset. Creating clusters of patients with similar DBC-codes would result in subsets with patients that have more similar processes, creating a more homogenous dataset. As a result, the process model for such a group has an increased chance of being understandable and achieving a higher fitness. Furthermore, Zanden stated that datasets with a higher number of process instances have an increased chance of producing “spaghetti-like” process models (although this could be up for debate as it depends on the types of processes, e.g. it might not apply to more linear processes outside the field of healthcare). Therefore, in any situation it is recommended to create smaller subsets of data. In Excel and/or Access, basic coding knowledge can be used to cluster process instances based on a specific data characteristic (such as DBC-codes). In ProM, the *Attribute value filter* can be used as a method to select for specific data characteristics. However, as was stated by Mans and experienced during this thesis, the Attribute value filter is prone to errors and does not always work. Additionally, Ramos [2009] has used the *LTL checker* to create subsets of patients with similar DBC-codes.

Third, as is mentioned in several reports [Mans *et al*., 2009; Ramos 2009], administrative tasks (or any other activities that are not of interest) can be removed from the dataset. This can be performed in ProM with the *Event filter* or by simple deletion in Access or Excel.

Load: With ProM*import* it is possible to convert the database from an .mdb file to an .MXML file for analysis in ProM. At the time which the six case studies were executed, the alternative Nitro tool was not yet available.

To conclude, the transformation of the data and the construction of the final database is not a straightforward approach, it is highly dependent on many factors and several software tools can be considered. Accordingly, extracting a general pre-processing methodology is difficult. None of the six

cases have provided a complete documentation on their methodology. However, we have identified several common important pre-processing steps:

- It should be considered to delete the administrative tasks or tasks that are of lesser interest.
- It is important to decrease the number of event classes in the dataset by renaming and aggregation of events (in healthcare data, laboratory tasks should especially be considered).
- Certain data objects, such as DBC-codes, should be used to create clusters of process instances with more similar processes (i.e. increase homogeneity).
- There are several software tools available for the pre-processing steps. ProM has a wide range of straightforward possibilities, but the (perhaps more complicated) MS Excel software allows for more freedom.

4.4.4 Mining methodology

Before actual process mining is started, most researchers [Gupta, 2007; Mans *et al.*, 2008, 2009; Riemers, 2009; Ramos, 2009] have added artificial start and end tasks to the event log as a means to increase process model understandability and fitness. This task is easily performed in ProM with the *Add artificial start/end task log filers*. In addition, to increase the understandability of the process model, the *Enhanced event log filter* is frequently used as a method to filter for event classes that occur less than a specified percentage in the event log. This allows the researcher to focus only on the most common events in the event log during the mining of the process model.

The *HeuristicsMiner* is the most frequently used mining algorithm to obtain a process model. However, most studies have only focused on the use of the default parameter settings of this plug-in. Therefore, depending on the nature of the dataset (i.e. raw, pre-processed, clustered etc.), the results have varied from "spaghetti-like" to understandable process models with a high fitness. The initial mining results can be used as idea generation for additional pre-processing steps. The newly created dataset in its turn can be analyzed according using a similar methodology. Therefore, it can be concluded that process mining is an iterative process, going back and forth between the pre-processing and mining stages.

In addition, the *Fuzzy Miner* [Günther & Aalst, 2007; Günther, 2009], *Genetic Miner* [Alves de Medeiros, 2006] and *Disjunctive Workflow Schema miner* (DWS miner) [Gupta, 2007] have been considered by some researchers to extract process models from the event log. However, these algorithms have only been used in a proof-of-concept manner only and did not nearly receive the level of attention that the HM was given. In addition, the DWS miner was not rated as a significant improvement of the HM.

Furthermore, several clustering algorithms have been considered as a means to cluster process instances and create more homogenous event logs. *Trace clustering* [Song *et al.*, 2009] (more specifically, the *self organising maps* algorithm with *Euclidean distance*) has been the most widely applied clustering algorithm [Mans *et al.*, 2009; Riemers, 2009; Ramos, 2009] and has produced varying results. In addition, the *Association rule miner* (ARM) was developed by Gupta [2007] and allows the clustering of process instances based on association rules. However, this algorithm has not been applied in any of the other five cases. Applying a mining algorithm such as the HM to a cluster should result in smaller and more understandable process models

4.4.5 Identified problems/inconsistencies/errors

During the analysis of the six cases, numerous problems with regard to missing information, inconsistencies and errors have been identified. Since these problems have already been extensively discussed in chapters 1 and 2, only a short overview is presented in this section.

Incomplete process mining methodology: All cases which have been analyzed were characterized by the lack of details on the process mining methodology. For instance, information on pre-processing steps was missing, no process mining algorithms are specified and the different steps of the analysis that were reported were scattered throughout the report. These problems increase the difficulty to understand the methodology that was used by the researcher (refer to Appendix B for the case by Zanden [2010] for the feedback by Tom Joosten on this part).

Missing information on datasets: Most studies failed to report detailed information on the datasets that have been used. As a result, it is difficult to understand what data attributes were available to the researcher and what parts of the data have been used. In addition, on several accounts it is not specified what modifications have been made to the data and it was not explicitly mentioned which datasets have actually been used during the analysis.

Missing information on software that is used: Information on the software that was used for the pre-processing steps is lacking in several reports, thereby making it difficult to retrace the steps of the original researcher.

Missing information on parameter settings: Most studies have only moderately (at best) considered the use of the different parameter settings for each plug-in. As is described by Gupta [2007], changing the parameter settings for the HM can dramatically influence the results (for the better or the worse). The other five cases have barely taken these consequences into account, using only the default settings or occasionally changing a setting or two. This approach could be an explanation for the disappointing results so far. Even worse, most researchers have not even reported the parameter settings that were used during their research, making it complicated to replicate the alleged results.

No fitness measure: Only occasionally were the fitness measures of the process models mentioned, making it difficult for the reader to infer how well the model represented the behaviour in the event log.

4.5 Interviews

In addition to literature, one of the most important sources of case study information is the interview [Yin, 2009]. Therefore, as an additional source of information, a short series of interviews was conducted to consult process mining experts on their process mining methodologies in healthcare. Whereas traditional process mining literature (i.e. the research papers with detailed information of process mining algorithms) mainly focuses on the use of a single mining algorithm, these interviews allow the experts to comment freely on the powerful combination of the variety of pre-processing steps and ProM plug-ins. Therefore, their expert opinion could serve as a powerful addition to the knowledge that was extracted from the six case studies.

4.5.1 The interviewees

The following process mining experts have been consulted during this project: dr. A.J.M.M Weijters, J.C.B. Rantham Prabhakara M.Tech, dr. A. Rozinat, dr. C.W. Günther and dr. ir. R.S. Mans. These experts have contributed to process mining research as ProM developers, researchers and lecturers. A brief introduction to the researchers and their work is provided in Appendix E (and Appendix C for Mans).

4.5.2 Interview questions

As the interviews were mainly focussed on methodologies, it was chosen to perform an open-questioned interview. Such an approach allowed us to zoom in on specific points of interest and skip points that were of lesser interest [Saunders *et al.*, 2000]. However, to guide the interviews, a basic format containing an introduction and questions with regard to the different stages of process mining research was created before the interviews were conducted. This format was sent to the interviewees upfront to indicate the goals and theme of the interview. This allowed the interviewees to prepare themselves. The complete interview format is presented in Appendix D.

4.5.3. Results of the interviews

In this section, only the most important results of the interviews are presented. The complete transcripts can be found in Appendix E (the answers by Mans are included in the report of the study research by Mans *et al*. [2009] in Appendix C and have also been integrated in the cross case analysis in section 4.4).

Interview with dr. A.J.M.M. Weijters

- The first step of the analysis is to investigate the data using the *log summary*, *performance sequence diagrams analysis* (PSDA) [Alves de Medeiros & Weijters, 2009] and the HM. Combined, these tools should provide the researcher with a sense of dataset complexity and variability in the process. For instance, a high number of different patterns and event classes indicate that deletion or aggregation of events should be considered. Such actions should always be in discussion with process experts who can indicate points of (dis)interest.
- For actual process mining, the HM is the preferred algorithm by Weijters and he recommends the following method to obtain a process model:
    a) Mine with *positive observations* set to 1, *all-activities-connected-heuristic* switched off and the *dependency threshold*, *length-one/length-two loops threshold* and *long distance dependency* set to 0,95.
    b) Filter the non-connected events and low frequent dependencies using the simple *Event filter* (in consultation with process experts).
    c) Start a new mining task on the filtered event log (obtained after *b*) with *positive observations set* to 1 and *all-activities-connected-heuristic* switched on.
- Miscellaneous comments:
    - The *LTL checker* [Aalst, Beer & Dongen, 2005; Beer & Brand, 2007] can be used to filter or cluster process instances based on certain data characteristics such as DBC-codes.

- The long distance measures, as well as the length-one/two loops are not really important during mining.
- It is recommended to always use both the artificial start and end tasks.

Interview with J.C.B. Rantham Prabhakara M.Tech

The nature of the interview with Rantham Prabhakara took a different turn, as his research focuses on process methodologies that are very different from the methodologies that we have identified in the six cases. As a result, his work is somewhat isolated from the research that we have considered during this thesis (and due to the limited focus cannot be used in this research). To illustrate, Rantham Prabhakara focuses his research on the development of the *Pattern Based Abstraction Viewer* (which is only implemented in ProM 6.0 and higher) [Rantham Prabhakara & Aalst, 2009], an algorithm that is used to automatically identify and extract patterns in the event log. In addition, he agrees with Weijters on the careful consideration of parameter settings for the HM and would recommend a similar approach.

Response of dr. A. Rozinat and dr. C.W. Günther (by e-mail)

The experts emphasize the iterative nature of process mining (methodologies) and indicate that they do not use something as a "generic" process mining methodology. However, there are several points that deserve attention in any process mining research:

- It is important to consider the goal of the process mining analysis and accordingly select data during the early stages of the analysis. For instance, during the data collection stage it can be determined to focus on diagnosis or treatment activities only.
- Accordingly, it is important to define the level of detail that is desired. To illustrate, is the goal merely to identify global process behaviour or is one looking for exceptions?
- Frequently used pre-processing steps are: removing less frequent events and events that are not in many of the cases (using the *Enhanced event log filter*). Moreover, they recommend renaming events using the *Remap element log filter* and filtering based on performance.
- To obtain clusters of similar process instances, *trace clustering*, *sequence clustering* and *activity mining* could be considered. With explicit knowledge the LTL checker can be used.
- For actual mining, the *HeuristicsMiner* and *FuzzyMiner* (FM) should be considered.
- Conformance checking can be achieved with the *conformance checker* or the f*uzzy model conformance indicator*.

4.6 Integrated process mining methodology

The resulting steps of the methodologies that have been identified in sections 4.4 and 4.5 are summarized in table 4.2. For the synthesis of the "global" process mining methodology, the explicit choice was made to limit the focus only to the most commonly used and promising steps (to avoid an information overload and establish a solid foundation for the best practices). As a result, steps such as the ARM, Genetic Miner (GM) and the pattern abstraction have not been taken into account. However, we emphasize that in additional research these steps could prove their value and therefore should be considered in future research.

It is important to consider the process mining methodology as an iterative process and no clear order of steps can and has been established yet. However, figure 4.1 visualizes the following steps that have been marked as most important for this thesis project:

- The *data collection* phase is highly dependent on the types of available information systems. In any case, data objects such as *timestamps*, *case IDs*, *events*, *originators* and *DBC-code information* should be included in the dataset. Additional information could serve as input for clustering steps later in the analysis. One should keep in mind the goals of the analysis and consequently adjust the required data and pre-processing steps.

- *Data transformation* could be achieved in either MS Access or MS Excel, the latter is preferred by some researchers. The goal is to check the data for completeness (and supplement when required) and remove all insignificant data.

- The combination of *renaming of events* and *aggregation of events*, either in ProM and/or MS Excel are crucial steps to decrease the number event classes in the event log and could significantly increase process model understandability and fitness. However, such actions should normally only be considered in consultation with process experts.

- *Creating an event log* traditionally has been performed with ProM*import*. At the time of the studies that have been considered during the multiple case study, the alternative Nitro tool was not available to the researchers yet. However, the Nitro tool has significant advantages over the ProM*import* tool as it provides much more options and does not require a complex procedure to create an event log. Additionally, Nitro has the ability to inspect the data with a variety of statistics without having to load the event log in ProM.

- *Clustering* can either be performed manually in Excel/Access or automated in ProM. In the former case, the focus of clustering is mostly on predetermined data objects such as DBC-codes. By clustering based on such information, more homogenous sets of process instances are obtained. Algorithms such as *trace clustering* automatically create clusters and do not require specific process knowledge. However, in these cases the clustering is not based on an easy identifiable data characteristic.

- The *Enhanced event log filter* is frequently used as a method to filter for event classes with a low frequency or that do not occur in many process instances. Decreasing the number of event classes reduces process model complexity and allows focusing on the main behaviour in the event log.

- It is recommended to always *add artificial start and end tasks* to the process model as this both increases the understandability and the fitness of the process model.

- The *HeuristicsMiner* is the main mining algorithm and is frequently used with default parameter settings only. However, Weijters has elaborated on a different approach that uses the HM to identify events that do not fit the process model, remove such events with filters and restart the analysis.

As became obvious during the analysis, process mining is an iterative process that is difficult to visualize as a simple linear methodology. Furthermore, the order of events is dependent on the software that is used. For instance, when renaming and aggregation is performed in ProM, first and event log has to be created. However, when both steps are executed in Excel it is not necessary to create an event log first. Therefore, at this moment figure 4.1 only serves as a global indication of the process mining steps that are used as the foundation for the development of the new methodology and the exact order has yet to be determined. To summarize, the following steps will be considered during the development of the new methodology: data collection, data transformation, renaming of events (ProM and Excel), aggregation of events (ProM and Excel), creating event log (ProM*import* and Nitro), clustering (manual and automated), event log, enhanced event log filter, artificial start/end tasks and the HeuristicsMiner.

ProM plug-ins and pre-processing activities		Cases								
		Mans et al. [2008]	Mans et al. [2009]	Riemers [2009]	Ramos [2009]	Zanden [2010]	Gupta [2007]	Weijters	Rantham Prabhakara	Rozinat & Günther
Pre-processing	Data transformation (various)	X	X	X	X	X	X	X	X	X
	Renaming of events (ProM)		X							X
	Renaming of events (Excel/Access)			X	X	X				
	Aggregation of events (ProM)		X							
	Aggregation of events (Excel/Access)			X	X	X				
	Creating event log (ProM*import*)	X	X	X	X	X	X	X		
	Creating event log (Nitro)									X
Clustering	Trace clustering		X	X	X					X
	Manual clustering (Excel/Access)			X	X	X				
	LTL checker				X				X	X
	Association rule miner						X			
Mining	Enhanced event log filter			X	X					X
	Add artificial start/end task	X	X	X	X		X	X	X	X
	HeuristicsMiner	X	X	X	X	X	X	X	X	X
	FuzzyMiner		X		X	X			X	X
	α-algorithm	X								
	GeneticMiner				X					
	DWS miner					X				

Table 4.2: Summary of the different steps that were identified in the process mining methodologies of the six cases and during the expert interviews.

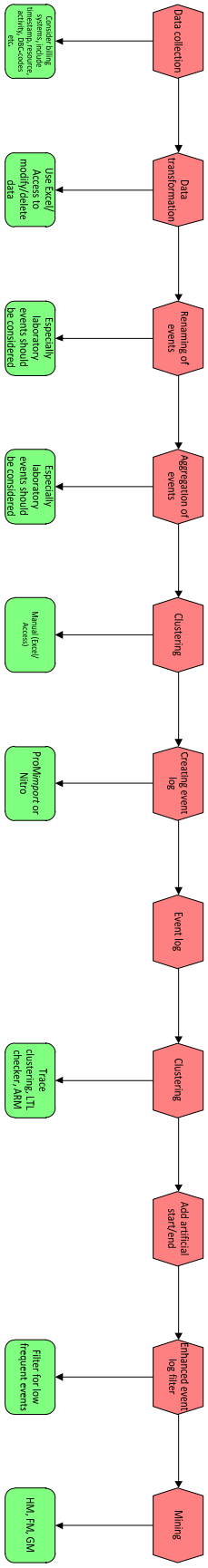


Figure 4.1: Synthesized process mining methodology from the six case studies and expert interviews.

5. Development of a method to report the best practices

One of the challenges that we faced during this thesis was to find a method to report the new methodology and best practices for process mining in healthcare. In the current process mining literature, no standardized method of reporting methodologies is used and as we have showed in the preceding chapters, this leads to several major problems. As a result, there is a need for a clear, structured and proven method for reporting process mining (and its techniques) and the process mining best practices. In this chapter, three methods are presented that were identified as the most promising solutions to these problems: pattern language, virtual machines and screencasts.

5.1 Pattern language

The method to report process mining methodologies should meet two criteria. For one, it should be accessible. As one of the goals of the thesis is make process mining available to a broader public, the method of reporting should be easy to understand and require very little knowledge on process mining. Secondly, the method should be reproducible. The new methodology should be applicable in a number of different cases. Therefore, besides a detailed description for process mining in the healthcare specific situations that we have studied, it is important to be able to create a higher level of abstraction in the methodology so that it is applicable in other healthcare case studies. Furthermore, process mining can be considered in a range of industries and is not limited to healthcare only. In the most favourable scenario, the methodology should be applicable to other business industries as well. In addition, the method should also be able to improve the methods of reporting process mining methodologies that have been used by researchers during their analyses, in order to make replication of the alleged results possible and allow other researchers to transfer the methodology to their research.

One of the possibilities that we have identified as method to report process mining methodology is the so called "*pattern language"*. According to Alexander *et al.* [1977], in a pattern language "*each pattern describes a problem which occurs over and over again in our environment, and then describes the core of the solution to that problem, in such a way that you can use this solution a million times over, without ever doing the same way twice.*" Subsequently, a pattern language is a "*network"* of patterns that call upon each other (i.e. there is a directed relation between the patterns). In a different environment, Demeyer *et al*. [2009] describe design patterns as *"generic solutions to recurring design problems. It is because these design problems are never exactly alike, but only very similar, that the solutions are not pieces of software, but documents that communicate* ***best practice****."* In many fields of research it is likely that researchers are often confronted by similar problems for similar projects. Therefore, a set of simple global solutions in the forms of patterns is suggested to deal with these common problems. As a result, this prevents researchers from reinventing the wheel over and over again.

To illustrate the use of patterns consider a word as a pattern. The relationship between words makes sentences and creates context. With certain language rules concerning syntax and grammar, an understandable and reusable language can be formed, allowing for a large amount of different variations (i.e. different paths throughout the set of patterns). Therefore, as a rule, each pattern at least has a name, a description and some cross references (as words have in a traditional dictionary).

Pattern language has already proven its worth in fields such as architecture [Alexander *et al*., 1977; Alexander, 1979] and computer science [Gamma *et al*., 1995; Demeyer *et al*., 2009][10], and to date there is still much attention to pattern language in literature [Noble *et al.*, 2010 & 2011] and on conferences such as PLoP[11] (Pattern Languages for Programs), EuroPLoP[12] and ChiliPLoP[13]. Furthermore, Mulyar & Aalst [2005a & 2005b] have worked on a pattern language for Coloured Petri Nets as a part of the Workflow Patterns Initiative[14] "*to help developers to build their models efficiently, while avoiding reinvention of already existing solutions of problems*".

During the development of a pattern language, the designer enjoys much freedom and the contents of the language largely depend on the nature and field of research. Indeed, in literature and on the internet a wide variety of pattern languages with different elements can be found. However, Alexander *et al*. [1977] have stated some criteria that are required in order to develop a "good" pattern language. For instance, the previously mentioned name, description and cross references are considered essentials in any pattern language in any field. Furthermore, generalizability is an important aspect that should be considered which implies that the patterns should be applicable in completely different systems as well (e.g. in the context of this research: the patterns for process mining should be applicable in fields besides healthcare as well). In addition, the level of abstraction can vary and the pattern network can be considered hierarchical in some cases, dependent on the designer and the intended goal. The links between patterns guides the reader through the network of patterns and the corresponding descriptions help the reader choose the right pattern (when there are alternatives). Alexander *et al.* [1979] state that "*these links between the patterns are almost as much a part of the language as the patterns themselves.*" Different researchers may add patterns to the language or modify existing patterns (e.g. when new process mining algorithms are developed), much as the real language we use on a daily basis, which is under constant revision. In any case, the patterns should be clear and reusable by ordinary people with ordinary intelligence to solve large and complex problems.

5.2 Potential of pattern language in process mining

In section 5.1 we have introduced the concept of pattern language, its applicability and advantages. In this section we present several reasons why we have introduced a pattern language into the field of process mining. To illustrate, in this context a pattern could refer to a certain step in the process mining methodology such as data collection or the HeuristicsMiner. A complete set of such patterns would form the pattern network which represents all process mining possibilities. A path through this network would represent a process mining methodology. As a result, the pattern language creates a benefit, both for reporting process mining research, as well as reporting the new methodology and best practices:

[10] Refer to http://www.soapatterns.org (retrieved 8 August 2011) for more applications of design patterns in computer science.
[11] Refer to http://www.hillside.net/plop/2011/ (retrieved 8 August 2011) for information on PloP.
[12] Refer to http://www.hillside.net/europlop/europlop2011/index.html (retrieved 8 August 2011) for information on EuroPloP.
[13] Refer to http://hillside.net/conferences/chili-plop (retrieved 8 August 2011) for information on ChiliPloP.
[14] Refer to http://www.workflowpatterns.com (retrieved 8 August 2011) for information on the Workflow Patterns Initiative.

Reporting process mining research: As we have learned from the analysis of previous process mining research, it is unclear how and why a certain methodology was used. Especially for readers that do not posses experience with process mining it can be time consuming to understand the research, its results and it can be difficult to put all the different elements of the research into perspective. Moreover, the lack of such details significantly decreases reproducibility. Furthermore, current literature on the existing process mining algorithms is various and complex and does not signify how to actually use the algorithm (in this case we refer to papers that were written by the developers of a specific process mining algorithm such as Weijters *et al.* [2007]). As a result, these algorithms require skill and time to master, reducing their accessibility and understandability. To counter these problems, introducing a pattern language as a method to report process mining methodology may increase accessibility, understandability and reproducibility of research and can contribute in the following manners:

- According to Demeyer *et* al. [2009], "*obsolete or no documentation"* and a "*limited understanding of the entire system"* are clear signs in software engineering that problems are on the horizon. In this thesis we have discussed the lack of detailed process mining documentation and its resulting problem areas. In addition, we showed that the present methodologies are limited and have not yet reached their full potential as only a limited number of process mining algorithms have been well tested and applied in a healthcare setting. A contributing factor to these problems is the lack of a complete overview of the present process mining possibilities (as was stated by Zanden). Therefore, consider a repository of patterns for all possible process mining steps in the methodology (such as aggregation of events and the HeuristicsMiner). A repository would be a central place for researchers to consult on the different possibilities and techniques in process mining. Instead of having to gather and read numerous papers on process mining algorithms that do not have a clear and direct connection to each other, users can consult the repository for a quick understanding of the algorithms in the form of a pattern and their relation to other process mining patterns in the network. Furthermore, the nature of the pattern language allows for modification and extension whenever existing process mining steps are updated or new process mining steps (e.g. new algorithms) are introduced. When new mining algorithms are introduced, providing a simple and clear pattern would allow users to easily and quickly understand the new material without having to read large and complex scientific papers (e.g. as healthcare experts are not necessarily interested in the specific mathematical algorithms that are used by a pattern). Using a uniform single language for reporting process mining methodology would increase accessibility and additionally allows for easy comparison of the different process mining steps that are available.
- By creating a repository of patterns, future researchers can refer back to a certain process mining pattern (such as the HeuristicsMiner) in the methodology when it is used during the research. Readers can consult the repository and search for the respective pattern to quickly learn about its goal, specifics and place in the complete process mining methodology. As a result, readers do not have to search for and read numerous and complex scientific papers (which are scattered throughout the internet) themselves. This puts the decision of the researcher into perspective, without having to explain the same pattern over and over again.

Moreover, by creating a chain of patterns it should be easy to demonstrate and reproduce a process mining methodology.

Reporting best practices: Demeyer *et al*. [2009] state that "*patterns turn out to be a compact way to communicate best practice: not just the actual techniques, but the motivation and rationale behind them*". They found that patterns are the most natural way of documenting the best practices they have encountered during their reengineering projects. Therefore, a pattern language can also be used to report the new methodology and best practices for process mining that are developed in this thesis as it allows for simple and quick description of the different steps in the methodology. Furthermore, users can quickly identify the necessary patterns and their context. A best practice can be referred to as a chain or network of specific patterns that are important to a specific problem, thereby including only a small part of the complete set of patterns in the network. For different situations or problems, different best practices can developed, each referring to a different set of process mining patterns that should be considered.

5.3 Template for the process mining patterns

In the previous section we have introduced pattern language and discussed the potential of patterns in process mining research. However, a process mining specific pattern had to be developed for this thesis. As was stated, the designer of a pattern language enjoys much freedom during the development. Therefore, we have used the work by Alexander *et al*. [1977], Gamma *et al*. [1995] and Demeyer *et al*. [2009] as a foundation for the process mining patterns. The contents of the pattern is based on their research (i.e. what are minimal requirements for a pattern) and process mining literature (i.e. what elements are important in process mining). This resulted in the following pattern template that shall be used to present the new methodology and best practices:

1. ***Pattern Name*:** The name of the pattern is important and should be short, concise and present the essence of the pattern.
2. ***Intent*:** A description of the goal and essence behind the pattern and the reason for using it.
3. ***Also Known As*:** Depicts alternative names for the pattern (if any).
4. ***Author/date*:** For future reference it could be convenient to know the editor of the pattern. Furthermore, the date gives an indication whether the pattern is up-to-date.
5. ***Consider These Patterns first*:** What patterns provide input for this pattern?
6. ***Alternatives*:** What patterns can be used as an alternative for this pattern?
7. ***Implementation*:** State software and versions that can be used for the pattern.
8. ***Motivation*:** Why does the pattern make sense and when is it applicable?
9. ***Involved data objects*:** Explains what parts of the data are important (e.g. events, originators, data attributes etc.), why and what they do.
10. ***Solution*:** States the actual solution of the pattern. This can include a short guide on how to perform the different steps, as well as an illustration of the solution.
11. ***Screencast*:** Provides a link to a screencast (see section 5.5), which will shortly explain and demonstrate the pattern's use.
12. ***Trade*-offs:** State the pros and cons of the pattern.

13. ***Known Uses*:** Examples of real uses of the pattern. A pattern can only be considered as pattern when it has been applied to a real world solution at least three times. Tracz [1995] calls this "*The Rule of Threes*". Otherwise, it is known as a proto-pattern[15].
14. ***Consider Next*:** Which patterns can be used as a follow up of this pattern?

5.4 Sharing Hosted Autonomous Research Environments (SHARE)

As we have learned from literature on research methodology [Aken *et al*., 2007; Yin, 2009; Saunders *et al.*, 2000], reliability/reproducibility is an important aspect of scientific research. To illustrate, using empirical research, Vandewalle *et al.* [2009] demonstrated that in the field of computer science there is a causal relationship between reproducibility of the presented research and the research impact of the corresponding paper. In addition, Pinowar *et al.* [2007] have demonstrated similar effects in the field of biomedical research.

A possible method to increase reproducibility is by facilitating reviewers to reproduce the alleged results. Distribution of required data files amongst reviewers could be considered. However, such an approach requires the reviewers to download and install all necessary software, which could be limited due to legal reasons or practicalities. Furthermore, ownership of the original data files is shared and this is not always desirable (especially in healthcare where privacy is a controversial topic). Therefore, in the ideal situation, reproducibility does not require registration, installation of software and transfer of data.

As a means to facilitate reproducibility in scientific research, Gorp *et al.* [2010 & 2011] have developed the SHARE[16] platform (Sharing Hosted Autonomous Research Environments) which allows researchers to build and share virtual machines [Dittner & Rule, 2007]. A VM can be regarded as a completely isolated guest system platform that supports the execution of a certain operating system and software. The advantage of such a platform is that users are only required to register for a certain session, i.e. the user does not need to install specific software and it is platform independent. In addition, software and data are limited to the specific VM and it cannot be transported out of the virtual world.

To illustrate the use of VMs in SHARE consider the following situation. An administrator can create new VMs for inspection by other reviewers (accessibility is controlled by the administrator). In the VM, certain software tools (such as ProM) and datasets (such as the AMC data) can be added that have been used by the administrator during his own research projects. When the VM is published (opened to public), this allows peers to use the exact same software, procedures and datasets that have been used in the original research and thereby reproducing the exact same results without having to install or download any software and/or data. In addition, the reviewers are limited to the use of the software

[15] Refer to Cunningham & Cunningham Inc., http://c2.com/ (retrieved 8 August 2011) for additional information on proto-patterns and pattern language.

[16] Gorp, P.M.E. Van, Blom, S., Belinfante, A. SHARE-Sharing Hosted Autonomous Research Environments. Refer to the SHARE website http://is.tm.tue.nl/staff/pvgorp/share/ (retrieved 8 August 2011).

and data in the VM only and cannot import or export any contents. Furthermore, any modifications to the VM are discarded once the VM session is closed, leaving the VM unaltered.

In the context of this thesis, VMs could serve a twofold purpose. First, a VM could serve as an introduction to the process mining patterns and best practices. In such a case, a VM with the necessary software, datasets and overview of the process mining patterns would allow anyone to test the patterns. This is especially convenient for people who do not have access to the necessary software and data. Moreover, by confining the boundaries of the process mining field within the possibilities of the VM and the developed pattern language, new users are less likely to be overwhelmed by the complete set of complex process mining possibilities that are scattered throughout numerous papers and reports.

Second, VMs could be considered by any researcher as a means to allow peers to reproduce the alleged results of his/her research. For process mining research this would mean the creation of a VM with the right software and data that has been used in the original process mining research. Despite the fact that VMs facilitate replication of research results, they would still require the researcher to document its methodology in a proper and accessible way.

5.5 Screencasts

In addition to the pattern language and VMs, screencasts[17] (i.e. digital video recording of a computer screen output) were identified as a simple and accessible method to report process mining research. In the context of this thesis, screencasts would serve a twofold purpose. First, screencasts were used as an additional method to show and explain the process mining patterns by translating the written pattern to a short video. Second, screencasts could also be considered as a method to report process mining research. In addition to providing a written report on the research methodology, the actual analysis itself could be filmed and published as well. This would provide the readers with the exact and complete details of the methodology and results of the original researcher. Combined with the pattern language and VMs, screencasts could be a powerful method to increase reproducibility and accessibility.

[17] Refer to http://www.techsmith.com/jing/ (retrieved 8 August 2011) for the software that was used during this thesis to create the screencasts.

6. Development of the process mining patterns

In chapter 4 we have synthesized a "global" process mining methodology from current process mining research and knowledge by several process mining experts. Subsequently, in chapter 5 we have introduced the concept of pattern language as a means to communicate process mining methodology and best practices. Therefore, the next logical step was the actual development of a new process mining methodology and a set of best practices for process mining in healthcare in the form of a pattern language.

In section 6.1 we will first discuss the development of the complete set of process mining patterns which have been developed on three different datasets. The resulting patterns will be briefly presented and discussed in section 6.2. From the total set of patterns, a subset of patterns was selected which showed the most potential for process mining in healthcare. By testing the effect of these patterns on two datasets we aimed to prove their added value as best practice, which is depicted in section 6.3. Lastly, section 6.4 provides the final details on the best practices.

6.1 Creating the process mining patterns

In section 4.6 we aimed to create a synthesized "global" process mining methodology from several studies and process mining experts and the result is illustrated in figure 4.1. This model identifies the most important steps in the methodology and puts them into perspective. As a result we already globally knew *what* steps we should use and *when* we should use them. However, the synthesized methodology still lacked the details on the different steps that allowed us to specify *how* the steps should be used. Information on many elements such as software, algorithms and parameter settings is missing, which is essential information for a reproducible methodology.

For each step in the process, several factors had to be considered. For instance, for several steps we identified two possible software tools that are used in current process mining research. As a result, both alternatives were considered as a pattern. Considering there are many different possibilities that have not been considered in this thesis and the constant development which leads to new algorithms, we had to find a way to deal with alternative steps. Therefore, we introduced the concept of hierarchy. In the event a step contained several alternatives; we decided to create two levels of patterns (or even three). The highest level would present the global information on the step with a reference to the more dedicated and software/algorithm specific pattern. Moreover, this approach would not clutter the resulting pattern network with many alternative steps that in fact present similar results, but allowed us to refer to the highest level pattern which the user could use to determine the place in the methodology and decide on the software/algorithm specific pattern. As one will learn from the resulting patterns, lower level patterns sometimes inherit information from the higher level patterns. This is indicated by giving the inherited information an alternative (lighter colour). For this thesis, the alternative steps basically contain: 1) using MS Excel as a pre-processing tool and 2) using MS Access and ProM as pre-processing tools.

Besides the dependencies and alternatives of the patterns, another important factor was the actual information that discusses how the pattern should be used. As a result, information from the case

studies and experience gained throughout this thesis project (when no information was specified in the reports) was used to complete the template that was specified in section 5.3.

As was already mentioned in chapter 5, all patterns form a directed network, indicating the dependencies between the different patterns. Therefore, during the development of the patterns we have also considered these dependencies and translated them into a pattern network. A network as such serves as a guide through the methodology and places the different patterns into perspective.

In addition, as a means to increase the accessibility of the patterns, several screencasts have been created for a number of patterns which visualize the actual use of the pattern. For these screencasts, the focus was solely on the application of the pattern.

The patterns for this thesis were developed mainly on two datasets: 1) the AMC data and 2) the Italian hospitals data. In addition, the GGzE data was used to confirm certain suspicions and/or ideas. In addition, the following software has been used: ProM 5.2, Nitro 2 and 3 and MS Excel 2007. For the sake of brevity, the complete development of the patterns is described in Appendix F.

As we have learned, the HM parameters have a huge influence on the resulting process models. However, there is no such thing as an optimal setting, thereby requiring tweaking them for each individual mining iteration. Therefore, as can be observed in Appendix F, we have only considered the default parameter settings for the HM (with the exception of the HeuristicsMiner pattern). This was an *explicit* choice as levelling the effect of the HM parameter settings for all patterns made it possible to ascribe the differences in the resulting process models solely to the patterns, thereby making the results fair for all patterns and allowing us to compare them.

6.2 Resulting process mining patterns

The development of process mining patterns that is described in section 6.1 led to a repository of 22 patterns, describing the complete process mining trajectory from data collection to actual mining. Due to the lack of space we will only briefly discuss the patterns in this section. The complete list of patterns is also available in Appendix I, where the patterns are presented in full detail. In addition, a special website[18] has been developed and devoted to this research. On this website, all patterns are presented and the user can browse through the different patterns in an interactive way.

As a means to increase reproducibility, a virtual machine in SHARE was created that allows users to test the different process mining patterns[19]. For several patterns we have created dedicated datasets (in the forms of .csv or .MXML files) that allow the user to focus solely on one specific pattern. In addition, it is also possible to go through the complete pattern network (with a few exceptions). However, the VM is limited as the MS Access software was not included (because it is not included in the best practices, see section 6.3). In addition, screencasts are available for a number of patterns that demonstrate their applications. All links are also available on the dedicated website.

[18] Refer to http://sites.google.com/site/prompatternlanguage/ (retrieved 8 August 2011).
[19] Refer to http://is.ieis.tue.nl/staff/pvgorp/share/?page=ConfigureNewSession&vdi=XP-TUe10G-WithOfficeAndAcrobat_ProM_4.vdi (retrieved 8 August 2011) for the virtual machine to test the best practices.

As was discussed in section 6.1, some patterns use the concept of hierarchy. As a result, there are 11 patterns at the highest level of hierarchy and 11 patterns at lower levels. The dependencies between these patterns are illustrated in the pattern network in figure 6.1 (and figure I.1). In this network only the highest level patterns are depicted. However, by using coded dependencies, the relationship between the lower level patterns can also be inferred. In addition, we make a distinction between patterns that can be skipped and patterns that cannot be skipped. For instance, the researcher can decide not to apply *aggregation of events*. However, to go *data transformation* to *event log*, the pattern that refers to the *creation of event logs* cannot be skipped. Below we present the full list of 22 process mining patterns in an alphabetic order.

Add artificial start/end task: Adding an artificial start and end task is easily performed in ProM (under *advanced filters*) and increases process model understandability (as it makes the start and end more visual) and fitness.

Aggregation of events: In datasets, it is possible many "double" events are present for a case on the same day. An example is the large amount of laboratory tasks in hospital data. These are of no particular interest for the process and greatly increase the dataset's size and complexity. By aggregating such events, the total number of events (ATEs) in the data is reduced, decreasing process model complexity. This pattern contains two alternatives:

- **Aggregation of events (Excel):** In MS Excel it is possible to aggregate events by first identifying events that need to be aggregated and second to delete these events using a pre-developed macro. For this pattern, some coding knowledge is required but it allows more freedom than the alternative (for instance using Excel the user can determine which particular events should be aggregated).
- **Aggregation of events (ProM):** In ProM, the *Repetitions-to-Activity filter* can be used to aggregate events automatically. It only requires the addition of this filter to the event log and is therefore easier than the alternative. However, considerable less freedom is available since for instance ALL events are aggregated automatically.

Clustering: In large heterogeneous datasets, it is highly likely that cases share similar attributes. For instance, hospital patients are either male or

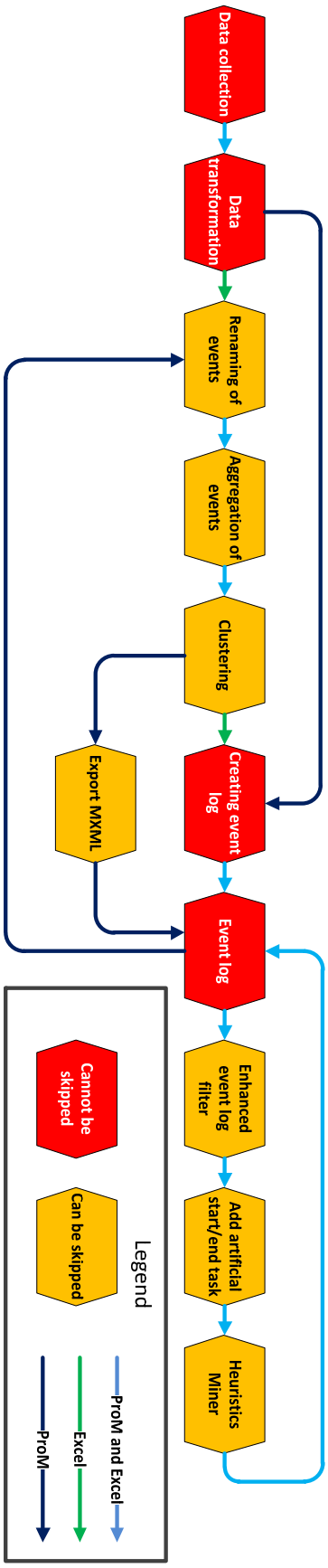


Figure 6.1: Process mining pattern network.

female. These attributes can be used to cluster cases to create a more homogenous dataset. The theory behind this pattern is that a process model is more likely to be less complex when the underlying data is more homogenous. This pattern contains two alternatives:

- **Clustering (Excel):** Using a pre-developed clustering macro it is easy to cluster PIs based on certain data attributes. For instance, in healthcare it should be considered to cluster patients based on their DBC-codes, thereby creating more homogeneous datasets (and hopefully less complex process models). This clustering macro is simple and allows clustering based on any data attribute. However, this approach is only useful when the researcher has some knowledge on the dataset and the case characteristics.

- **Clustering (ProM):** ProM contains a wide variety of automated clustering algorithms. Most of these algorithms do not require the researcher to posses any specific process knowledge or to specify a data attribute that is used as input for clustering. Currently this pattern contains one alternative:

    - **Trace clustering:** Trace clustering is the most commonly used automated clustering algorithm and usually the *self organising maps* with *Euclidean distance* is used. The result of this pattern varies and it cannot always be inferred why certain process instances are clustered (i.e. clustering is based on mathematical algorithms but can appear arbitrary in some cases when no obvious process model improvement is achieved).

Creating event log: The event log is at the basis of process analysis in ProM and can be created according to two alternatives:

- **Creating event log (Nitro):** The Nitro tool is a commercial product that can transform MS Excel files into the .MXML file for analysis in ProM. In addition, it features several analysis tools itself, as well as recently (July 2011) introduced filtering options. The benefit of Nitro is that it renders the complex steps that precede the alternative ProM*import* obsolete.

- **Creating event log (ProM*import*):** The benefit of ProM*import* is that is can deal with a wide variety of data formats as input. However, for the methodology that is subject in this thesis, extensive and complex data transformation steps (in MS Access) are required before the data can be transformed with ProM*import* into a .MXML file.

Data collection: The first step in data analysis is collecting a dataset. The most important data objects for process mining (and accordingly should be included in the dataset) are: *timestamp*, *case ID*, *originator*, *activity/event*.

Data transformation: There is a possibility that data which is extracted from an information system contains information that is not of interest to the researcher. Therefore, the data transformation stage serves to remove such information. Additionally, data can be checked for completeness and when required empty fields can be artificially filled. This pattern contains two alternatives:

- **Data transformation (Access):** In MS Access, data can simply be removed or added using the standard options that are available. In any case, the data objects mentioned under *Data collection* should be kept in the dataset.

- **Data transformation (Excel):** As with the alternative pattern, in MS Excel, data can simply be removed or added using the standard options that are available. In any case, the data objects mentioned under *Data collection* should be kept in the dataset.

Enhanced event log filter: The *Enhanced event log filter* is a simple filter in ProM that allows filtering of event classes that are low frequent and/or events that occur in few PIs. Reducing the number of event classes reduces process model complexity.

Event log: The event log is the basis for process mining in ProM and at least should contain a *timestamp*, *case ID*, *originator*, and *activity/event*.

Export .MXML: The export function in ProM is convenient when the researcher is interested in saving modified event logs for future analysis. For instance, when clustering has been applied, individual clusters can be saved as a new event log.

HeuristicsMiner: The HM is the most predominantly used mining algorithm in ProM. In most cases, the default settings are used to mine a process model. However, Weijters has elaborated on a different approach, using the HM to select and filter events that are non-frequent. Accordingly, both methods could be considered as a supplement and they are not mutual exclusive.

Renaming of events: The renaming of events is a useful method to reduce the number of event classes in the dataset. For instance, many events that are not of interest for the main process behaviour may be renamed to the originator that has executed them. This pattern contains two alternatives:

- **Renaming of events (Excel):** In MS Excel, some coding knowledge is required to rename events. However, using this pattern allows for much more freedom and speed compared to the alternative.

- **Renaming of events (ProM):** In ProM, renaming events is simple but can be time-consuming, as no automated method is available (as with the alternative pattern). Therefore, when many events need to be renamed it is not recommended to use this pattern.

6.3 Development of the process mining best practices

The pattern language and network introduced in section 6.2 contain a number of patterns that can be considered as alternatives (e.g. renaming of events in Excel or ProM). Moreover, in most patterns several decisions have to be made with regard to the execution of the pattern (e.g. renaming one or multiple event classes). As a result, there are a number of options available when the goal is to analyze healthcare process data. However, the goal of this thesis was to provide a set of best practices for process mining in healthcare, indicating what patterns to use and how to use them. As a result, we need to reduce the number of patterns for the best practices by eliminating the least favourable alternatives

and specify additional information on the application of the remaining patterns. Therefore, this section will present the selection and creation of the best practices.

In the pattern network we can distinguish two main approaches for the preparation of data. One approach makes use of MS Access, ProM*import* and ProM as pre-processing tools. The second approach uses MS Excel and Nitro for pre-processing tasks. These are patterns that occur before the actual mining in ProM (i.e. before the pattern *Enhanced event log filter*).

As we have learned during the development of the process mining patterns, using the MS Excel approach is much faster and easier compared to the MS Access approach (due to the increased freedom and ability to use the much simpler Nitro instead of ProM*import)*. As a result, we decided to focus our best practices on the use of the MS Excel approach. Therefore, we have retained the following process mining patterns as best practice: 1) Data collection, 2) Data transformation (Excel), 3) Renaming of events (Excel), 4) Aggregation of events (Excel), 5) Clustering (Excel), 6) Creating event log (Nitro), 7) Event log, 8) Enhanced event log filter, 9) Add artificial start/end task, 10) HeuristicsMiner and 11) Trace clustering. The resulting pattern network for the best practices is depicted in figure 6.2. As with the original pattern network in figure 6.1, we make a distinction between patterns that cannot be skipped and patterns that can be skipped.

To assess the quality of the best practices, during the development of the process mining patterns (Appendix F) we have also recorded the effect of the different best practice patterns on the process models.

Table 6.1 provides a quick overview of the effect of the best practice patterns on the AMC data and table 6.2 provides an overview for the data that originated from the Italian hospitals. Whenever a best practice was applied to the dataset, a process model has been created with the HM to observe the result (again for comparison we used default settings, see section 6.1). In both tables, for each pattern it has been indicated what exact steps have been executed (such as renaming and/or aggregation). Subsequently, the fitness (ExtraBehaviourPunishment) and several other details (e.g. number of PIs) on the resulting model are provided. Finally, it is judged whether the resulting process model was understandable and usable. The understandability depends on the layout of the process model and therefore is a subjective measure (and is largely influenced by the number of event classes). Understandability was rated as bad (red), moderate (orange) and good (green). The usability (same rating as understandability) depends both on the understandability and fitness of the process model. For the fitness measure, we generally accepted a value of over 0,5 as usable (however a few exceptions are identified). To increase validity of the best practices, in chapter 8 we have introduced validation of the process models that result from the best practices by process experts.

For all best practices, event logs were created according to the *Creating event log (Nitro)* pattern. When not mentioned otherwise, mining has been performed with the *HeuristicsMiner* with default settings. For the *Enhanced event log filter*, the first filter refers to in how many PIs the event class is present and the second filter refers to the total frequency of the event class in the event log. By default we used 1% for both filter.

Raw data: For both the AMC (spaghetti-model with 417 event classes) and Italy data (low fitness), mining on the raw data resulted in very large and complex process models that were not usable. The fitness measure for the AMC model was 0,48 and for the Italian model -0,31.

Data transformation (Excel): When removing the *complete* and *schedule* type events from the Italy data, the resulting process model has a decreased fitness measure (-1,17). However, the complexity is reduced as there are less event classes (from 95 to 45).

Renaming of events (Excel): For the AMC data, when only the laboratory tasks had been renamed (to the originator), the resulting process model had a slightly increased fitness measure (0,49). However, with 294 event classes remaining, the process model was still very complex. Renaming all events (to the originator) resulted in a process model that was much easier to understand with a fitness of 0,63. However, using this approach there is no detail on the exact events that have been executed.

Aggregation of events (Excel): For the AMC data, both the event log with aggregation of only laboratory tasks (from now on AMC 1) and aggregation of all events (from now on AMC 2) resulted in worse process models (a fitness of -0,04 and -0,35 respectively). In addition, the former model was still very complex (i.e. 294 event classes) and the latter had a moderate understandability, but due to the low fitness measure it was not usable. Aggregation for the Italian data did not result in improvement as the resulting process model had a fitness of -1,54 and the layout was still very complex.

Clustering (Excel): For the AMC data, clustering was performed based on the six DBC-codes for both AMC 1 and AMC 2. As can be learned from table 6.1, this did not result in any improvements with regard to both fitness and understandability for neither event log. For the Italian data, we performed clustering based on the four hospital codes that were available. In this case, clustering did lead to improvement. However, the fitness measures were still considered far too low and therefore the resulting process models were not usable.

Add artificial start/end task: Adding artificial start/end tasks to the global AMC event logs significantly increased the fitness measures of both event logs. However, AMC 1 still resulted in a complex process model (i.e. 294 event classes). AMC 2 resulted in a small process model (i.e. 28 event classes) and a good fitness. Therefore, the latter was considered usable. The event logs for DBC-codes M13 and M16 were considered (see Appendix B). Only the event log of M16 with all events renamed and aggregated was usable as it had a good fitness (0,51) and an understandable layout (i.e. 19 event classes). For the Italian data, this pattern has improved the models for all hospitals and the global event log. However, the only usable process model was obtained for hospital 4.

Enhanced event log filter: For AMC 1, when only the first filter was applied (<1% presence in PIs), the fitness decreased only slightly to 0,63. However, the number of event classes decreased to 111 and therefore the resulting process model was still complex. After the second filter (<1% presence in total), only 17 event classes remained and the fitness increased to 0,78, resulting in a usable process model. For AMC 2, filtering resulted in a decrease of seven event classes to 21. However, the resulting fitness measure did not change. For the different M13 and M16 event logs, several usable process models were created, which is an improvement compared to the results after the previous pattern. Especially for M13

this is an improvement as during previous patterns, no usable process model was obtained. In addition, for the Italian dataset, a major improvement was achieved for all datasets with applying either one or two filters. Apart from the dataset for hospital 1, all resulting process models were usable.

HeuristicsMiner: Using the approach that was suggested by Weijters we were able to create usable process models for both the AMC 2 and M16 datasets. An interesting note is that on some occasions it was better to specify the *positive observations* to 1, and sometimes it was better to specify it to 10. This again confirms the difficulty with process mining, making it difficult to create a global methodology that works in all situations. In addition, the resulting process models were slightly different compared to the models that resulted from the *Enhanced event log filter* pattern. For the Italian data, applying a similar approach produced a usable process model when *positive observations* was specified to 10. This result is also comparable to the result obtained after the *Enhanced event log filter pattern*.

Trace clustering: Using the KMEANS (with Euclidean distance) algorithm on the AMC 2 dataset, we obtained two clusters. For only one cluster the resulting process model was usable. When applying SOM (with Euclidian distance), five clusters were obtained. However, four clusters contained only very few event classes (1, 1, 3 and 7). In addition, these clusters also represented only a small portion of the PIs (lower than 60 compared to 682 in the complete event log). Therefore, SOM can be used to create clusters with PIs that have only very few different event classes (and accordingly have similar process models). These clusters can be of potential interest to a researcher who is looking for the exceptions in the event log. Subsequently, the largest cluster pertained to 520 PIs. Accordingly, the number of event classes was much higher (28 for the unfiltered set and 21 for the filtered set). The resulting process model was usable, but it was not significantly better (or worse) than the process models obtained with the patterns *Enhanced event log filter* and *HeuristicsMiner*. This accounts for both the global process models, as well as the clusters resulting from *Clustering (Excel)*. In addition, for the Italian data, SOM clustering did not result in any improvements compared to the previous process mining patterns.

Conclusion: As expected, when the HM is used on the raw event logs of both the AMC and Italian hospitals, the resulting process model is very complex and not usable. However, in this section we showed that with the sequential application of the process mining best practice patterns it is possible to create process models that have a good fitness and are much more understandable. As a result, we can state the following:

- *Renaming of events* has a significant influence on the results. This is both true for renaming laboratory events, as well as renaming all events. However, in the former case it is possible that still a large number of event classes remain, resulting in a complex process model nevertheless. In the latter case, details on the actual events is lost (e.g. for the AMC analysis it was only known which originator executed it).

- *Aggregation of events* in itself did not result in better process models and therefore we can conclude that it is no use to solely apply this pattern. However, in combination with the other patterns (such as *Enhanced event log filter* and *Add artificial start/end task*) the result can be improved (and it can still also decrease the variation in the event log).

- *Clustering* of process instances in itself did not result to process models with significant improvements. However, after applying additional patterns (such as *Enhanced event log filter* and *Add artificial start/end task*) results significantly improvement for several clusters. As we have learned, it is possible to extract clusters of patients with similar DBC-codes or create event logs for a specific hospital, which lead to usable process models. These clusters can be used to compare different groups of patients or hospitals which would not have been possible when the clustering pattern would not have been applied.

- As became obvious, *adding artificial start and end tasks* to an event log significantly increases the fitness of the process model (on all tested datasets). In addition, the resulting process models were much more understandable since a clear start and end could now be identified. This is especially convenient in larger process models. However, this does not necessarily result in usable process models (as there could still be a lot of event classes left).

- The *Enhanced event log filter* significantly decreases the number of event classes in an event log (sometimes only after both filters have been applied) and thereby increases the understandability. However, it does not necessarily increase the fitness of the model (but it does not decrease it by much either) after the addition of the *artificial start and end task*. A possible drawback is the loss of information in the process model. Furthermore, the percentage that is specified greatly depends on the interests and knowledge of the researcher.

- The *HeuristicsMiner* approach that was suggested by Weijters indeed proved to be a good alternative to the other best practice patterns and produced results that exceeded results of mining on a raw event log with default settings. The results are not necessarily better or worse than the results that are obtained with the other best practices. However, it should be noted that this approach is less convenient for larger event logs, as it may require much manual effort when filtering for event classes. Therefore, both approaches should be considered as complements and should both be considered by any researcher.

- *Trace clustering* did not produce the results that we had initially aimed for and expected. As we have learned, the clustering results are especially useful to identify groups of patients that have undergone very few events and have very similar processes. Process models that have resulted from the clusters with more event classes are not necessarily better or worse than the process models that are mined from the global event log that was used as an input for trace clustering. Furthermore, it cannot always be determined what the exact reason for clustering was (i.e. the patients do not necessarily have similar data characteristics). As a result, this pattern could serve as additional value but should not be considered as the only pattern to obtain a better model.

From the previous set of conclusions we learn that it is indeed not possible to create a set of “golden” best practices that always lead to the best result. In some cases, applying one pattern might lead to a better result than in other cases and the usefulness of the best practices and the results are highly dependent on the interests of the researcher. However, in this section we have showed that applying the process mining best practices indeed leads to better results than default mining on raw data. These

best practices should therefore always be considered in their own right and their usefulness depends on the knowledge and intentions of the researcher. Furthermore, actions such as skipping and/or iteration should always be considered by any user. Lastly, the next section presents the final details on the best practices for process mining in healthcare.

	Rename laboratory events	Rename all event classes	Aggregate laboratory events	Aggregate all event classes	Enhanced event log filter	Add a rtificial start/end task	HM approach (positive observations)	Extra Behavior Punishment	# Process instances	# Event classes	# Events	Is it understandable?	Is it usable?				
Raw																	
AMC global														0,48	682	417	43.615
Renaming of events (Excel)																	
AMC global	X													0,49	682	294	43.615
AMC global	X	X												0,63	682	28	43.615
Aggregation of events (Excel)																	
AMC global	X		X											-0,04	682	294	20.778
AMC global	X	X	X	X										-0,35	682	28	13.129
Clustering (Excel)																	
AMC M11	X		X											0,52	134	111	4.255
AMC M11	X	X	X	X										0,31	134	18	2.452
AMC M12	X		X											0,41	8	58	168
AMC M12	X	X	X	X										0,15	8	14	118
AMC M13	X		X											0,04	296	215	10.172
AMC M13	X	X	X	X										-0,46	296	25	6.593
AMC M14	X		X											0,2	93	131	2.296
AMC M14	X	X	X	X										-0,32	93	18	1.433
AMC M15	X		X											0,55	12	48	223
AMC M15	X	X	X	X										0,49	12	10	144
AMC M16	X		X											-0,07	139	119	3.658
AMC M16	X	X	X	X										0,06	139	19	2.383
Add artificial start/end task																	
AMC global	X		X			X								0,64	682	294	20.778
AMC global	X	X	X	X		X								0,66	682	28	13.129
AMC M13	X		X			X								0,51	296	217	
AMC M13	X	X	X	X		X								-0,09	296	27	
AMC M16	X		X			X								0,56	139	121	
AMC M16	X	X	X	X		X								0,51	139	19	2661
Enhanced event log filter																	
AMC global	X		X		X	X								0,63	682	111	
AMC global	X		X		XX	X								0,78	682	17	
AMC global	X	X	X	X	X	X								0,66	682	21	
AMC M13	X		X		X	X								0,51	296	116	10.019
AMC M13	X		X		XX	X								0,62	296	19	8.544
AMC M13	X	X	X	X	X	X								-0,1	296	24	
AMC M13	X	X	X	X	XX	X								0,1	296	13	
AMC M16	X		X		X	X								0,62	139	87	
AMC M16	X		X		XX	X								0,79	139	15	
AMC M16	X	X	X	X	X	X								0,5	139	17	
HeuristicsMiner																	
AMC global	X	X	X	X		X	1							0,34	682	18	
AMC global	X	X	X	X		X	10							0,71	682	18	
AMC global	X	X	X	X	X	X	1							0,62	682	14	
AMC global	X	X	X	X	X	X	10							0,3	682	14	
AMC M16	X	X	X	X		X	1							0,33	139	14	
AMC M16	X	X	X	X		X	10							0,51	139	14	
AMC M16	X	X	X	X	X	X	1							0,32	139	14	
AMC M16	X	X	X	X	X	X	10							0,51	139	14	
Trace clustering (KMEANS)																	
AMC global cluster 0	X	X	X	X	X	X								-0,2	172	24	
AMC global cluster 1	X	X	X	X	X	X								0,54	452	16	
Trace clustering (SOM)																	
AMC global cluster 0	X	X	X	X										-0,54	26	7	104
AMC global cluster 0	X	X	X	X		X								0,68	26	7	156
AMC global cluster 0	X	X	X	X	X	X								0,68	26	7	156
AMC global cluster 1	X	X	X	X										-0,32	520	28	12.776
AMC global cluster 1	X	X	X	X		X								0,69	520	28	13.816
AMC global cluster 1	X	X	X	X	X	X								0,69	520	21	13.789
AMC global cluster 2	X	X	X	X										-0,11	35	3	105
AMC global cluster 2	X	X	X	X		X								0,96	35	3	175
AMC global cluster 2	X	X	X	X	X	X								0,96	35	3	175
AMC global cluster 3	X	X	X	X										-1	43	1	86
AMC global cluster 3	X	X	X	X		X								0,98	43	1	172
AMC global cluster 3	X	X	X	X	X	X								0,98	43	1	172
AMC global cluster 4	X	X	X	X										0,98	58	1	58
AMC global cluster 4	X	X	X	X		X								0,98	58	1	174
AMC global cluster 4	X	X	X	X	X	X								0,98	58	1	174

Table 6.1 (left): The results of the best practices on the AMC dataset.

	Aggregate all event classes	Enhanced event log filter	Add artificial start/end task	HM approach (positive observations)	Extra Behavior Punishment	# Process instances	# Events classes	# Events	Is it understandable?	Is it usable?					
Raw															
Italy global											-0,31	369	95	6.484	
Data transformation (Excel)															
Italy global											-1,17	369	45	2.838	
Aggregation of events (Excel)															
Italy global	X										-1,54	369	45	2.475	
Clustering (Excel)															
Hospital 1	X										-0,52	95	34	876	
Hospital 2	X										-0,7	73	29	417	
Hospital 3	X										-0,6	101	31	785	
Hospital 4	X										-1,43	96	24	307	
Add artificial start/end task															
Italy global	X		X								0,49	369	45	3.213	
Hospital 1	X		X								0,13	95	36	1.066	
Hospital 2	X		X								0,5	73	31	563	
Hospital 3	X		X								0,48	101	33	987	
Hospital 4	X		X								0,64	96	26	499	
Enhanced event log filter															
Italy global	X	X	X								0,46	369	37	3.193	
Italy global	X	XX	X								0,47	369	19	2.992	
Hospital 1	X	X	X								0,13	95	36	1.066	
Hospital 1	X	XX	X								0,07	95	16	1.000	
Hospital 2	X	X	X								0,49	73	31	563	
Hospital 2	X	XX	X								0,49	73	19	543	
Hospital 3	X	X	X								0,5	101	26	980	
Hospital 3	X	XX	X								0,48	101	22	967	
Hospital 4	X	X	X								0,64	96	26	499	
Hospital 4	X	XX	X								0,63	96	14	477	
HeuristicsMiner															
Italy global	X		X	1							-0,31	369	16		
Italy global	X		X	10							0,56	369	16		
Trace clustering															
Hospital 1 cluster 24	X										-0,76	24	28	172	
Hospital 1 cluster 24	X		X								-0,22	24	30	224	
Hospital 1 cluster 24	X	X	X								-0,22	24	30	224	
Hospital 1 cluster 24	X	XX	X								-0,28	24	22	216	
Hospital 1 cluster 60	X										-0,4	60	30	619	
Hospital 1 cluster 60	X		X								0,44	60	32	739	
Hospital 1 cluster 60	X	X	X								0,44	60	32	739	
Hospital 1 cluster 60	X	XX	X								0,22	60	15	698	
Italy global cluster 347	X										-1,51	347	45	2453	
Italy global cluster 347	X		X								0,5	347	47	3.147	
Italy global cluster 347	X	X	X								0,47	347	37	3.127	
Italy global cluster 347	X	XX	X								0,47	347	17	2926	

Table 6.2 (above): The results of the best practices on the Italian dataset.

6.4 Best practices for process mining in healthcare

1. **Data collection:** For the sake of convenience, it is recommended to collect data in a file format that is easily loaded into MS Excel (such as a .csv file format). The data collection pattern specifies the required data attributes. For healthcare data, DBC-code information is a valuable and essential attribute and should therefore be included as well.

2. **Data transformation (Excel):** In MS Excel it is straightforward to delete insignificant information from the data (keep in mind the specified data objects). Additionally, it is recommended to check for completion of the data and homogeneity of the format of each data attribute.

3. **Renaming of events (Excel):** Keep in mind that this pattern exists to reduce the number of event classes and could serve as a first step for the aggregation of events. Renaming is especially important in healthcare data for all laboratory tasks (or any other events that are not of interest for the main behaviour in the process). In such a case it is recommended to rename the events to the originator that has executed the event. When the resulting process model is still complex and characterized by many different event classes, it could be considered to rename additional event classes as well (up to all). However, this is highly dependent on the researcher's interest and knowledge and actions as such will decrease the level of detail on the events in the process model.

4. **Aggregation of events (Excel):** It is recommended to aggregate the renamed laboratory tasks in the dataset per patient per day. In addition, when many repeating events occur, it could be considered to apply aggregation for all event classes (i.e. per patient per day) or just a selection of event classes. Again, this highly depends on the researcher's interest and knowledge.

5. **Clustering (Excel):** This pattern is perhaps more useful during the later stages of the analysis, when the researcher is more familiar with the data and the final goals. Clustering in Excel shows great potential when clustering is applied to DBC-codes. Creating more homogenous groups of patients based on these data attributes is likely to result in more understandable process models.

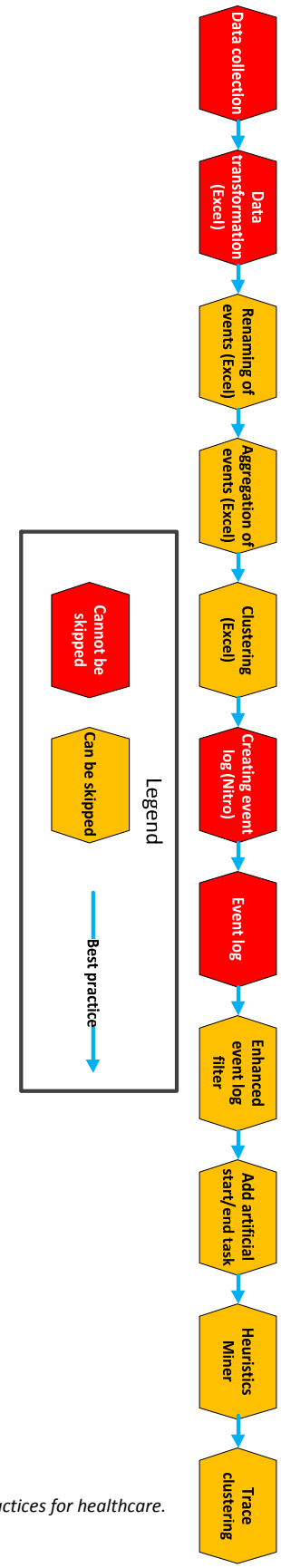


Figure 6.2: Pattern network of the process mining best practices for healthcare.

6. **Creating event log (Nitro):** Using the Nitro tool to create an event log in the .MXML (or .XES for ProM 6.0 and higher) format is simple and straightforward and allows for a first inspection of the event log.

7. **Event log:** This pattern forms the basis for the analysis in ProM and does not require any additional explanation.

8. **Enhanced event log filter:** Despite the renaming and aggregation of events, it is possible there still remains a large set of different event classes. One possible method to reduce the number of event classes is to filter the less frequent classes. The Enhanced event log filter is simple and effective. Using a 1% limit proved to be effective during the development, but it can be considered to increase or decrease whenever too few or too many events are filtered. This varies per dataset and perhaps requires several iterations to get it right.

9. **Add artificial start/end task:** Using the artificial start and end task allows anyone to quickly observe the start and end of a process model. In addition, it greatly improves the fitness measure of the model. It is recommended to always use this pattern before using the next pattern.

10. **Heuristics Miner:** When the previous patterns have been executed, it can be considered to apply the default HM approach. This approach will most likely result in an understandable and usable process model. In addition, the researcher may change the parameters intuitively to obtain a better process model. Furthermore, it is also possible to use the approach that is suggested by Weijters. Since the two approaches are not mutual exclusive, both produce good results and are easily performed, it can be considered to try both.

11. **Trace clustering:** In addition to manual clustering it can be decided to use automated clustering in ProM. However, trace clustering returns mixed results and the researcher should not rely on the results of trace clustering only. Therefore, trace clustering should be regarded as an additional technique to improve the process model when all other patterns have been applied.

Final note: Up till now, the best practices have been applied linear and straightforward (see figure 6.2 for the pattern network). However, the researcher can conclude that despite the best efforts, no sensible or good process model is obtained. In such a case, it is possible to return to a previous pattern and execute it again with a different approach, taking into account the newly discovered knowledge. For instance, it can be decided to perform some additional renaming and aggregation of events since there were too many event classes left in the process model after the first mining run. This highlights the iterative and complex nature of process mining. As a result, it is not possible to provide a “golden” set of best practices that are applicable exactly as they are stated to all cases. In the end, it is up to the researcher to decide which best practices to apply and how to apply them in the analysis. However, in this thesis we have showed that applying the best practices greatly improves the chance of producing usable process models and therefore they can greatly aid researchers who are new to the field of process mining. Once they have gained additional experience they can decide to use a different mining approach.

7. Reproducing results of previous process mining research

In addition to the validation of the best practices by process mining experts which is described in chapter 8, we have also attempted to reproduce the results of some of the preceding process mining studies (refer to chapter 4 for an overview of these studies) with the newly developed patterns. This part of the research served a twofold purpose.

First, by using the newly developed patterns we aimed to be able to reproduce the results of previous process mining studies. By comparing the process models that have been created with the patterns (with a focus on the best practice patterns) to the process models created by the original researcher, we aimed to show that the patterns indeed lead to results that are similar or better than the original methodology. Such a result would strengthen the validity of the patterns and best practices as at least an alternative and possibly better process mining methodology.

Second, by creating virtual machines in SHARE, we facilitated the possibility for the readers of this thesis to replicate our and the original results themselves (using the pattern language). In addition, we have created a set of screencasts that show the exact methodology that was used during our attempt to replicate the original results. This allows users of the VM to follow and replicate our steps exactly. Therefore, as an additional benefit, these VMs serve as an introduction and example to other researchers on the possibilities of VMs and screencasts as a means to communicate process mining research and results.

In total we have reproduced four cases in three VMs. The first VM pertains to the AMC data and was used to reproduce the results of Mans *et al.* [2009] and Ramos [2009]. The second VM focuses on the GGzE data and the results of Zanden [2010]. In the third VM we aimed to reproduce the results of Gupta [2007].

For the sake of brevity, only a summary of the methodology and results is provided in this chapter and we refer to Appendix G for the full description on these matters, as well as the process models. Accordingly, all links to the screencasts and VMs are provided in Appendix G and on the dedicated website that was developed for this thesis.

7.1 Reproducing the results of Mans *et al.* [2009] and Ramos [2009]

In our attempt to reproduce the results by Mans *et al.* [2009] and Ramos [2009] we have used a variety of patterns: Export .MXML, Data transformation (Excel), Renaming of events (Excel), Aggregation of events (Excel), Creating event log (Nitro), Add artificial start/end task, HeuristicsMiner, Trace clustering and Clustering (Excel). In total, three results were reproduced and compared.

First, we compared the process model of the complete event log that was produced with the pattern language (figure G.1) to the original model by Mans *et al.* (figure G.2). As can be observed, the new process model is slightly smaller and is therefore easier to understand than the original model. These differences are most likely caused by variations in the pre-processing phase. To illustrate, we have renamed all events to the originator that have executed them (as was recommended by Ramos). However, Mans *et al.* have only renamed a limited number of event classes (e.g. laboratory tasks). Due

to the lack of details in the report of Mans *et al.* we were unable to copy the exact pre-processing methodology.

Second, we considered process models for the largest cluster of patients in the global event log (obtained with trace clustering). In this case, the new process model (figure G.3) was more complex compared to the original models by Mans *et al*. (figure G.4) and Ramos (figure G.5). Again, these variations are most likely the result of differences between the pre-processing steps.

Third, process models for a cluster of patients with the DBC-code M16 were considered. The process model obtained with the patterns (figure G.6) was slightly more complex than the model that was created by Ramos (figure G.7). However, both models were understandable. Moreover, the fitness measure of the former model (0,90) was slightly higher than the fitness measure of the latter (0,84).

To conclude, we have showed that the newly developed patterns provide the possibility to create process mining results that are similar to the results created with previously used methodologies. The differences between the results can be contributed to lack of details on the original methodologies which inhibit the possibility to follow the exact same pre-processing steps in a new mining study.

7.2 Reproducing the results of Zanden [2010]

In the case of the GGzE data [Zanden, 2009], two process mining results were compared and we have used the following patterns to create the new process models: Data transformation (Excel), Clustering (Excel), Aggregation of events (Excel), Creating event log (Nitro), HeuristicsMiner.

First, we considered the level 3 activities (i.e. events with the lowest level of aggregation). However, both the new (figure G.8) and original process model (figure G.9) were very large and complex. This is not surprising when we consider the large number of event classes (57) in the event log.

Mining on the event log with the level 1 activities (i.e. highest level of aggregation) proved to be more successful. Since there were only 8 event classes in the event log, both the new (figure G.10) as well as the original process model (figure G.11) were very small and understandable. Despite several differences between frequencies of events and dependencies, the fitness measures were nearly identical: Proper completion and StopSemantics of 0,0 for both models, ContinuousSemantics of 0,359 for both models, ImprovedContinuousSemantics of 0,583 (Zanden) and 0,597 (new), and ExtraBehaviourPunishment of 0,559 (Zanden) and 0,572 (new).

To conclude, for the lowest level of aggregation (level 3 activities), the process models are complex regardless of the methodology (original or pattern language). Considering the large number of event classes this does not come as a surprise. Additional pre-processing steps such as the *Enhanced event log filter* could have resulted in better process models but these options have unfortunately not been used by the original researcher (in chapter 8 we demonstrate the benefit of filtering in the GGzE event log). However, for the level 1 activities, the resulting process models are more easily compared. Indeed, the process models differ slightly, but these differences are partly explained by the incomplete methodology of the original research which inhibited us from using the exact same pre-processing steps. However, there are many similarities between the two process models and all fitness measures are nearly

identical. Therefore, these results show that with the use of the process mining patterns it is indeed possible to replicate the results of previous mining methodologies.

7.3 Reproducing the results of Gupta [2007]

The case by Gupta proved to be more difficult to reproduce because several plug-ins were used for which we had not developed any patterns (i.e. *DWS miner* and *ARM*, see section 6.1). However, part of the reason why we have not developed a pattern for these plug-ins will become obvious in this section as, despite the well documented approach, we were unable to replicate most of the interesting alleged results.

The first part of the research focuses on the *HeuristicsMiner* only (and artificial start/end tasks). Using the HM pattern we obtained process models that were comparable (or better) in size and fitness to the results of Gupta. However, the most interesting part of the research by Gupta was not on the HM.

The second part of the research focuses briefly on the DWS miner. However, despite the fact that an exact similar approach was used, we were unable to reproduce the alleged results in any possible way. As a result, we were unable to obtain smaller and simpler process models. In addition, Gupta does not rate the DWS miner as a significant improvement of the HM either.

The focal point of the last part of the research by Gupta was the ARM. However, again we were unable to replicate most of the results. For instance, results by Gupta showed far more association rules than our results. In addition, clustering based on association rules did not improve the resulting process models significantly in our tests. Therefore, we could not perceive the ARM as additional value to the process mining methodology.

To conclude, for both the DWS miner and the ARM we were unable to replicate the alleged results in any way. This result does not influence the process mining patterns as they have not been applied to a large extend during the recreation of the results by Gupta. However, it does show that despite a well documented approach, it is difficult for other researchers to replicate alleged process mining results. This again highlights the importance of the patterns and additional tools such as VMs and screencasts as method to communicate research methodologies.

7.4 Conclusion

In this chapter we have demonstrated a new possible way to communicate process mining research by the use of virtual machines, screencasts and process mining patterns. In addition, we have showed that the patterns do indeed lead to process models that are similar to process models created with conventional process mining methodologies. This strengthens the findings that we have presented in earlier chapters. Furthermore, the research by Gupta shows that even with a well documented methodology the reproducibility of research can still be low. This again highlights the need for a new methodology to report process mining research and results.

8. Validation of the best practices

During the development of the process mining patterns in the previous chapters, the quality of a process model was either judged by its fitness measure and layout and/or it was compared to existing process models. However, a high fitness measure implies that the process model represents the behaviour in the event log well, but it does not necessarily mean that the resulting process model is also usable by process experts in the field of healthcare. During the development of the patterns, no explicit knowledge on the datasets and process was available to judge the true usability of the process models. In short, we have demonstrated that the (best practice) patterns can lead to process models that are smaller, less complex and achieve a higher fitness than conventional methodologies, but the usability of the resulting process models had yet to be determined.

To increase the validity of the process mining patterns and best practices we conducted two additional case studies at the Atrium MC and the GGzE. These healthcare institutions have been the source of data for Riemers [2009] and Zanden [2010] in previous process mining research. By involving healthcare experts from these institutions, with knowledge on the data and the actual processes, we aimed to validate the usability of the process models and the findings of the preceding chapters.

For the validation of the process mining patterns and best practices we originally aimed at two goals. First, we aimed to replicate the original results with the process mining patterns, which are similar approaches as presented in chapter 7. In contrast, in this chapter the process models would be compared and judged by process experts. Second, we aimed to compare the process models obtained with the best practices for process mining in healthcare with the process models created by the original researcher. In the latter case, our goal was to create process models that exceed the quality of the original research, thereby confirming the superiority of the process mining patterns and best practices.

8.1 Validation of the best practice results at Atrium

For the Atrium case we used the process models by Riemers [2009] as a reference during the validation stage. In addition, dr. Nico van Weert (manager quality and safety) and Pieter Plateel (division director) provided their expert opinion on the process models on behalf of the Atrium.

Initially we planned to follow the original goals of the validation that were stated earlier in this chapter. However, due to unexpected turns of events and prolonged discussion on a number of process models, we did not have a chance to present and discuss the results of the best practices. However, still some interesting results were obtained that are discussed first. Secondly, we present the results of the process models that were judged by the Atrium.

First, the general belief of the Atrium is that process mining and process models have yet to prove their worth, as the current results have not provided them with significant additional information. For instance, especially the placement of the events in the process models was very distracting and non-logical to them. As a result of this particular placement, the illusion is created that certain events are executed earlier in the process than reality (this is due to the unstructured nature of the placement of events by the HM algorithm).

In addition, the Atrium indicates that the whole process analysis should be more interactive. To illustrate, the process models for the validation had been created before the presentation of the results to the Atrium. Therefore, an explanation on the methodology that was followed during the analysis was necessary. Accordingly, Atrium indicated that they preferred to be involved in the decision making process during the analysis of the data. However, due to the limited time that was available during the presentation it was not possible to take the feedback into account directly. Atrium agreed on the fact that combined knowledge of the process (data) and process mining would allow for a researcher to create the required process models that the Atrium would be interested in. As a result, the comments by the Atrium highlight the importance of the goals of this thesis.

Secondly, we have created several process models with the pattern language. As was explained, we only presented the results of the replication of the models created by Riemers. During this replication we used the following patterns: Data collection (Excel), Renaming of events (Excel), Aggregation of events (Excel), Clustering (Excel), Creating event log (Nitro), Event log, Enhanced event log filter, Add artificial start/end task and HeuristicsMiner. For the sake of brevity we only present the results of the validation and the reader is referred to Appendix H for the complete details on the methodology and the resulting process models that have been presented to the experts at Atrium.

Result (1): Mining on the raw event log of *conservatief poliklinisch* did not result in a process model (figure H.1) that was of any use to the Atrium as it was far too complex. This judgement is similar to the judgement that was given for the process model by Riemers (figure H.2).

Result (2): The filtered event log of *conservatief poliklinisch* (events with frequency of less than 1% were discarded) proved to be more usable and the resulting process model (figure H.3) raised some interesting questions for the Atrium. It was noted that the layout of the original process model (figure H.4) was better (which is however the result of the algorithm itself) and therefore easier to understand. However, the process model created with the process mining patterns was judged as slightly more correct than the original model.

Result (3): Trace clustering resulted in several clusters of PIs. The process model that was created for the largest cluster (figure H.5) was not judged as usable by the Atrium. In his report, Riemers has mentioned similar results (although no models or fitness measures are provided).

Result (4) The process model that was obtained for the raw *enkelvoudig poliklinisch conservatief* event log (figure H.6) was judged as not usable at all. However, a similar result is mentioned in the report by Riemers (although no models or fitness measures are provided).

Result (5) The process model (figure H.7) obtained after filtering for event classes that occurred in less than 1% of event PIs showed the most potential. The model was judged as correct but it was noted that it had mixed two different processes. In addition, they expected that the events in the processes would be placed linear instead of parallel. However, this result may be contributed to the timestamp that only indicated the day on which an event was executed.

As can be learned from the results and comments, most process models created with the patterns were not judged as a significant improvement of the original process models (with two exceptions). However, most results were at least comparable to the original research by Riemers. In addition, the Atrium questioned the usability of process mining and models in general as they did not provide the information that they desired. Moreover, the Atrium pleaded for a more interactive approach for process mining, integrating more process knowledge in the analysis.

8.2 Validation of the best practice results at GGzE

In the case of the GGzE data we used the process models that were created by Zanden [2010] as a reference for the validation of the newly created process models with the pattern language. Tom Joosten (senior researcher) was involved as the process expert on behalf of the GGzE (and was also one of the supervisors of Zanden) and would provide his expert opinion on the process mining results (subsequently, all statements in this section have been verified by Joosten before they were included).

The process models for the first part of the validation had already been created during the recreation of the GGzE case in chapter 7. As a result, it was only necessary to create the process models according to the process mining best practices that served as the second part of the validation. For a complete overview of the methodology that was used during the creation of the new process models, as well as the models themselves, we refer to Appendix H.

First, we presented our attempt to replicate the original results by Zanden. Indeed, for the event log that contained the level 3 activities (lowest level of aggregation), no usable process model (figure G.8) was extracted as it was far too large and complex (i.e. 57 event classes). It was judged that this result was very similar to the result by the original researcher (figure G.9).

Subsequently, the process model (figure G.10) for the level 1 activities (highest level of aggregation) was presented. Although this model was much more understandable due to the low amount of event classes (8), the level of abstraction was too high to present any interesting details. Again this result was similar to the original research (figure G.11).

Before we started the validation process of the best practices, we asked Joosten what parts of the dataset would be most interesting for GGzE. This would provide us with a direction for clustering of the data. As a result, he indicated that the GGzE was interested in the differences between the different programs in the diagnoses *Pervasieve ontwikkelingsstoornissen* and *Aandachtstekortstoornissen en gedragsstoornissen*. In addition, Joosten indicated that he recognized the potential of process mining but that the present results (by Zanden) provided too little detail as the level of aggregation was too high (i.e. level 1 activities). Therefore, it was requested to keep aggregation low and try to use the level 3 activities, as these would provide the most detailed information on the process.

In order to improve the results of the original research we applied the following best practices to the data: Data transformation (Excel), Clustering (Excel), Creating event log (Nitro), Event log, Enhanced event log filter, Add artificial start/end task, HeuristicsMiner. We used the *Clustering (Excel)* pattern to create datasets for the different diagnoses that Joosten had indicated. Another addition to the original

result was the application of the *Enhanced event log filter* and the *Add artificial start/end task*. Since there were only very few double events we decided not to apply aggregation of these double events. In addition it was decided not to perform any renaming in order to maintain the lowest level of aggregation. Subsequently, the application of the best practices led to the following result:

Result 1: For the global event log of *Pervasieve ontwikkelingsstoornissen* it was mentioned that the resulting process model (figure H.8) was a significant improvement compared to the results that have previously been discussed. Especially the addition of the artificial start and end tasks made the process model much more comprehensible. However, the global event log contained many different diagnoses and therefore it was difficult to extract any usable information from the process model.

Result 2: For the subset *Pervasieve ontwikkelingsstoornissen NAO* the process model (figure H.9) received much positive acclaim. Joosten indicated that the model was very recognizable and that it presented the real situation (at the time of which the data originates) to a large extend. In addition, a few remarkable situations were discovered that comply with the situation at the time of the data.

Result 3: The process model that was created for the subset *Gecombineerde type* (figure H.10) also received a positive evaluation. Joosten indicated that the result was very recognizable and he could clearly identify two different parts of the process: diagnostics and treatments. The process model served as a clear confirmation of the present perceptions of the real process flow.

For the GGzE, the best practices lead to a significant improvement in process models compared to the previous process mining results. By making use of process knowledge (know which parts to focus on) and the best practices we were able to create process models that reflected reality to a high degree and that served as a confirmation of the layout of the real process. As a result, Joosten was interested in the results of the thesis and confirmed that process mining could indeed be of added value for GGzE.

8.3 Conclusion

In this chapter we aimed to validate the added value of the process mining patterns and best practices by involving process mining experts to judge the resulting process models. For the Atrium case we showed that the patterns could be used to create results that are similar to or slightly better than the results achieved with conventional process mining methodology. Due to circumstances the pattern language could not be fully validated. However, the Atrium agreed that a combination of process and process mining knowledge would yield a higher potential for process models. This highlights the importance of the goals of this thesis. As a result, it should be considered to redo the Atrium case, involving more process knowledge during the creation of the process models to test the full potential of the patterns and best practices.

With the addition of some process knowledge we were able to show that the best practices did lead to a significant increase in results for the GGzE case. The resulting process models reflected the real processes to a large extend and the different paths through the process were clearly visible. As a result, the GGzE was interested in the additional application of process mining and the results of this thesis. Subsequently, the validation at GGzE shows the added value of the results and goals of this thesis.

9. Conclusions, limitations and recommendations

The goal of this thesis was to develop a set of process mining best practices, thereby contributing both to the accessibility and the reproducibility of process mining research in healthcare. In addition, we aimed to produce results that are better, or at least comparable, to the present process mining methodologies in healthcare. In this chapter we will discuss the main conclusions and limitations of this thesis and, based on these, make recommendations for future research. First, we reflect on the quality of the research methodology in section 9.1. The foundations of this research could be specified as *accessibility*, *reproducibility* and *improved results* in process mining research in healthcare. Therefore, we used these keywords as pillars for the conclusions, limitations and recommendations on the newly developed process mining pattern language and its associated best practices, SHARE virtual machines and screencasts that we have introduced in this thesis. These are presented in sections 9.2 to 9.4 respectively.

9.1 Reflection on the research methodology

In chapter 3 we have introduced a set of quality criteria that any scientific research methodology should meet: construct validity, internal validity, reproducibility and external validity. To assure the *construct validity* of this research, multiple sources of data have been considered for the solution design. Apart from a multiple case study, we have introduced process mining expert knowledge as an additional source of information. *Internal validity* was secured by introducing multiple datasets to develop and test the new methodology, an approach that has not been used in many process mining studies so far. To increase the *reproducibility*, a detailed record was kept of all activities during this project. In addition, all procedures and results are thoroughly explained in this thesis report and corresponding appendices. As reproducibility is one of the main problems that we have identified in process mining research, we aimed to set an example with this report. Finally, to assure the *external validity* of the results, two additional cases have been executed for the verification of the newly developed methodology. By involving process experts we secured the required knowledge to judge the results of the thesis.

9.2 Conclusions

***Accessibility*:** There is much complicated literature on the available process mining plug-ins, but these papers mostly focus on the underlying mathematical algorithms and results, and few papers actually explain how to use the plug-in. In addition, literature on the important pre-processing steps is nearly non-existing. As a result, the accessibility of process mining is low and new users have a hard time gathering all the required knowledge. Moreover, even experienced process mining researchers have difficulties identifying the most important process mining steps in the large pool of possibilities.

As a means to increase accessibility, we aimed to integrate the existing process mining methodologies into a single and complete methodology. Whereas previous process mining researchers have focused on one or two preceding studies only, by taking a helicopter view of the present process mining methodologies in healthcare we identified several important algorithms and steps that are common in a number of studies. This allowed us to obtain a broader and more complete overview of the current process mining possibilities. In addition to the written literature, interviews with process mining experts

assured the most up to date information on process mining in healthcare (something which has not been considered by many previous researchers). By integrating both the knowledge from the case studies and process mining experts we were able to create one “global” process mining methodology of the present process mining research in healthcare.

The concept of pattern language is introduced as a method to communicate the new methodology. Whereas present process mining literature focuses mostly on the underlying algorithms of process mining, the pattern language allows us to focus on the application of the corresponding plug-ins and techniques. Moreover, by creating one single template for all process mining patterns, users can get familiarized with the presentation of plug-ins and allows easy comparison of alternatives.

Based on the research steps in the present “global” methodology that was synthesized earlier, a set of 22 process mining patterns was created that represent these different steps, ranging from data collection to actual mining in ProM. The central repository of process mining patterns that is created in this thesis eliminates the need for researchers to gather information that is largely scattered throughout the internet and research reports. In addition, the dependencies of the different process mining patterns are easily visualized in the pattern network. As such, researchers can quickly identify preceding, following and alternative process mining steps, thereby placing each pattern into perspective.

As mentioned, for several steps in the methodology there are alternative patterns. Moreover, the level of detail in the patterns was kept low to increase external validity. Therefore, to make the patterns accessible in a healthcare setting we have marked a set of 11 patterns as best practices of process mining in healthcare. As a result, researchers in healthcare need only to consider these patterns to obtain their process models. Moreover, some additional information is provided for the best practices with regard to their application in a healthcare setting (e.g. what data objects to consider).

In addition, to increase accessibility we have introduced the concept of screencasts. We have created screencasts for several patterns in the repository that solely demonstrate the application of that specific pattern on a real dataset.

Furthermore, a virtual machine was created that allows readers to test the new pattern language themselves, without having to download and/or install any software and data. In this VM we provided all means that are necessary for testing such as: software, datasets and the complete pattern repository.

The pattern language and its best practices, the pattern network, screencasts and virtual machine have been combined into a single repository (i.e. website) that was created for this thesis project. By creating a central place that contains all required knowledge on process mining (in healthcare) we have reduced the requirement for researchers to spent much time and effort gathering scattered information on process mining research, which to the best of our knowledge is an unique approach in this field.

***Reproducibility*:** During this research we have showed that by referring to the process mining patterns, the need for extended explanations in the actual report can be limited and the pattern can communicate large amounts of information in a few words. This could greatly contribute especially to process mining research in published work, where the amount of space is limited. Readers can consult

the central repository whenever a pattern is mentioned and require additional information. This would allow researchers to communicate their complete methodology, thereby increasing its reproducibility.

In addition, screencasts can be used by researchers as an additional method to present their methodology and results as the screencasts give an exact representation of how the analysis has been conducted. Again, this could especially be of interest for published work where the lack of space forces researchers to omit important details on the methodology. Moreover, screencasts can be used to present additional results that could not be included in the research paper. In addition, we have learned that through visualization, introduction of new process mining concepts to the process mining community could be simplified (e.g. this would have been beneficial for the research by Gupta[2007]).

Furthermore, in this thesis we showed that a virtual machine can be used as a method to increase reproducibility, as it allows researchers to replicate the alleged results with exactly the same software and data without any additional actions from the reviewer's side. As with the screencasts, VMs could also be of particular interest for researchers that introduce new process mining plug-ins or techniques.

By integrating the pattern language, screencasts and virtual machines, communicating research methodologies and research results would be easier and complete, thereby increasing the reproducibility of process mining research. Especially for published research, which deals with limited space to present its complete methodology and all results, a combination as proposed in this thesis could be of added value. Moreover, researchers that introduce new process mining plug-ins or techniques could benefit from the increased reproducibility and accessibility.

***Improved results*:** In this thesis we have first demonstrated that the pattern language can be considered as an alternative to the current process mining methodologies as it is able to produce similar results. This was achieved by recreating previous process mining results with the newly developed pattern language. To increase validity and reproducibility of these results, virtual machines and screencasts have been created that demonstrate the methodology and results of the patterns and allow reviewers to replicate the results of this thesis.

Second, we tested the result of the process mining patterns and best practices on three datasets and indeed confirmed that sequential application of the best practices leads to increasingly more comprehensible process models with higher fitness measures and therefore had increased usability.

However, to increase the validity of the best practices, two additional case studies were performed that involved process experts who could judge the process models' true usability. The goals were to validate that 1) the patterns could serve as an alternative to the present methodologies and 2) that the best practices lead to process models that exceed the quality of the original research. Indeed for both cases it was confirmed that the patterns could serve as an alternative to the present methodologies. In the second case, the best practices produced process models which quality significantly exceeded the quality of the original process models.

As a result, we can conclude that the pattern language as a whole can indeed serve as an alternative to the current process mining methodologies as it able to produce similar results. In addition, the best

practices that are recommended in this thesis produce results that are of superior quality compared to the process models created with the conventional methodologies.

9.3 Limitations

***Accessibility*:** The alleged accessibility of the process mining patterns and best practices has yet to be determined. First, the general concept of the process mining pattern language needs to be assessed and tested by persons without prior knowledge on the patterns and preferably little knowledge on process mining in general. In addition, it is necessary to validate the accessibility of the accompanying best practices. Only when inexperienced process mining researchers are able to produce good results with the pattern language and best practices can we make valid conclusions on the claimed accessibility.

Furthermore, a process mining methodology is dependent on an almost infinite number of factors. For instance, the researcher's interests and availability of data (attributes) greatly influence the choices that can and should be made. Moreover, healthcare data is complex and comes in a wide variety of forms. As a result, it is not possible to create a "golden" set of patterns or best practices that can be blindly applied to any case. In the end, the application of the patterns still highly depends on the knowledge and intuition of the user.

In addition, we have introduced the concept of virtual machines and screencasts as a method to introduce researchers to the process mining patterns. However, as with the patterns themselves, it needs to be validated that the combination that is proposed in this thesis does indeed contribute to the accessibility of process mining.

Moreover, the current repository of process mining patterns is limited and only focuses on a very few process mining steps. For instance, it is only possible to use the patterns to create process models with the HeuristicsMiner at the present time. Considering the many additional process mining techniques and possibilities that are available, the current pattern language is far from complete. An additional problem is the constant development of the process mining algorithms and techniques. This requires willingness researchers who perform continuous maintenance and updates of the pattern repository.

***Reproducibility*:** By introducing the process mining patterns we aimed to facilitate in the reproducibility of process mining research. By referring to a certain pattern, a researcher could briefly indicate what process mining algorithms and techniques were used during the research. Throughout this thesis we have referred to the patterns numerous times when discussing the process mining analyses that have been conducted during this thesis project. However, as the patterns only provide general information on the goals and application of the pattern, researchers are still required to specify additional information on the application of the pattern. For instance, when the Aggregation of events (Excel) pattern is applied, a researcher would still need to specify what events exactly have been aggregated. As a result, the pattern language still calls upon additional information in the report and cannot solely be used by referring to a pattern.

Furthermore, it has to be validated that the patterns indeed make process mining research more reproducible by providing more detailed information on the research methodology. In addition, it

should be considered whether the patterns actually make reporting such a methodology easier from the presenting researcher's point of view.

We have introduced the concept of virtual machines as a means to facilitate the replication of alleged results. However, it is not validated that an approach as such does indeed lead to increased reproducibility or that it has a positive effect on the understandability of the research by its reviewers. As a result (and in fact for all three pillars), it should be considered whether the virtual machines are worth the additional effort. In addition, similar questions are raised for the use of the screencasts.

***Improved results*:** During this thesis we have demonstrated the added value of the process mining patterns and best practices, as they were able to create process models that exceed the quality of process models that are created with conventional process mining methodologies. However, during this thesis, the patterns were only used and tested by the persons who created the patterns. As a result, during the tests and validation stages, more knowledge was available on the process mining patterns and how to apply them than would be for new users. Therefore, the results of the patterns could be biased and the best practices might not lead to better results for users that are inexperienced with process mining and the process mining patterns/best practices. Subsequently, the claims regarding the improved results of the new methodology need to be validated.

In addition, new users might be fooled by the concept of a high fitness measure. As we have demonstrated in this thesis, a high fitness measure does not necessarily imply that the model is indeed usable by process mining experts or corresponds to reality, as the model is based solely on the information that is available, which could be incomplete or incorrect. Therefore, the results of process mining should still always be validated by persons with knowledge on the actual real process.

9.4 Recommendations

***Accessibility*:** First, the accessibility of the pattern language needs to be verified. This would require users with little knowledge on process mining and the ProM software to use the patterns (and best practices) in a new environment and try to create process models. When such inexperienced users are indeed able to produce usable process models we are able to confirm the added value of the patterns to the accessibility of process mining. Ideal candidates for validation are the process experts in healthcare, who possess the knowledge on the actual processes but lack the experience with the concepts of process mining and the accompanying software. The former is important as in this thesis we have demonstrated that some process knowledge is required by any researcher in any process mining research to make the right decisions.

As a means to combine future researchers' interests, virtual machines could serve as means to facilitate the previously discussed validation of the accessibility of the patterns. As a result, both the added value of the VMs and the process mining patterns can be studied.

In addition, the pattern repository is limited at the present time and therefore requires expansion. Currently, the pre-processing and mining patterns are highly focussed on producing process models with the HeuristicsMiner. However, ProM features a variety of algorithms that allow for many different

mining and analysis possibilities. Moreover, several plug-ins have been identified in this thesis which need additional research. Future researchers should focus on the development of new patterns to create a complete network of all process mining possibilities.

***Reproducibility*:** First, the pattern language should be validated as a means to increase the reproducibility of process mining research. One such an approach could be the following. Consider that a researcher performs a process mining analysis and uses the patterns as a means to report his methodology. Subsequently, a group of test subjects should be able to reproduce the alleged results by using the report of the original researcher and thereby confirming the reproducibility of the process mining patterns.

As with virtual machines that serve to increase accessibility, it needs to be validated that the additional effort of creating a set of VMs is worth the added value for reproducibility. Future researchers could combine the validation of the VMs with the validation of the reproducibility of the patterns themselves.

In addition, the pattern language facilitates reporting of process mining research in a more efficient and complete way. However, at the moment the format of the reports still depends on the different researchers and could therefore vary from case to case. Therefore, future researchers should focus on standardizing the way process mining research is reported. One such an approach would be by creating a standardized template, indicating exactly what parts of the analysis need to be reported and what additional information is required. Other process mining researchers could simply download such as template and fill it in, thereby removing the possibility of forgetting to report vital information. Moreover, standardizing reports could also add to the understandably and accessibility of process mining research, and it allows for easier comparison of different process mining studies.

If we take a complete different view on reproducibility we can conclude that the pattern language has solely been considered in the healthcare domain so far. However, as we have discussed, process mining is far from limited to the healthcare domain and applied in a variety of industries. Therefore, future researchers should focus on the applicability of the process mining pattern language and accompanying VMs and screencasts in different industries.

***Improved results*:** During this thesis we have validated the results of the patterns and the best practices in a number of cases. However, despite the fact that the results seem to be conclusive, additional research is needed to verify that the patterns and best practices do indeed lead to results that are better than current process mining methodologies. Validation as such could be achieved by applying the patterns and best practices in new and unrelated case studies.

In addition, as was discussed during the limitations of the research, it should be validated whether the best practices can be used by persons who are inexperienced with process mining (process experts in the field of healthcare would make excellent candidates). Moreover, it should be validated that when the best practices are used by a test group as such, they still lead to results that are of superior quality compared to the results of conventional process mining research methodologies. Only when such as result is achieved can we truly conclude that the best practices are of added value for process analysis in the field of healthcare.

10. References

Aalst, W. van der, Hofstede, A. ter, Weske, M. (2003). Business Process Management: A Survey. In Aalst, W.M.P. van der, Hofstede, A.H.M. ter, Weske, M. (Eds.) *Lecture Notes in Computer Science* 2678 (pp. 1-12). Springer-Verlag, Berlin, Germany.

Aalst, W.M.P. van der, Weijters, AJ.M.M., Maruster, L. (2004) Workflow Mining: Discovering process models from event logs. *IEEE Transactions on Knowledge and Data Engineering* 16(9), pp. 1128-1142.

Aalst, W.M.P. van der, Beer, H.T. de, Dongen, B.F. van (2005) Process Mining and Verification of Properties: An Approach based on Temporal Logic. *BETA Working Papter Series*, WP 136, University of Technology, Eindhoven, the Netherlands.

Aalst, W.M.P. van der, Weijters, AJ.M.M. (2005) Process Mining. In Dumas, M., Aalst, W.M.P. van der, Hofstede, A.H.M. ter (Eds.) *Process-Aware Information Systems: Bridging People and Software through Process Technology* (pp. 235-255). Wiley & Sons, Hoboken, New Jersey, USA.

Aalst, W.M.P. van der, Reijers, H.A., Song, M. (2005) Discovering Social Networks from Event Logs. *Computer Supported Cooperative Work* 14(6), pp. 549-593.

Aalst, W.M.P. van der, Reijers, H.A., Weijters, A.J.M.M., Dongen, B.F. van der, Alves de Medeiros, A.K., Song, M., Verbeek, H.M.W. (2007a) Business process mining: An industrial application. *Information Systems* 32, pp. 713-732.

Aalst, W.M.P. van der, Netjes, M., Reijers, H.A. (2007b) Chapter 4: Supporting the Full BPM Life-Cycle Using Process Mining and Intelligent Redesign. In Siau, K. (Ed.) *Contemporary Issues in Database Design and Information Systems Development* (pp. 100-132). IGI Global, Hershey, Pennsylvania, USA.

Aalst, W.M.P. van der, Dongen, B.F. van, Günther, C.W., Mans, R.S., Alves de Medeiros, A.K., Rozinat, A., Rubin, V., Song, M., Verbeek, H.M.W., Weijters, A.J.M.M. (2007c) ProM 4.0: Comprehensive Support for Real Process Analysis. In Kleijn J., Yakovlev, A. (Eds.) *Lecture Notes in Computer Science* 4546 (pp. 484-494). Springer-Verlag, Berlin, Germany.

Aalst, W.M.P. van der, Dongen, B.F. van, Günther, C., Rozinat, A., Verbeek, H. M. W., Weijters, A. J. M. M. (2009) ProM: The Process Mining Toolkit. *CEUR Workshop Proceedings* 489, pp. 1-4.

Aalst, W.M.P. van der (2011) Process mining: discovery, conformance and enhancement of business process. Springer-Verlag, Berlin, Germany.

Aken, J.E. van, Berends, H., Bij, H. van der (2007) Problem Solving in Organizations: A Methodological Handbook for Business Students. University Press, Cambridge, United Kingdom.

Alexander, C., Ishikawa, S., Silverstein, M. (1977) A Pattern Language. Oxford University Press, New York, United States.

Alexander, C. (1979) A Timeless Way of Building. Oxford University Press, New York, United States.

Alves de Medeiros, A.K. (2006) Genetic Process Mining (Doctoral dissertation). Beta Dissertation Series D-89, University of Technology, Eindhoven, the Netherlands.

Alves de Medeiros, A.K., Weijters, A.J.M.M. (2009) ProM Framework Tutorial. University of Technology, Eindhoven, the Netherlands.

Beer, H.T. de, Brand, P.C.W. van den (2007) The LTL Checker Plugins: a (reference) manual (Unpublished manual). University of Technology, Eindhoven, the Netherlands.

Cassell, C., Symon, G. (1994) Qualitative Methods in Organizational Research: A Practical Guide. Sage Publication, London, United Kingdom.

Chapman, P., Clinton, J., Kerber, R., Khabaza, T., Reinartz, T., Shearer, C., Wirth, R. (2000) CRISP-DM 1.0: Step-by-Step Data Mining Guide, SPSS Inc.

Davenport, T. (1993) Process Innovation: Reengineering work through information technology. Harvard Business School Press, Boston, USA.

Demeyer, S., Ducasse, S., Nierstratz, O. (2009) Object-Oriented Reengineering Patterns. Square Bracket Associates.

Dittner, R., Rule, D. (2007) The Best Damn Server Virtualization Book Period. Elsevier, Amsterdam, the Netherlands.

Dongen, B.F. van, Aalst, W.M.P. van der (2005) A meta model for process mining data. In Casto, J., Teniente, E. (Eds.) *Proceedings of the CAiSE Workshops* 2(2) (pp. 309-320). FEUP, Porto, Portugal.

Gamma, E., Helm, R., Johnson, R., Vlissides, J. (1995) Design Patterns: Elements of Reusable Object-Oriented Software. Addison-Wesley, Boston, Massachusetts, United States.

Gupta, S. (2007) Workflow and Process Mining in Healthcare (Unpublished master's thesis). Department of Mathematics and Computer Science, University of Technology, Eindhoven, the Netherlands.

Günther, C.W. (2009) Process Mining in Flexible Environments (Doctoral dissertation). Beta Dissertation Series D-117, University of Technology, Eindhoven, the Netherlands.

Günther, C.W., Aalst, W.M.P. van der (2007) Fuzzy Mining – Adaptive Process Simplification Based on Multi-Perspective Metrics. In Alsonso, G., Dadam, P., Rosemann, M. (Eds.) *Lecture Notes in Computer Science* 4714 (pp. 328-343). Springer-Verlag, Berlin, Germany.

Gorp, P.M.E. Van, Grefen, P. (2010) Supporting the internet-based evaluation of research software with cloud infrastructure. *Software & System Modeling* 1-18-18. Springer Berlin/Heidelberg, Germany.

Gorp, P.M.E. Van, Mazanek, S. (2011) SHARE: a web portal for creating and sharing executable research papers. In proceedings of the International Conference on Computational Science (Procedia Computer Science, Vol. ??, pp. ??-??, Elsevier), ICCS 2011, Tsukuba International Congress Center, Tsukuba Japan, June 1 - June 3, 2011.

Janssen, R.J.P. (2010) Process Mining in Healthcare: A Literature Review (Unpublished literature review). Department of Industrial Engineering & Innovation Sciences, University of Technology, Eindhoven, the Netherlands.

Janssen, R.J.P. (2011) Process Mining in Healthcare: A meta-analysis and validation for the derivation of best practices (Unpublished project proposal). Department of Industrial Engineering & Innovation Sciences, University of Technology, Eindhoven, the Netherlands.

Kindler, E. (2006) On the Semantics of EPCs: A Framework for Resolving the Vicious Circle. *Data & Knowledge Engineering*, 56(1), pp. 23–40.

Kohlbacher, M. (2010) The effects of process orientation: a literature review. *Business Process Management Journal* 16(1), pp.135 – 152.

Lassche, R. (2010) Care Pathway Analysis and Redesign: a Methodology (Unpublished master's thesis). Department of Industrial Engineering & Innovation Sciences, University of Technology, Eindhoven, the Netherlands.

Lenz, R., Reichert, M. (2007) IT support for healthcare processes – premises, challenges, perspectives. *Data & Knowledge Engineering* 61, pp. 39-58.

Mans, R.S. (20XX) User manual for converting data from a Microsoft Access Database to the ProM MXML format (Unpublished manual). University of Technology, Eindhoven, the Netherlands.

Mans, R.S., Schonenberg, M.H., Leonardi, G., Panzarasa, S., Cavallini, A., Quaglini, S., Aalst, W.M.P. van der (2008) Process mining techniques: An application to stroke care. *Studies in Health Technology and Informatics* 136, pp. 573-578.

Mans, R.S., Schonenberg, M.H., Song, M., Aalst, W.M.P. van der, Bakker, P.J.M. (2009) Application of Process Mining in Healthcare – A Case Study in a Dutch Hospital. *Biomedical Engineering Systems and Technologies: Communications in Computer and Information Science* 25(4), pp. 425-438.

Mans, R.S. (2011) Workflow support for the healthcare domain (Doctoral dissertation). University of Technology, Eindhoven, the Netherlands.

McCormack, K. (2001) Business Process Orientation: Do You Have It? *Quality Progress* 34(1), pp. 51-58.

Mulyar, N., Aalst, W.M.P. van der (2005a) Patterns in Colored Petri Nets. BETA Working Paper Series, WP 139, University of Technology, Eindhoven, the Netherlands.

Mulyar, N., Aalst, W.M.P. van der (2005b) Towards a Pattern Language for Colored Petri Nets. In Jensen, K. (Ed.) *Proceedings of the Sixth Workshop on the Practical Use of Coloured Petri Nets and CPN tools (CPN 2005)* 576 (pp. 39-38). University of Aarhus, Aarhus, Denmark.

Murata, T. (1989) Petri Nets: Properties, Analysis and Applications. *Proceedings of the IEEE* 77(4), pp. 541-580.

Noble, J., Johnson, R.A. (Eds.) (2010) Transactions on Pattern Languages of Programming I. *Lecture Notes in Computer Science* 5770. Springer-Verlag, Berlin, Germany.

Noble, J., Johnson, R.A., Harrison, N.B., Zdun, U. (Eds.) (2011) Transactions on Pattern Languages of Programming II. *Lecture Notes in Computer Science* 6510. Springer-Verlag, Berlin, Germany.

Pinowar, H.A., Day, R.S., Fridsma, D.B. (2007) Sharing Detailed Research Data Is Associated with Increased Citation Rate. PloS ONE 2(3), e308.

Rantham Prabhakara, J.C.B., Aalst, W.M.P. van der (2009). Abstractions in process mining: a taxonomy of patterns. In Dayal, U., Eder, J., Koehler, J., Reijers, H.A. (Eds.) *Lecture Notes in Computer Science* 5701 (pp. 159-175). Springer-Verlag, Berlin, Germany.

Ramos, L.T. (2009) Healthcare Process Analysis: validation and improvements of a data-based method using process mining and visual analytics (Unpublished master's thesis). Department of Industrial Engineering & Innovation Sciences, University of Technology, Eindhoven, the Netherlands.

Riemers, P. (2009) Process improvement in Healthcare: A data-based method using a combination of process mining and visual analytics (Unpublished master's thesis). Department of Industrial Engineering & Innovation Sciences, University of Technology, Eindhoven, the Netherlands.

Rozinat, A. (2010) Process Mining: Conformance and Extension (Doctoral dissertation). Beta Dissertation Series D-136, University of Technology, Eindhoven, the Netherlands.

Rusu, S. (2010) Discovery and Analysis of Field Service Engineer Process Using Process Mining (Unpublished master's thesis). Department of Mathematics and Computer Science, University of Technology, Eindhoven, the Netherlands.

Saunders, M.N.K., Lewis, P., Thornhill, A. (2000) Research Methods for Business Students (2nd edition). Pearson Education, Upper Saddle River, New Jersey, United States.

Shan, Y., Jeacocke, D., Murray, D.W., Sutinen, A. (2008) Mining medical specialist billing patterns for health service management. In Roddick, J.F., Li, J., Christen, P., & Kennedy, P.J. (Eds.) *Conferences in Research and Practice in Information Technology* 87, pp. 105-110.

Song, M., Aalst, W.M.P. van der (2007a) Towards Comprehensive Support for Organizational Mining. *BETA Working Paper Series*, WP 211, University of Technology, Eindhoven, the Netherlands.

Song, M., Aalst, W.M.P. van der (2007b) Supporting Process Mining by Showing Events at a Glance. In Chari, K., Kumar, A. (Eds) *Seventeenth Annual Workshop on Information Technologies and Systems* (pp. 139-145). Montreal, Canada.

Song, M.S., Gunther, C.W., Aalst, W.M.P. van der (2009) Trace Clustering in Process Mining. In Ardagna, D., Mecella, M., Yang, J. (Eds.) *Lecture Notes in Business Information Processing* 17(2) (pp. 109-120). Springer-Verlag, Berlin, Germany.

Swolfs, B. (2010) Using information about financial control for process control in healthcare (Unpublished master's thesis). Department of Industrial Engineering & Innovation Sciences, University of Technology, Eindhoven, the Netherlands.

Tracz, W. (1995) Confessions of a Used Program Salesman: Institutionalizing Software Reuse. Addison-Wesley, Boston, Massachusetts, United States.

Two Crows Corporation (2005) Introduction to Data Mining and Knowledge Discovery (third edition). Two Crows Corporation, Potomac, Maryland, United States of America.

Vandewalle, P., Kovacevic, Vetterli, M. (2009) Reproducible Research in Signal Processing – What, Why, and How? *IEEE Signal Processing Magazine* 26(3), pp. 37-47.

Veiga, G.M., Ferreira, D.R. (2009) Understanding Spaghetti Model with Sequence Clustering for ProM. *International Workshop on Business Process Intelligence 2009.*

Yang, W.S., Hwang, S.Y. (2006) A process-mining framework for the detection of healthcare fraud and abuse. *Expert Systems with Applications* 31, pp. 56-68.

Yin, R.K. (2009) Case Study Research: Design and Methods (fourth edition). SAGE Publications, London, United Kingdom.

Weijters, A.J.M.M., Aalst, W.M.P. van der, Alves de Medeiros, K. (2006) Process Mining with the HeuristicsMiner Algorithm. *BETA Working Paper Series,* WP 166, University of Technology, Eindhoven, the Netherlands.

Weijters, A.J.M.M., Ribeiro, J.T.S. (2010) Flexible Heuristics Miner. *BETA Working Paper Series*, WP 334, University of Technology, Eindhoven, the Netherlands.

Wilson, A., Zeithaml, V.A., Bitner, M.J., Gremler, D.D. (2008) Services Marketing: Integrating Customer Focus Across the Firm. McGraw-Hill Education, England.

Zanden, L. van der (2010) Process improvement in mental healthcare: A data-based method for care delivery process analysis in GGzE Centre child and adolescent psychiatry (Unpublished master's thesis). Department of Industrial Engineering & Innovation Sciences, University of Technology, Eindhoven, the Netherlands.

Appendix A. Data collection protocol

Introduction into the research: According to Yin [2009], a brief overview of a case study is considered mandatory for each data collection protocol. An introduction as such will serve to guide the reader through the analysis, stating the main goals and findings of the research. Accordingly, this should allow the reader understand the analysis more easily.

Information on the dataset: For each dataset, a number of important characteristics that can have an influence on the methodology and results have been analyzed. Most datasets contain a large amount of data attributes (up to 50 or more). However, for the case analysis we have chosen to focus on the most important and common aspects, being:

1. *DBC-code information*: The presence of DBC-codes could be used for the clustering of patients that have undergone similar diagnosis and/or treatments. Such homogenous clusters could result in improved process model quality.
2. *Number of hospitals*: As with DBC-code information, different hospitals in the dataset can also be used as selection criteria for the clustering of patients.
3. *Number of process instances*: The number of process instances reflects the number of patients in the dataset. An increased number of process instances results in an increase in events and a potential increase in variance in the process, thereby increasing process model complexity.
4. *Number of events*: As with 3, an increased number of events increases data complexity.
5. *Number of event classes*: An increased number of different event classes could increase process model complexity.
6. *Timestamp*: A timestamp can be used for correctly establishing the order of events in a process instance. When absent, in most situations it is difficult or impossible to determine the exact order of events. In addition, many formats of timestamps have been identified. For example, it is possible only the day on which an event occurred is known, or that we also know the exact time.
7. *Originators*: Originators are an interesting subject for renaming and/or clustering and come in various forms such as departments and/or actual persons.
8. *Event types*: An increased number of event types could contribute to the complexity of process models, as for instance a start and end event is present for each event (thereby doubling the number of event classes).

Pre-processing methodology: During the preliminary analysis of the different cases, it became obvious that one of the most important steps during the complete process mining methodology is data pre-processing (e.g. preparation of the data). Pre-processing in fact begins during the early stages of the construction of the database. During this phase, the researcher decides which information to include and exclude in the final dataset that is transformed into the event log for analysis in ProM. In addition, important steps such as renaming and aggregation of events, as well as clustering of patients based on their characteristics in the data are performed during this stage. The most important questions regarding pre-processing that needed to be answered included:

1. What steps were initiated for construction of the event log?
2. What pre-processing steps such as renaming, aggregation and clustering were performed?

3. In what order were the pre-processing steps executed?
4. Are there any other important steps that need to be considered (such as injection of timestamps)?

The case studies vary in the level of detail on their pre-processing methodology which made retracing the steps of the original researcher difficult and possibly incomplete. For this analysis, pre-processing was considered all activities that have a fundamental and permanent effect on the dataset/event log. To illustrate, mining plug-ins were not considered pre-processing as they do not change the data in any way. Certain clustering and filtering plug-ins in ProM were more difficult to classify as they do make changes to the data, but are easily reversed and changed. Therefore, depending on their place in the analysis, they can either be considered during the pre-processing or during the mining phases.

Process mining methodology: The preliminary analysis showed there are several mining algorithms that were frequently used by all researchers. However, there are also some algorithms that were only explored by a few researchers. As explained, the focus of the analysis was mainly on the process mining and clustering algorithms, and not on the process performance analysis algorithms. This was an explicit choice, as the discovery of an understandable process models is much more dependent on the right parameter settings, choices with regard to clustering and other decisions than is required for most performance analysis plug-ins. Once appropriate process models have been extracted from the event log, it is relatively easy to analyze the performance of these models without much extra effort. Visualizing a proper process model with a high fitness measure proved to be the biggest challenge in all six cases. Therefore, important issues that were investigated for the process mining methodology are:

1. What process mining algorithms are used for analysis of the data?
2. What were the parameter settings of these algorithms?
3. In what order were the algorithms executed.
4. What was the process mining result?

Identified problems/inconsistencies/errors: Part of the initial problem statement of this research is that the method of reporting the process mining methodology and results is inconsistent and unstructured. Preliminary analysis of the case studies identified many inconsistencies, errors and incomplete methodologies. Therefore, several issues and problems that were encountered during the analysis of the case studies are reported.

Concluding remarks: For each case, the findings of the analysis were briefly evaluated and reported.

Interview with researcher: During the analysis of the cases many questioned were raised with regard to the methodology and results. Therefore, we have tried to contact the original researchers to obtain additional information on the unresolved matters. Considering the fact most research has been performed more than a year prior to this project, we reckoned the possibility that the researchers could not recall all specifics. Note that in the case study reports the unresolved matters are presented in the case analysis, even though they were resolved during the interview. This should give the reader an overview of what information was provided in the report and what information was missing.

Appendix B. Case study reports

Title: Process Mining Techniques: an Application to Stroke Care [2008]

Authors: Mans, R.S., Schonenberg, M.H., Leonardi, G., Panzarasa, S., Cavallini, A., Quaglini, S., Aalst, W.M.P. van der

Introduction into the research

This (published) paper is an excellent example of the low reproducibility of process mining research in healthcare as it provides very little to no information on which algorithms are used, how they are used and when they are used to create the alleged results.

The goal of the research was to serve as a proof-of-concept for process mining (with ProM) in healthcare and to study differences between clinical pathways of different hospitals. Additionally, a second dataset is used to study patients' behaviour after they had suffered from a stroke.

The authors show that process mining can be of value for studying the differences between carepaths between different hospitals (with the HM). Furthermore, process mining can also serve as a potential tool to detect unexpected behaviour (see Yang & Hwang [2006] and Shan *et al.* [2008] for additional research on the application of process mining to detect fraudulent behaviour by medical service providers). In addition, it is demonstrated that performance analysis can be used to identify bottlenecks in the process. Finally, the authors recommend further research into the application of conformance checking as this research has only looked at process mining from a discovery point of view.

The brevity of this analysis is result of the low level of detail on the methodology in the original research.

Information on the dataset

Two datasets were used in this research. The first dataset originates from four hospitals in the northern part of Italy and contains information on 386 patients with a confirmed diagnosis of first-ever ischemic stroke (this set is also used by Gupta [2007] and the reader is referred to that case study for additional details on the dataset). Additionally, a second dataset contains information on patients from the moment of stroke onset until arrival in the hospital. This information was gathered through direct interviews with 234 patients. The origin of the latter dataset is not mentioned.

Pre-processing methodology

It is mentioned that MS Access was used for the construction of the database. However, no additional information on pre-processing activities can be extracted from the report. As a result, it is unknown how the dataset was split into different clusters for each o f the four hospitals.

Mining methodology

1. *HeuristicsMiner*: The HM was used on the different clusters for each hospital and this returned some small but understandable process models. Unfortunately, there is no information on the

parameter settings that were used, nor any information on the fitness measures of the process models.

2. *Performance analysis*: For the second dataset, a Petri net was constructed and analyzed to gain more insight in the process' performance. However, there is no information on the plug-ins that were used to create and analyze the Petri net.

Identified problems/inconsistencies/errors

- The paper does not provide any information on the pre-processing methodology.
- There is a lack of information on what algorithms have been used to create the alleged results. Moreover, for the plug-ins that have been mentioned, a description of the parameter settings is missing.
- A clear process mining methodology is lacking.
- No information is presented on the fitness measures of the models or how they are judged by healthcare professionals.

Concluding remarks

As the authors of the paper stated, due to the limited amount of space that was available for the publication, it was not possible to report all details and decisions of the process mining methodology and results. As a result, this makes replication of the results complicated and therefore the study can only serve as an introduction into the possibilities of process mining. Therefore, this paper serves as a good example of the low reproducibility of process mining research. Finally, the process model of the methodology that was used in this research can be found in figure C.1.

Interview with the researcher

Questions with regard to this research were pooled with the research of Mans *et al*. [2009].

Title: Application of Process Mining in Healthcare – A Case Study in a Dutch Hospital [2009]

Authors: Mans, R.S. , Schonenberg, M.H., Song, M., Aalst, W.M.P. van der, Bakker, P.J.M.

Introduction into the research

This research served as an exploratory study for the application of process mining using the ProM tool in a healthcare environment. The authors aimed to use process mining to understand the three different perspectives of process mining: (1) the control flow perspective, (2) the organizational perspective and (3) the performance perspective. In order to achieve their goals, the authors used data logs extracted from a hospital information system.

The authors present an inconclusive opinion on the results of process mining in healthcare. Traditional mining methods have proven to be able to present understandable process models (as was confirmed by employees of the hospital). However, these methods do have problems dealing with large unstructured processes, as in these cases the mining algorithms produce complex, "spaghetti-like", process models. Despite these complex results, the authors are convinced of the usability of process mining in healthcare. Moreover, it has been proven that it is possible to investigate the three perspectives of process mining with the ProM tool in the healthcare environment. Therefore, the authors recommend further research in the use and development of new and existing mining techniques to discover high level processes instead of spaghetti-like models. Comparable case studies need to be performed at other departments/hospitals that are similar to this case in order to verify the usefulness of process mining in healthcare.

This paper provides a solid introduction into the field of process mining in healthcare as it discusses all three of process mining and their potential in a healthcare setting. However, the paper lacks a detailed description of the algorithms and methodology that were used to create the alleged results, thereby making it difficult to reproduce these results.

Information on the dataset

The mining was performed on a dataset which originates from the Amsterdam Medisch Centrum, Amsterdam in the Netherlands. It contains information on 627 gynaecological oncology patients who were treated during 2005 and 2006. From the recorded logs from the hospital's billing system, all diagnostic and treatment activities were extracted. The data was aggregated from activities originating from several different departments, since patients often deal with several departments at the same time or during the whole process [Lenz & Reichert, 2007]. Unfortunately, no time stamps were present in the log and the authors could only recover on which date the activities were executed (as for billing purposes, it is not important at which exact time the activity took place). As a result, the ordering of events after process mining does not necessarily represent reality. In total, 376 different events were discovered.

Pre-processing methodology

It is mentioned that the pre-processing steps are important for data analysis and the researchers used two methods to reduce the size and complexity of the dataset. However, there is not much information on the pre-processing activities and the creation of the event log besides:

1. *Representatives*: Representatives (i.e. renaming) for laboratory tasks have been created to reduce complexity of the dataset. However, the paper fails to report on how the representatives were created and which software was used.
2. *Aggregation*: As with representatives, there is no additional information on how aggregation of certain events has been performed and which software was used.

Process mining methodology

3. *HeuristicsMiner:* The HM was used to mine the process models, as it can deal with noise and exceptions in the data and enables the user to focus on the main behaviour in the event log. There is a lack of specific details on the dataset was used for the first run (i.e. pre-processed or raw data). However, most likely the pre-processed dataset was used (which includes the aggregated and renamed activities). Unfortunately, there is no mention of any settings or decisions for the heuristics miner. The results after this first mining run are "spaghetti-like" and difficult to interpret. No fitness measure is reported.
4. *Trace clustering (self organising maps)*: Trace clustering was used (because of its speed) to create smaller portions of the dataset (clusters) that contain instances that are more similar to each other compared to the remainder of the dataset. Eventually, nine clusters were obtained. The only settings that are mentioned are the *self organising maps* and the *Euclidian distance*. The resulting process models (created with the HM) proved to be much more useful when compared to the process models under 3. Unfortunately, once again there is no mention of the specific parameter settings, nor is any fitness measure provided. In addition, it is not explicitly mentioned in the report but it can be inferred from the figures that artificial start and end tasks were added.
5. *Fuzzy Miner:* The FM was most likely used on the pre-processed dataset, and not the clusters derived under 4 (as can be inferred from the resulting model). The result of fuzzy mining was an understandable process model with the possibility to replay the instances in the process model, which could lead to increased understanding of the behaviour in the actual process. However, the paper lacks details on the decisions that were made and the settings that were used. Moreover, a fitness measure is lacking.

Identified problems/inconsistencies/errors

- Detailed information on the pre-processing methodology, as well as software that was used during that phase is lacking. For instance, it is not mentioned how the representatives were created and how aggregation was performed. In addition, it is not clearly stated which events have been considered for these pre-processing steps.

- It cannot be inferred exactly which datasets (e.g. pre-processed or raw) have been used for certain parts of the analysis.
- Information on the parameter settings that were used for the HM, trace clustering and FM is missing.
- Some steps, such as the addition of artificial start and end tasks, are not mentioned in the report. However, it can be inferred from the process models that these were indeed added.

Concluding remarks

This study provides a pleasant introduction into the possibilities of process mining in healthcare. Considering the nature of this paper, it is not surprisingly that some questions remain after a thorough analysis of the report as published articles only have limited space to present their results. However, does result in a low reproducibility of the research. Finally, the process model of the methodology that was used in this research can be found in figure C.2.

Answers by Mans

Dr. ir. R.S. Mans has worked as a PhD student at the Eindhoven University of Technology since 2007 on the subject "*Workflow support for the healthcare domain*" [2011]. He has published several papers on the topic of process mining [2008 & 2009] in healthcare and has been consulted by many students on his methodology[20].

As one can infer from the analysis of the two papers by Mans *et al.* [2008, 2009], many questions remain unanswered. Therefore, an open questioned interview was conducted to obtain some answers to the most important questions. Note that at the time of the interview, several years had passed since the original research. In addition, the research was performed by multiple researchers, and therefore Mans could not comment on all actions as he did not perform the complete analysis.

Answers with regard to pre-processing:

- It was unclear during the analysis which program was used for the pre-processing activities. Mans revealed that MS Access is usually used to construct a database with a lot of information on the process. ProM*import* is then used to transform the data into the .MXML file for analysis in ProM. Usually, multiple event logs are created containing specific data on issues like treatments or diagnosis (these are separated in MS Access).
- Filtering of the data for certain events is performed inside ProM with the *Event filter*. However, the second set in Mans *et al*. [2008] was filtered in MS Access.
- It is also a possibility to consider complete events only and omit the start events (i.e reduce the number of event types). This should reduce model complexity. This can be performed in ProM.

[20] Refer to https://venus.tue.nl/ep-cgi/ep_detail.opl?taal=US&rn=20000530 (retrieved 8 August 2011) for a complete list of the work by Mans.

- Aggregation and renaming is performed inside ProM. The *Remap Element Log Filter* (for renaming activities) and *Repetitions-to-Activity filter* (for aggregation) are filters that Mans has used extensively.

Answers with regard to mining:

- Self organizing maps are used by Mans *et al*. and this is frequently adopted by other process mining researchers (as we will learn from the other case studies). According to Mans, this is partially the case because these students consulted Mans on his work. Mans has used SOM because he had once achieved a good result with it and accordingly uses it ever since.
- As is clear from the case analysis, there is no optimal HM parameter setting. Mans explained that he usually only considers the following parameters: *relative-to-best threshold*, *AND threshold* and the *positive observations*. He considers the last one especially important. He cannot recall what parameter settings he used and normally "plays around" with the settings until he obtains a (according to him or the experts) good process model.

Miscellaneous:

- It was noted that for some studies no time stamp was present or it was only known on which day an activity was executed. According to Mans, this does make a difference since the exact order of events cannot be determined which results in many possible patterns in the event log. However, Mans noted that for some processes it is not important to include a complete time stamp. As an example, for the diagnosis phase, many activities are ordered and executed at the same time and therefore it is not necessary (or possible) to establish an exact order of events (because they are indeed parallel in reality).
- It was indicated by Mans that it was unclear to him what exactly he had done for process mining. Moreover, most of the time, research is performed in collaboration with other researchers and as a result he does not always posses information on the choices and settings that are used during the complete research. He suggested himself that it would be convenient for a standardized methodology for reporting process mining and its results. However, he noted that it might be a problem when not all researchers would consequently report their choices and results according to such a template.

Title: Process improvement in Healthcare: A data-based method using a combination of process mining and visual analytics [2009]

Authors: Riemers, P. (Supervised by Jansen – Vullers, M.H., Dongen, B.F. van)

Introduction into the research

The goal of this study was to find a data-based step-wise method, based on the CRISP-DM framework (Cross Industry Standard Process for Data Mining, the reader is referred to Chapman *et al.* [2000] and Two Crows Corporation [2005] for additional information), to implement the control and the diagnosis phases of the BPM lifecycle in a healthcare environment. This was achieved by both investigating the applicability of the ProM and MagnaView[21] (MagnaView is a data visualization software tool) tools as a way to obtain detailed insights into hospital processes. The use of the ProM software is limited (while MagnaView enjoys considerable more attention), albeit that the report does show that process mining with ProM has potential in a healthcare setting. Furthermore, it shows interesting details on the construction of the database and the final analysis itself. The resulting approach for the control and diagnosis steps of the BPM lifecycle is presented and consists of seven steps: (1) Build database, (2) Introduction session, (3) Preliminary analysis, (4) Preliminary meeting, (5) 2nd analysis, (6) Final meeting and (7) Documentation. This approach is validated and extended in a different hospital by Ramos [2009] (it is briefly mentioned by Riemers that the research would be continued by another student).

Information on the dataset

Two datasets were used for this study, one homogeneous set called *mammacare* (DBC-code diagnoses 317 and 318) and one heterogeneous set called *diabetic foot* (DBC code diagnosis 432). Both datasets originate from the Atrium Medisch Centrum Parkstad, Heerlen in the Netherlands. The former dataset is split into two sub-sets called *conservatief poliklinisch* (DBC-code treatment 101) and *enkelvoudig poliklinisch conserva*tief (DBC treatment 104). These sub-sets have 95 and 39 event classes and 2000 and 1500 PIs respectively. In total, during the period 2006-2008, 4700 unique PIs are found for the complete *mammacare* treatment process, indicating that most patients (75%) only followed one sub-process. For the *diabetic foot* dataset, only information on the most complex sub-process is provided, the so called *Operatief met klinische episode(n)* (DBC treatment 203) process. It contains 104 patients and 74 event classes. It is mentioned that a large number of events (up to 75%) is only performed for a small number of patients.

Pre-processing methodology

The database and event log were constructed according to the following steps:

1. *Extract*: The data for this research was extracted from the hospital database and a new database was constructed by an expert at MagnaView B.V.
2. *Transform*: During pre-analysis of the data, several problems with respect to the data were identified. First, no timestamps were available for the events. Second, appointment data did not

[21] Refer to http://www.magnaview.nl/ (retrieved 8 August 2011) for additional information on MagnaView.

contain any DBC-code and as a result no explicit match between treatment events and appointments could be made. Third, for most events only the reporting specialist was mentioned (who is not necessarily the specialist that has performed the event). Fourth, information regarding the department was absent for many events, making automated aggregation difficult.

Aggregation was performed on two levels: events and DBC-codes (although it is not mentioned which software was used). In the former case, all laboratory tests (>80) were renamed to *Klinische chemie* and the new event class was aggregated subsequently. These tasks were performed in order to reduce the number of event classes and total number of events. The laboratory tests are not of any interest for the process model that represents the global behaviour in the event log. In the latter case, all process instances with similar DBCs (at the treatment level) were clustered to form subsets of patients with more homogenous processes. In MagnaView, the data was visualized and after the order of events had been determined (by observation and interviews), artificial time was added to the events ("*this was only important for analysis in MagnaView, not ProM*", Riemers [2009]). However, it is not mentioned how the artificial time was added.

3. *Load*: ProM*import* was used to create an .MXML file format for analysis in ProM.

Process mining methodology

The methodology that was reported is highly unstructured as much information is presented across different sections in the report. In addition, detailed information on many steps is missing. However, globally the following research steps have been executed:

4. *Filtering*: In order to deal with the large number of event classes in the event log, automated filtering was applied to reduce the complexity of the resulting process models. In ProM this is performed easily with the *Enhanced event log filter* (initially this is not mentioned but it is referred to later on in the report). Event classes with a frequency lower than 1% were discarded. However, some event classes were kept in the database after consultation with the experts, as low frequency does not necessarily equal uninteresting. For this step, no distinction is made between the different datasets.
5. *Clustering*: As well as manual clustering (see pre-processing), trace clustering (SOM) was used to cluster patients inside the DBC-clusters that were created earlier. Next to a focus on activities, no additional information on the parameter settings is provided.
6. *HeuristicsMiner*: The HM was the only algorithm that was used to mine the process models. Few details have been provided on the application of the HM, its parameter settings and decisions. Results varied across the different event logs, ranging from complex models with low fitness measures, to more comprehensible process models with higher fitness measures. Below we present the results of the different process mining steps.

Mining on the *Conservatief poliklinisch* event log:

6.1 When the HM (with default settings) is tested on the original unfiltered and un-clustered event log, a complex process model is extracted for which several problems were identified: the fitness measure of the model is very low (less than 0,05, no information on which measure was used) and moreover, many ordering relationships are not included in the model.

6.2 When the event log was filtered (not specified which filter) for event classes with a low frequency (<1%), only 17 event classes remained in the event log. The HM (unknown which settings) was used to produce a process model. Due to a lower amount of event classes, the resulting model was much simpler than the model under 6.1. However, the fitness measure only increased slightly (just above 0,05). As a result, it was concluded that removing low frequent event classes creates a more homogenous event log, but it does not necessarily decrease the variation in the process (as stated by Riemers).

6.3 Clustering with SOM (unclear which event log is used, i.e. the filtered or unfiltered) resulted in two large clusters (883 and 669 PIs). The resulting process models (mined with the HM with default settings) were comparable but had different fitness measures of >0,1 and <0,01 respectively. Consequently, it can be concluded that clustering of the data does not necessarily lead to better process models for all PIs.

Mining on the *Enkelvoudig poliklinisch conservatief* event log:

6.4 The second sub-process was analyzed according to the same methodology that was used for the first sub-process and this yielded conflicting results. Mining on the unfiltered event log resulted in a process model with higher fitness (>0,1) than the process model extracted from the filtered event log (<0,01). After clustering (SOM), a process model with a fitness of 80% was extracted. However, it should be noted that this cluster only contained three main and six very low frequent event classes (only executed once or twice in the total log). Therefore, the latter process model was very simple and not useable as a method to identify global patient behaviour.

Mining on the *diabetic foot* event log:

6.5 Due to the heterogonous nature of the *diabetic foot* dataset, the resulting process models were much more complicated. To illustrate the differences between the two diagnoses, only the most complex *diabetic foot* sub-process was used, namely *Operatief met klinische episode(n).* Unfortunately, not much information is provided on the process mining methodology for this part of the research. Mining on the initial raw event log resulted in a complex process model with low fitness (<0,01) and much variability in process patterns. Filtering the data retained 59 event classes but the resulting process model was still difficult to understand and had a low fitness measure (<0,01). Clustering with SOM resulted in clusters that were too small to be useful for mining (the largest cluster contained only 15 PIs and resulted in a process model with a fitness measure of 0,01).

Identified problems/inconsistencies/errors

- Detailed information on both the *mammacare* and *diabetic foot* datasets is missing.
- The process mining methodology is far for complete as many elements are missing. Moreover, the different steps that are mentioned are scattered throughout the report. Combined, this drastically lowers reproducibility and understandability of the research.
- For various steps it is not mentioned what software tools were used.
- No parameter settings have been specified for most plug-ins.
- It cannot be inferred which datasets (e.g. filtered or unfiltered) have been used exactly for most steps in the analysis.

Concluding remarks

This research served as a proof of concept for the applicability of a data-based step-wise method to manage the diagnosis and control part of the BPM lifecycle in healthcare. Moreover, it aimed to construct a guideline on how to perform these steps. The report shows the applicability of process mining in a healthcare setting and it provides several important and useful insights. The resulting process models could be generated within limited time but were complex and had a low fitness measure. As a result, the medical experts did not judge the models as very useful. Moreover, the process models were difficult to interpret and therefore it did not result in any actions from the client's side. Furthermore, this research showed that there is a lot of variability in the patients' processes. Considering the report by Riemers was created with the intention of serving as a guideline, too many questions remain after thorough analysis and therefore the reproducibility is low. Finally, the process model of the methodology that was used in this research can be found in figure C.3.

Answers by Riemers

The interview with Riemers mostly served to gain additional knowledge on the dataset (which is not presented here). This was required since the Atrium data would be used during the validation part of this research.

- According to Riemers, he performed the renaming and aggregation of events (chemical and microbiology activities) in MS Excel. This was an explicit choice as he found it easier to use Excel than using the options in ProM.
- Riemers cannot recall the specific parameter settings he used during the analysis. However, he mentioned that he did not spend considerable effort in trying various settings and used the default settings mostly.

Title: Healthcare Process Analysis: validation and improvements of a data-based method using process mining and visual analytics [2009]

Authors: Ramos, L.T. (Supervised by Jansen – Vullers, M.H., Weijters, A.J.M.M.)

Introduction into the research

This research is an extension of the work that was performed by Riemers [2009] and it provides the reader with additional information on process mining in healthcare, adding slightly more detail and additional algorithms to the research by Riemers. Although the focus of the research is divided between MagnaView and ProM, the results show that understandable process models can be mined with the HM, albeit after some pre-processing and clustering has been applied. The report is divided into two parts. First, the work of Riemers is validated in a different healthcare setting. Second, the goal is to recreate the results of MagnaView in the ProM software and to make a comparison between the approaches with the different software.

The result of this thesis is the validation and adaptation of the method that was proposed by Riemers. In future research, this framework can (and according to the researcher should) be used to analyze hospital process data. Finally, it is concluded that process mining with healthcare data is possible. However, one needs to spend much effort on pre-processing of the data by reducing the number of event classes which should decrease the size and complexity of the dataset. As a result, the process models have an increased chance for usability.

Information on the dataset

The dataset that is used for this research is an extension (i.e. a few additional PIs) of the dataset that was used by Mans *et al.* [2009] and again it is only known for events on which day they were executed. The dataset contains information on 682 patients that were treated at the gynaecology oncology department in the AMC and consists of 43.615 ATEs. The diagnosis and treatment processes from the 3rd of January 2005 until the 20th of March 2008 were extracted from the hospital's billing system. The corresponding dataset is divided into six sub-processes based on the DBC-code information: *M11 maligniteit vulva*, *M12 maligniteit vagina*, *M13 maligniteit cervix*, *M14 maligniteit endometrium*, *M15 maligniteit myometrium* and *M16 maligniteit ovarium / tuba*.

Pre-processing methodology

The dataset was constructed according to the following process:

1. *Extract*: The data was extracted from the hospital's billing system "*for the reason that all activities performed for each patient must be trustworthy acquired in order to charge the correct amount of money to each patient*" [Ramos, 2009].
2. *Transformation*: The administrative tasks were removed from the dataset as these were not of interest for the main patient treatment process. However, it is not mentioned how this was performed and which software was used.

3. *Load*: The .MXML file was already available (obtained from Mans) and did not need to be constructed. Further on in the project, a .XLS file was used for pre-processing and adding data-attributes in Excel. However, it is not mentioned how this step was executed.

Process mining methodology

The first part of the research focussed on the validation of the method that was developed by Riemers [2009]. The first of the validation was a preliminary analysis:

4. *Pre-processing*: It was decided that several pre-processing steps were not necessary until right before the actual mining. In conclusion with medical experts at AMC it was decided to rename all events into the English name of the department that had performed the event. Furthermore, it was decided to aggregate events at the level of a visit to a certain department per day. These steps are justified as the initial goal is to gain process knowledge at the level of departments and possibly from thereon investigate certain parts of the process in more detail. In addition, it appears (from the process models) that artificial start and end tasks were added to the event log (which is not explicitly mentioned in the report).
5. *Clustering*: Similar to the research by Mans *et al.* [2009], SOM with Euclidian distance and default settings was used to cluster patients. This resulted in two clusters with 69 and 613 patients.
6. *Performance sequence diagram analysis*: The result of the PSDA showed that there were 306 unique process patterns for the group of 613 patients and 44,2% of the patients had a unique pattern. For the group of 69 patients, it is reported that there were 49 unique patterns and 63,8% of the patients had unique patterns. This result indicates that there is much variability in the process (and thereby making the actual mining more complex).
7. *Mining*: The HM with default settings was used to produce process models for both clusters that were obtained with SOM under 5. No fitness measures are provided but it is mentioned that the models are large, complex and not usable.

During the preliminary meeting with the process experts some feedback on the results was provided. Subsequently, a second process mining iteration was started. For this part of the analysis, the pre-processed dataset from before the preliminary analysis stage was used (step 3).

8. *Pre-processing*: It was decided to aggregate all laboratory tasks that were performed on the same day. Moreover, the administrative tasks were omitted. Furthermore, only patients that started their treatment process in the timeframe of the dataset were considered for this part, which resulted in a group of 362 patients. Again it is not mentioned that any artificial start and end tasks were added but this can be inferred from the reported process models. Unfortunately, it is not mentioned how the pre-processing steps were executed and what software was used.
9. *Clustering*: SOM was used with Euclidian distance and default settings to cluster the event log. This resulted in one large cluster with 315 patients and one smaller cluster with 47 patients.
10. *Performance sequence diagram analysis*: The PSDA was used to investigate the different patterns in the dataset. It was revealed that the group of 315 patients had 206 different patterns and 47,5% of the patients had an unique process pattern. The group of 47 patients

contained 44 unique patterns, resulting in 87% unique patterns. These results indicate that there is much variability in the process.

11. *HeuristicsMiner*: The HM (no settings are mentioned) was used on both clusters that were obtained under 9. The resulting models (no fitness measures are provided) were still complex but of better quality compared to the process models obtained under 7. In addition, it is reported that the process models did not provide any additional information for the medical stakeholders.

An additional analysis was executed to investigate the use of ProM by trying to replicate the MagnaView results in ProM. The part of the analysis started with the dataset that was obtained under 5.

12. *Pre-processing*: Artificial time (i.e. adding more detail in the form of hours and minutes) was added to the data in MS Excel but it is not explicitly mentioned how this was performed. Adding time allowed the researcher to use the data for the dotted chart analysis. Furthermore, artificial start and end tasks were added to the data.
13. *Clustering*: For this part of the analysis it was decided to cluster the patients according to their DBC-code information (there are six different DBC-codes in the dataset) by using the *LTL Checker* plug-in in ProM. Subsequently, SOM with Euclidian distance and default settings was used to cluster patients inside a DBC-code cluster. This resulted in the following:
    - *M11*: One cluster with 50 patients.
    - *M12*: Three clusters with 2, 2 and 1 patient.
    - *M13*: One cluster with 138 patients.
    - *M14*: Four clusters with 53, 10, 3, and 2 patients.
    - *M15*: Three groups with 3, 2 and 1 patient.
    - *M16*: Three groups with 57, 19 and 19 patients. The largest group has 35 different patterns and 74,2% unique patterns. The most common pattern is repeated 8 times.
14. *Mining*: Three algorithms were tested to obtain a process model: the HM, GM and FM. The ExtraBehaviourPunishment fitness measure was chosen to compare the models as it measures both the completeness and preciseness of the process model [Alves de Medeiros, 2006]. All analyses were performed on the group of 57 patients within the *M16* cluster.
    - *HeuristicsMiner*: During the analysis with the HM some settings had been changed. For instance, to deal with the laboratory tests and repeated visits, the *AND threshold* was changed to 10 and the *Length-one-loops threshold* was changed to 0,999. This resulted in an understandable process model with a fitness measure of 0,8389.
    - *Genetic Miner*: Mining with the GM (no settings are specified) resulted in a good process model with a fitness measure of 0,8767.
    - *Fuzzy Miner*: As with the other two mining algorithms, the FM produced a sensible process model. However, it is not possible to judge the quality of the FM model with the EBP measure and the FM uses a measure called *log conformance* (which measures only completeness). The resulting process model had a fitness measure of 0,8244. Unfortunately, there is no information provided on the settings that were used for the FM.

Identified problems/inconsistencies/errors

- The process mining methodology of the research is far from complete and scattered throughout the report, thereby lowering the reproducibility.
- Information on some software tools that were used is missing.
- There is contradicting information in the report as the presented process models do not fit their description (e.g. aggregation is visible in the model but not mentioned in the report).

Concluding remarks

For most steps in the analysis of hospital data it was preferred to use the MagnaView tool over ProM. However, when one is interested in the process patterns in the dataset it is recommended to use the latter tool. Moreover, as MagnaView cannot produce any process models, the researcher is automatically referred to ProM for that part of the analysis.

The results of process mining differ. When mining is performed on the complete dataset, no useable models are produced. However, for the smaller DBC clusters it was possible to extract more sensible models with high fitness measures. In addition, this research has proven that there is a lot of variability in the healthcare processes. To illustrate, even for the clustered patient group with DBC-code M16, almost 75% of the patients had a unique process pattern.

As with the report of Riemers, this report was intended to serve as a guideline for analyzing healthcare process data. However, many inconsistencies and errors are identified thereby lowering the reproducibility of the methodology. Finally, the process model of the methodology that was used in this research can be found in figure C.6.

Answers by Ramos:

- Ramos has used MS Access and MS Excel to create the database for his research and perform some pre-processing tasks.
- MagnaView was used to filter for patients that started their treatment during the timeframe of the event log.
- Ramos has used a specially developed LTL Checker formula to cluster for the different DBC-codes in the event log.
- No algorithms besides the few that are mentioned in the report were used during the research.
- According to Ramos, the AMC perceived the results as "*usable*".

Title: Process improvement in mental healthcare: A data-based method for care delivery process analysis in GGzE Centre child and adolescent psychiatry [2010]

Authors: Zanden, L. van der (Supervised by Jansen-Vullers, M.H., Kleingeld, P.A.M., Joosten, T.C.M.)

Introduction into the research

This paper is based on the research by Riemers [2009] and Ramos [2009], but in contrast with the previous research, the focus is on mental healthcare trajectories. Unfortunately, the use of process mining is limited and not discussed in much detail as for instance no information is provided on the parameter settings. Moreover, no additional algorithms (such as clustering) have been used as an effort to create better process models.

Information on the dataset

The dataset that was used for this project was extracted from the DBC-information system at GGzE Centre child and adolescent psychiatry. The data was recorded from November 2007 until November 2009 and consisted of 21 text files that contained information on care trajectories, DBC trajectories, diagnoses, activities, day spending activities, and days of stay. In total the dataset counted 3.511 clients and 3.607 care trajectories were opened for these clients. Of these trajectories, 2.127 are closed care trajectories (i.e. both a starting date and a finishing date are defined in the dataset).

Pre-processing methodology

The construction of the initial database with MS Access was complex. Generally, the method created by Mans [20XX] was followed to transform the database into a .MXML file and the report mentioned the following steps:

1. *Design database architecture*: During this step the different data tables and connections between the tables are constructed.
2. *Import data*: The data was imported into the tables by means of text files.
3. *Remove flags*: All flags were removed from the database.
4. *Filtering*: The data was filtered for GGzE Centre child and adolescent psychiatry. To limit the focus of analysis, only some DBC trajectories were considered, namely 4 and 7, named *Kinder & Jeugd* and *Forensisch jeugd* respectively.
5. *Remove unmatched records*: All unmatched records that occurred due to the previous step were omitted.
6. *Verification*: The database was verified by the parallel construction of a twin database by a GGzE project manager.
7. *Deletion of unfinished care trajectories*: This step was performed in MS Access.
8. *Removal of multiple tasks (performed by team)*: Some events were executed by more than one resource and accordingly were recorded in the dataset an *n* number of times for *n* resources. Therefore, *n-1* activities had to be removed per patient if they were executed by *n* resources as

a team at the *same* time during the process. It can be inferred that this task was performed in MS Access.

9. *Transformation*: The data was prepared in MS Access and subsequently it was transformed with ProMimport. The report includes an extensive, but still far from complete, description of this process (refer to Mans [20XX] for the complete methodology). Several event logs were created for analysis in ProM (based on the level of abstraction of the events).

Process mining methodology

10. *HeuristicsMiner*: The first mining step was performed with the HM. It is not clearly mentioned which dataset was actually used, nor are any parameter settings specified. The result is a complex "spaghetti-like" process model, from which we can infer that the level 3 activities (lowest level of abstraction) was used. Another process model is mined for the event log that contained the closed care trajectories, was clustered based on level 1 naming (highest level of abstraction), activities executed by more than one resource were grouped and resources were renamed as *multiple_name level1 activity*. It is not mentioned which parameter settings are used. The result is a much simpler and smaller process model (since only 8 event classes remained). This process model had a fitness measure of: ProperCompletion and StopSemantics have a fitness of 0,0, ContinuousSemantics = 0,359, ImprovedContinuousSemantics = 0,583, and ExtraBehaviorPunishment = 0,559.
11. *Fuzzy Miner*: The FM was applied to the data because of its ability to deal with unstructured processes. For this part of the analysis, the following pre-processing criteria were applied: closed care trajectories, activities level 2, professions level 2, activities executed by more than one resource were grouped and resources were renamed as "*multiple_name level1 activity*", pervasive diagnosis only, care trajectories containing more than 5 events, no waiting times longer than 60 days. Default parameter settings were used and there is no mention of any parameter settings.

Identified problems/inconsistencies/errors

- The process mining methodology is not reported completely and several steps are scattered throughout the report. As a result, considering the intentions to serve as a guideline, the reproducibility is low.
- Detailed information on the dataset is missing.
- No parameter settings are specified.

Concluding remarks:

Only a limited number of process models were mined during this research. Moreover, only at the highest level of abstraction could sensible process models be created, resulting in only a very global overview of the process and no detailed insights. No additional pre-processing such as clustering was performed in order to create more sensible process models at lower levels of abstraction. Furthermore,

the report is somewhat unstructured, making analysis more difficult. Finally, the process model of the methodology that was used in this research can be found in figure C.5.

Feedback by Tom Joosten (supervisor GGzE)

Tom Joosten reckoned the potential of process mining in the GGzE setting. However, the research by Zanden on the use of process mining at GGzE was limited. The few process models that have been produced and that were understandable only considered a high level of abstraction and therefore did not provide much detailed insights in the process. As a result, Joosten would very much like to see the additional value of process mining at GGzE by for instance using process mining at lower levels of abstraction. Moreover, Joosten indicated that the report was unclear on certain parts of the analysis. To illustrate, it was not always clear to him which decisions and/or actions were taken and the reasons for these actions was lacking in some cases (this remark is a good indication of the low accessibility and reproducibility).

Answers by Zanden:

- Zanden indicated that she had difficulties while exploring the different possibilities of the ProM tool. For instance, in many cases the literature on the ProM plug-ins did not specify enough details to understand how the plug-ins actually worked and should be applied to the data (this is an excellent example of the low accessibility of process mining).
- Pre-processing was performed in MS Access as she found it easier to use than the options in ProM.
- Zanden has not investigated the use of the different *HeuristicsMiner* parameter settings in much detail.
- According to Zanden, in large event logs it is inevitable to obtain “spaghetti-like” process models, even when dealing with a linear process in the first place.

Title: Workflow and Process Mining in Healthcare [2007]

Authors: Gupta, S. (Supervised by Aalst, W. M. P. van der, Weijters, A. J. M. M., Alves de Medeiros, A. K.)

Introduction into the research

The goal of this research was to identify possibilities and problems with the existing process mining algorithms in ProM and subsequently develop a new algorithm to deal with these problems. Accordingly, this research led to the proposition of the association rule miner (ARM) which subsequently has been developed and implemented in the ProM tool. This paper serves as a good example of how process mining could be described, as there is a good focus on both the theory behind mining algorithms, as well as the practical implications of the parameter settings.

Information on the dataset

Two datasets were used during this research. The first dataset was used to test the current ProM plug-ins and originates from the intensive care unit from the Catharina Ziekenhuis Eindhoven, the Netherlands. It contained detailed information on 23.779 patients and information on the time of events (timestamp), complications, diagnosis, investigations, measurements and characteristics of a patient's clinical admission. Several subsets of this dataset have also been used throughout the research. The second dataset was used for experiments with the newly developed ARM. The data originates from four hospitals in the districts in the northern Italian region of Lombardia and pertains to 386 patients of acute strokes and transient ischemic attack on first-ever stroke patients on which the effect of guidelines by the American Heart Association was studied (the same dataset used by Mans *et al.* [2008]).

Pre-processing methodology

1. For the first part of the research, the dataset of the Catharina Ziekenhuis was used to test the current process mining algorithms. A database was constructed in MS Access and ProM*import* (version 3.0) was used to create a .MXML file that could be loaded into ProM (version 4.0). No additional information is provided on the pre-processing steps.

Process mining methodology

Testing the HeuristicsMiner

2. The HM was tested on a sub-set (called *complications*) with 576 PIs and 185 event classes (including artificial start and end tasks; although they are not mentioned it can be inferred from the presented process model that these have indeed been added). With default parameter settings this resulted in a "spaghetti-like" process model with many dangling tasks, missing connections and low fitness (ImprovedContinuousSymantic = -0,44). A second test was performed on a set called *treatments* with 2.711 PIs and 253 event classes. Mining with default settings resulted yet again in a complex process model with low fitness, dangling and missing tasks.

3. During additional tests of the HM, higher values for parameters like the *Positive Observations*, *Dependency threshold*, *Length-one-loops threshold* and *Length-two-loops threshold* were used. The resulting models are not reported but are allegedly of better quality than the models under 2. However, many dangling tasks and missing connections are still observed. Additionally, the all-activities-connected heuristic was tested and it was discovered that switching this parameter off results in a process model with an increased number of dangling tasks. Unfortunately, it is not mentioned which datasets were used for this part of the analysis.
4. The HM was tested on a small log (*uro-genitaal*) that contained 6 PIs and 18 different event classes (including the not explicitly mentioned artificial start and end tasks). Mining with default settings resulted in a process model with a good fitness (i.e. 0,88, although it is not mentioned which measure was used). However, as a result of using a small event log, it could easily be observed that some connections were missing. By changing the following parameters: *Dependency threshold* = 0,5, *Positive observations* = 1, and *L1L threshold* = 0,5, a model with a fitness of 1,0 was achieved. However, when the exact same procedure was tested on another small log called *CNS* (15 PIs and 22 event classes), no "perfect" model could be achieved. In fact, the process model had a poor fitness measure of -0,17. This result shows that despite the possibility to change the parameter settings, this does not always lead to a good process model (or that there is something such as an optimal parameter setting).
5. A large event log with 2.711 PIs and 9 event classes was used for additional testing. This log only contained events that occurred more than 6,03% in the complete event log (this was achieved by using the *Enhanced event log filter* in ProM). Mining with the HM and default parameter settings resulted in a process model with many missing connections. Changing the *Relative-to-best threshold* = 0,99, *Positive observations* = 1, *Dependency threshold* = 0,1, *L1L threshold* = 0,1 and *L2L threshold* = 0,1; resulted in a better process model (the fitness measure is 0,78 but it is not specified which measure). However, it is concluded that the HM has troubles capturing all connections in the event log, as well as identifying all the mixed AND/XOR join/split relationships between events that are characteristic for healthcare data. Therefore, despite a high fitness measure, the quality of the model is questionable.

Testing the Disjunctive Workflow Schema (DWS) Miner

6. The DWS miner was tested on the *complications* log. With default settings this resulted in two discriminant rules, a global process model and two process models based on the discriminant rules. These resulting models are easily understood but since the DWS miner is based on the HM it also inherits all its problems. Therefore, in the end, the DWS miner is not ruled the ideal process mining algorithm when using healthcare data.

Testing the Association Rule Miner

For the sake of brevity and the little variation between the different tests with the ARM, not all tests have been included in this case report. Representative tests have been chosen which include all tested options, thereby providing a complete overview of the methodology and possibilities of the ARM.

7. The ARM was tested on a log containing treatment and complications data (2.269 PIs and 174 event classes). The *Apriori* algorithm was tested with default settings, except the population size was set to 100 and this resulted in five rules. This result can be blamed to the low frequent behaviour in the event log and a high value of the support threshold. However, in this case lowering the minimum support threshold to 0,1 did not result in any additional rules. Furthermore, the *PredictiveApriori* algorithm was tested which resulted in eight rules.
8. A complications log with 38 PIs and 65 event classes was used to test the clustering of PIs based on association rules. From a figure it can be inferred that the *Apriori* algorithm resulted in four rules. When the HM is applied to a cluster based on an association rule, a much simpler process model is obtained compared to mining on the complete (i.e. non-clustered) event log. However, common problems such as dangling tasks are still present.

Evaluation of the ARM

The event log originating from the Italian hospitals was used to verify the ARM as an additional algorithm to gain useful insights into processes.

9. The event log that is used for the first illustration contained 373 PIs and 7 event classes (i.e. measurements). Mining with the HM (no settings are specified) resulted in a small and understandable process model. Both the *Apriori* and *PredictiveApriori* algorithm resulted in several association rules. However, it is noted that low frequent events are difficult to capture, even when a high number of required rules is specified. Therefore, it is difficult to decide how the ARM parameters should be specified to capture all potential rules.
10. For the last test, an event log with 380 PIs and 35 event classes with regard to therapies was used. Mining with the HM (no settings mentioned) resulted in a complex process model. Both the *Apriori* and *PredictiveApriori* algorithms are used to mine association rules. Both algorithms produced many rules with ranging confidence and accuracy. Clustering was performed based on a rule that was obtained with the default *Apriori* mining settings. The resulting process model of this cluster was much simpler than the initial process model of the complete log.

Identified problems/inconsistencies/errors

- The methodology is incomplete and unordered for some parts of the research.
- It is cannot always be inferred from the report which datasets have been used for analysis.

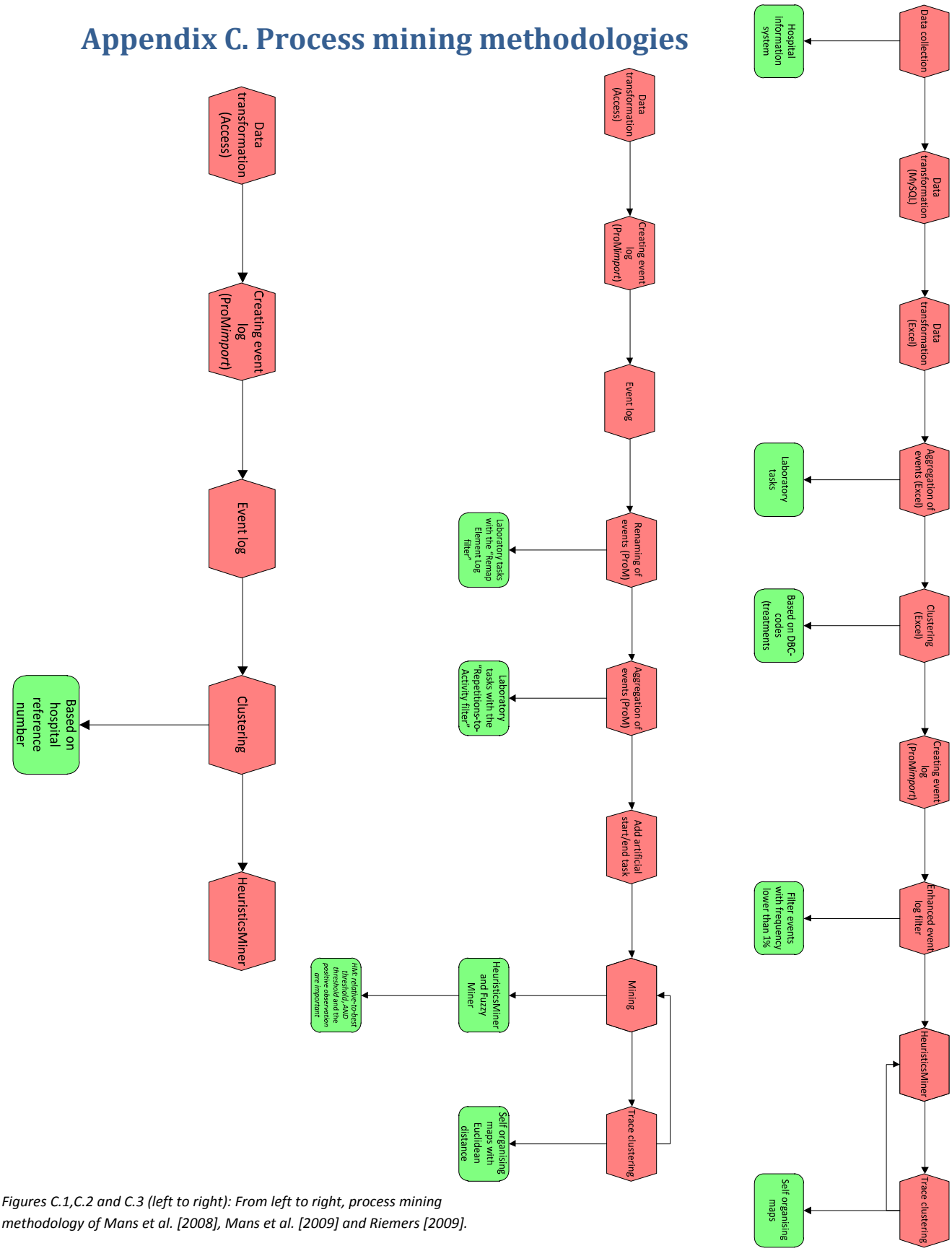
Concluding remarks

This study is very well documented and most of the steps and decisions are well reported. Moreover, an extensive analysis of the different HM, DWS miner and ARM parameters is presented and their implications are considered. Therefore, despite its shortcomings, this report could serve as an example of how process mining should be reported. Figure C.4 depicts the process model of the methodology.

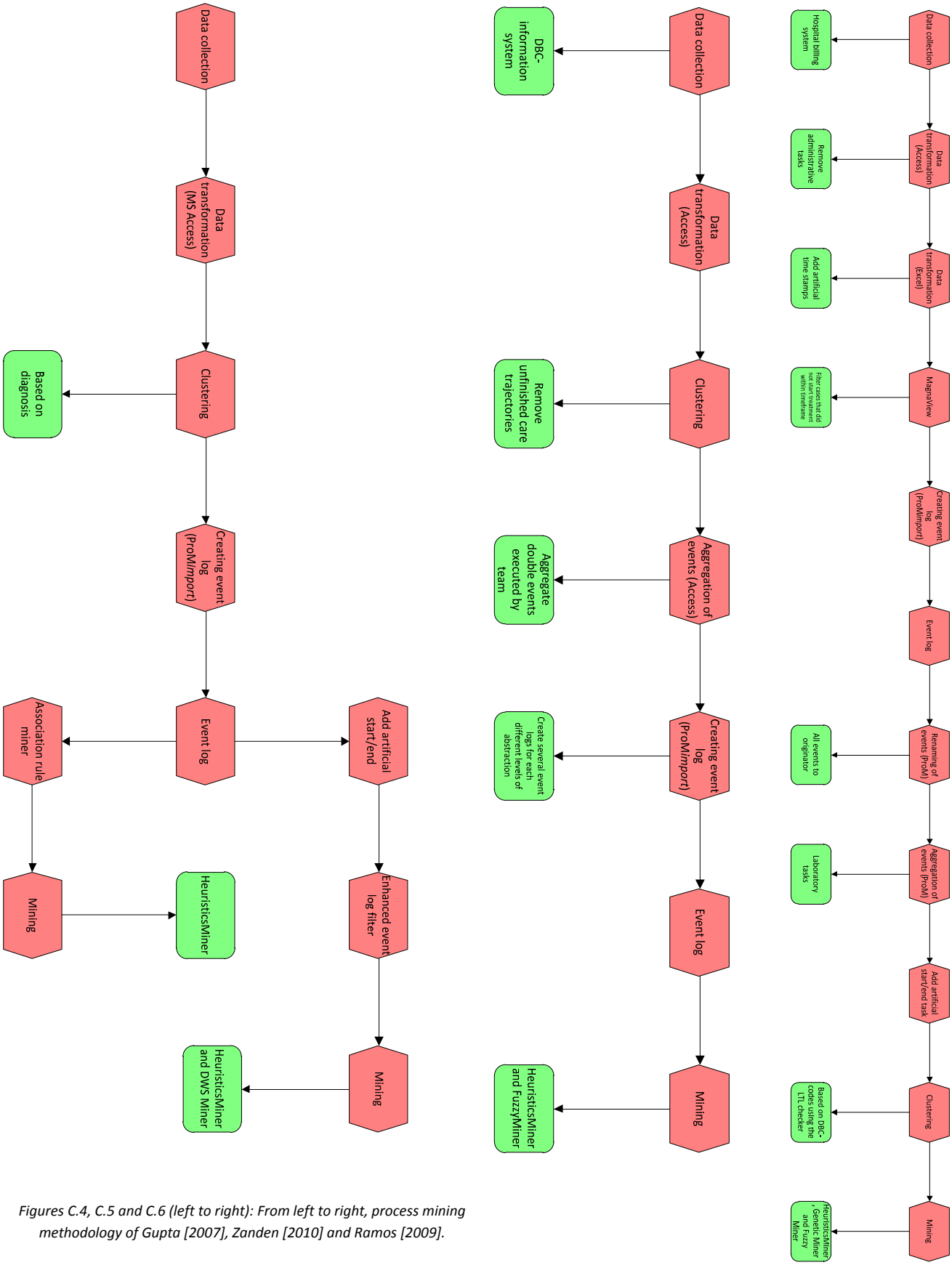
Interview with the researcher

No interview was performed because no reaction from the researcher was obtained.

Appendix C. Process mining methodologies



Figures C.1,C.2 and C.3 (left to right): From left to right, process mining methodology of Mans et al. [2008], Mans et al. [2009] and Riemers [2009].



Figures C.4, C.5 and C.6 (left to right): From left to right, process mining methodology of Gupta [2007], Zanden [2010] and Ramos [2009].

Appendix D. Interview format

The following is a general overview of the interview format that has been used for the interviews during this research (sent to the interviewees as an e-mail). Note that deviation from this "guideline" was possible and this format is reported solely to provide the reader with an idea of the focus of the interview and the kind of questions that were asked.

E-mail of the rough interview outline

As I have explained in the previous e-mail, I am interested in the opinion and approach of experts for certain process mining problems. In this document I will explain the setting and the questions I have for you.

I have analyzed several theses by TU/e students and some papers by Ronny Mans on the application of process mining in healthcare. Their methodologies have been compared and as it appears now, the written reports do not show sufficient specifics for high reproducibility. Therefore, I am interested in the specifics as I aim to increase reproducibility of process mining research. In addition to the written reports, I aim to include process mining knowledge that is provided by several process mining experts. This means that I hope you can be as specific as possible in your answering, detailing why and how you would perform certain actions (or why not). This for instance includes certain parameter settings that you use or think are important, plug-ins and other software.

Please consider the hypothetical case in which you are in the position of one of the previously mentioned researchers, i.e. your goal is to extract a process model from hospital data. Note that for my research I am only interested in how eventually you obtain a process model and not in additional possibilities such as performance analysis, social networks etc (it might be possible however you use these in order to obtain a process model, in that case I am interested in them).

Roughly, I have divided the analysis in three parts: database construction, pre-processing and process mining. Not necessarily do research steps belong to one stage only and of course the whole analysis can be considered iterative, please state whenever this is the case. Furthermore, it might very well be possible you have not performed some tasks I am asking for. In that case, please state this and explain what you think you would do should you have to do it.

Consider that you start at the beginning, that is: extract the data from its source and transform it into a usable format.

Database construction

1. *What part of a hospital information system (e.g. billing system) do you consider most valuable as a data source and why?*
2. *What software do you use for the construction of the database from which you will create an event log and how?*
3. *What kind of data characteristics and attributes would you be after besides the activities (e.g. timestamps, DBC-codes, patient characteristics.) and why?*

4. *What steps do you perform to get to an event log and how?*
5. *Is there any other information that is relevant?*

Let's consider the case that we have a (raw) event log with at least the following (this is to show roughly what other researchers have been dealing with and what I am using in during thesis):

- *The data originates from one hospital department but could contain multiple treatment paths (i.e. variability in process paths is probably high, as is the typical case and problem for many hospital datasets anyway).*
- *1.000 PIs (different patients)*
- *300 event classes (this includes treatment, diagnosis, administrative and laboratory activities)*
- *50.000 events*
- *Timestamp with information on the day an activity is performed.*
- *Several other data attributes such as DBC-code and patient's characteristics.*

Applying process mining to the previously described event log most likely results in a spaghetti-model, and therefore probably needs pre-processing.

Pre-processing

6. *What pre-processing steps would you perform (e.g. removing certain tasks to make the model less complex), why would you perform them and what plug-ins/software would you use?*
7. *Is there any other information that is relevant?*

Now let's consider that the pre-processing stage is finished and an event log is created for analysis in ProM. As a result, we can now begin the actual mining stage.

Process mining

8. *Which steps would you take to obtain a good process model? And what is the order of these steps?*
9. *What algorithms would you use and why?*
10. *What parameter settings would you use or consider candidates for changing and why?*
11. *How would you judge a model for its value/quality and why?*
12. *Would you consider further processing of the data, such as clustering, to obtain even better process models for specific groups of patients?*
13. *Is there any other information that is relevant?*

Appendix E. Interview transcripts

Interview with dr. A.J.M.M. Weijters

Dr. A.J.M.M. Weijters is an associate professor at the Eindhoven University of Technology and is the (co-)author of many process mining papers. He is the leading developer and continually works on the HeuristicsMiner, one of the most frequently used mining algorithms in the ProM framework. Furthermore, he has supervised numerous (PhD-)students on their process mining research[22].

Database construction

Dr. Weijters acknowledged he has hardly ever needed to construct a database with hospital data from scratch (when required a thesis student performs this work) and usually generates his own data with CPN tools. Therefore, during the remainder of the interview a raw event log such as *AMC global* was discussed, containing low and high frequent events, as well as treatment and laboratory events.

Pre-processing methodology

When a .MXML file is created or obtained, the first step is to see what kind of data and process is being is being considered for analysis. This implies globally exploring the lay-out of the process and its content. Weijters proposes several possible methods which he combines and uses frequently (this is an iterative process with no explicit ordering):

1. *Performance sequence diagram analysis* [Alves de Medeiros & Weijters, 2009]: Using the PSDA it is possible to examine the variability in the process patterns and many low frequent patterns indicate a high variability in the process.
2. *Log summary*: ProM automatically displays useful information on the home screen under the log summary and dashboard. The number of event classes and their corresponding frequency is easily examined using these screens. Many low frequent events classes and a high number of different event classes are an indication that a "spaghetti-like" model will be obtained when the HM is applied to the event log (in its raw state).
3. *HeuristicsMiner*: The HM is used (with default settings) to examine the layout of the process model. With much variability and different event classes, a "spaghetti-like" model is highly likely and this indicates filtering is required.

These three steps are used as a first option to explore the event log. For hospital data, it is highly likely we are dealing with many redundant and low frequent event classes such as laboratory or administrative tasks and Weijters suggests deleting or aggregating such events. His decision for which events are candidate for deletion or aggregation would depend on discussions with the process owners who are familiar with the process and can indicate points of interest and redundant information (this is an iterative process).

[22] Refer to http://is.tm.tue.nl/staff/aweijters/ (retrieved 8 August 2011) for a complete overview of the research by Weijters.

Process mining methodology

After the dataset has been pre-processed, Weijters uses the following approach with the *HeuristicsMiner*:

1. Mine with default settings to obtain a global process model.
2. Start a new mining tasks with:
    a. *Positive observations* is set to 1. The *Positive observations* parameter has interference with the *Dependency threshold*, since the latter already takes into account the frequency of events and relationships between events (in ProM 6 the P*ositive observations* parameter is omitted).
    b. *Dependency threshold*, *L1L/L2L threshold* and *Long distance threshold* parameters are set to 0,95.
    c. *All-activities-connected-heuristic* is **switched off**.
3. Taking the results under 2 into account, filter the non-connected events and low frequent dependencies using the simple *Event filter*. Take note that this should also be in consultation with the medical experts and process owners. Furthermore, take into account the position of events in the process model under 1. After these decisions points, determine to filter certain events from the event log.
4. Start a new mining task with the filtered event log (obtained after 3). This time all activities connected is **switched on** and positive observations is set to 1.

This is just one approach that hopefully leads to improved process mining results. However, Weijters indicated several other useful approaches that can be considered to obtain an understandable process model. He acknowledged he does not normally use clustering techniques such as *trace clustering*. However, the *LTL Checker* [Aalst, Beer & Dongen, 2005; Beer & Brand, 2007] is an important tool for him as it allows for filtering based on specific and interesting case attributes (e.g. DBC-codes, specific diseases or originators) that are indicated for each patient. Furthermore, the *Dotted Chart Analysis* [Song & Aalst, 2007b]was indicated as an interesting opportunity for filtering as it allows zooming in on PIs with a specific throughput time. Subsequently, these PIs can be extracted for additional analysis. Using this approach it is possible to make a distinction between the different treatments in the event log.

Miscellaneous

- It is possible that it is only known on which day an event is performed. The HM places these activities in parallel in the resulting process model. However, in some (parts of) processes (such as the diagnosis phase) this should not be a problem since a lot of events are indeed executed in parallel in reality. In agreement with the process owners, it is also possible to add artificial time stamps when the exact ordering is known.
- It is recommended to always use artificial start and end tasks.

- The *Flexible HeuristicsMiner* [Weijters & Ribeiro, 2010] is an improvement of the HM and is able to deal with XOR constructs (i.e. visualization) in a more efficient way than the HM. Moreover, it is possible to visualize the individual in- and outgoing arcs from certain events.
- *Long distance threshold* and *L1L/L2L thresholds* are not considered the most important parameters.
- When using the *all-activities-connected-heuristic*, the other parameters are not used entirely anymore. At least all activities are connected, even when the dependency measure is below the threshold. Therefore, each non initial event has at least one in going arc and each non final event has at least one outgoing arc. All other arcs are dependent on the threshold.

Interview with J.C.B. Rantham Prabhakara M.Tech

J.C.B. Rantham Prabhakara M.Tech[23] works as a PhD student at the Department of Mathematics and Computer Science at Eindhoven University of Technology. Under the supervision of prof. dr. ir. W.M.P. van der Aalst he is performing research on process mining at Philips Healthcare with a focus on pattern abstraction as a method to simplify process models.

The interview with Rantham Prabhakara provided insight on a process mining methodology that had not been identified during the other interviews or any of the literature that was analyzed. Therefore, the nature of this interview was slightly different and isolated. However, the results are of potential interest for the process mining methodology in this thesis and (in any case for) future research. Therefore, we decided to include these results in the thesis.

The main focus of the process mining research by Rantham Prabhakara is on processes for medical imaging equipment such as MRI scanners (see Rusu [2010] for an example of his field of work). All actions on and by these machines are logged on a daily basis. As a result, the logs are very detailed, voluminous and heterogeneous. To illustrate the complexity, a horizontal movement of the equipment is not just logged as one single "movement" but consists of many different events that are all logged accordingly. Moreover, the timestamp in the event log may not be reliable as the different compartments of the equipment are not necessarily synchronized and many actions are not instantly logged. Therefore, the resulting data may be even more complicated than the healthcare data that was considered during the six cases and this thesis project.

Pattern abstraction

Due to the complex nature of the data that Rantham Prabhakara works with, simplification of the data is an absolute requirement before the analysis of the process can start. Therefore, Rantham Prabhakara has developed the *Pattern Based Abstraction Viewer* [Rantham Prabhakara & Aalst, 2009] which is implemented in ProM 6.0 and higher. This option allows for semi-automatic abstraction and extraction of patterns in the event log, actions which simplify the corresponding event log. The difference with the methodologies that are found in the six cases is that the algorithm automatically detects patterns in the

[23] Refer to https://venus.tue.nl/ep-cgi/ep_detail.opl?fac_id=92&rn=20080871&voor_org_id=&taal=US (retrieved 8 August 2011) for a complete list of the work by Rantham Prabhakara.

event log and reports them to the researcher. Subsequently, the researcher can perform a series of multiple filtering and modification steps to the different patterns and select the patterns that are to be used for abstraction. These patterns can be assigned a specific name and an event log with the selected pattern abstractions can be stored for future analysis. Using the FM, the new event log can be used to obtain a process model where the pattern abstractions can subsequently be visualized in more detail.

Conventional process mining

Little time during the interview was spent on a process mining methodology more similar to the six cases. However, it was clear that Rantham Prabhakara prefers to use the FM when encountering complex process models. Besides the FM, the HM is considered very useful when dealing with slightly less complex process models. He agrees with Weijters on the specific parameter importance and would approach a mining problem in a similar way. Additionally, he would try to use algorithms such as *trace clustering* to identify specific groups of homogenous process instances for additional process analysis.

Response of dr. A. Rozinat and dr. C.W. Günther (by e-mail)

Dr. A. Rozinat is one of the founders of Fluxicon, the company that has developed the Nitro software tool. She received her PhD at the Eindhoven University of Technology under the supervision of prof. dr. ir. W.M.P. van der Aalst[24] and dr. A.J.M.M. Weijters on the topic of "*Process Mining: Conformance and Extension*" [2010]. Her work has led to numerous publications and additions to the ProM software.

Dr. C.W. Günther is the co-founder of Fluxicon. Günther received his PhD at the Eindhoven University of Technology under the supervision of prof. dr. ir. W.M.P. van der Aalst and dr. A.J.M.M. Weijters on the topic "*Process Mining in Flexible Environments*" [2009]. He has (co-)developed several ProM plug-ins such as the Fuzzy Miner and has published numerous articles on ProM and process mining.

Response by email (quoted):

As for the database construction, I am not sure this is not on a different level: In every project (not just in healthcare), one first needs to determine the scope of the process (start, end, and correlation of cases) and the level of detail (activities) and extract the data accordingly. In a hospital, there are diagnosis/treatment processes as well as organizational processes that might be of interest. And the underlying information system can be expected to vary a lot. Usually, a DBA will actually extract the data in a CSV file or a DB dump, which is then the starting point for me to work with Nitro and ProM.

As for the actual analysis and the pre-processing steps, I realized that this usually happens in a very explorative, interactive way. I personally have not done healthcare projects yet. But with all complex processes there is an iterative analysis process in place that is hard to capture. By looking at the data and trying to understand it, mining a model, filtering, mining a model again etc. one narrows the analysis in a

[24] Prof. dr. ir. W.M.P. van der Aalst is a full professor at Eindhoven University of Technology, Department of Mathematics & Computer Science. He initiated and led the development of the ProM software and has written countless papers and books on process mining and ProM. Refer to http://wwwis.win.tue.nl/~wvdaalst/index.htm (retrieved 8 August 2011) for a complete list of his work.

certain direction that is also dependent on the question of the analysis. For example, is it the goal to discover the main flow of the process? Or are exceptions also important? So, I think that you tackled a very interesting and important topic, but I find it hard to derive my work process in a structured way without making many detailed observations or note-keeping in the course of multiple projects.

Here are some of the tools/plug-ins that I would probably try (without a particular order of importance):

Filtering:

- *Removing less frequent events (advanced event filter)*
- *removing events that are not in many of the cases (advanced event filter)*
- *filtering based on performance (e.g. throughput time)*
- *Remap filter to explicitly map low-level events on more high-level activities*

Clustering:

- *Trace clustering*
- *Sequence clustering*
- *Activity mining*
- *Explicit clustering using LTL checker based on explicit or discovered knowledge*

Mining:

- *HeuristicsMiner*
- *FuzzyMiner*

Checking quality of mined model:

- *Conformance Checker*
- *Fuzzy model conformance indicator*

Appendix F. Development of the patterns

When not specified otherwise, the following can be assumed: 1) the HM with default settings was used to obtain a process model, 2) models are judged by EBP fitness and 3) event logs are created with Nitro. In addition, the patterns are described in the order that they were created and tested.

Information on the datasets that were used during the development of the patterns

AMC: The AMC log was obtained as an .MXML file that contained the raw data (i.e. it still contained all information). As a result there were many irrelevant data attributes, a large number of different event classes, and many laboratory events. Using the HM with default settings on the raw event log resulted in a very large and complex process model with an EBP fitness of 0,48. Among others, the AMC log contained the following:

- 682 process instances
- 43.615 events
- 417 event classes
- 28 originators (department name)

Italian hospitals: For the data originating from the Italian hospitals we considered the treatments data for the development of the patterns. This data was available as a raw .MXML file. Mining on the raw event log yielded a large and complex process model with a EBP of -0,31. Among others, the Italy log contained the following:

- 369 cases
- 6.484 events
- 95 event classes
- 1 originator

GGzE: The GGzE data was used for testing patterns and verifying suspicions that were raised during the development. These results have not been taken into account for most of the patterns.

Data collection

All data that is considered for this research was already available in the forms of .mdb, .csv or .MXML files and therefore there was no need to extract the data from a data warehouse. As a result, this pattern is mostly based on the reports of Mans *et al*. [2008, 2009], Riemers [2009], Ramos [2009], Zanden [2010] and Gupta [2007].

From the six cases and the interviews we learned that it is important to determine what kind of information is required for the analysis before the data collection stage is started. From all six cases and Dongen & Aalst [2005] we have learned that the most important parts of the data for process mining are: *timestamp*, *case identifier* and *activity/event*. In addition, many reports make use of *originators* (could be both/either a requesting and/or acting originator) and for process mining in healthcare *DBC*-

code information is considered very important as it can be used to cluster PIs that have undergone similar care trajectories (thereby creating a more homogenous set of PIs).

It should be considered that it is always possible to discard information anytime during the analysis and that some information can be useful during later stages of the analysis (e.g. for clustering/filtering purposes). In any case, the result of the pattern should be usable data (preferably in any format that can be loaded into MS Access or MS Excel) which in the ideal situation contains information on: *activities/events*, *timestamps*, *originators*, *case identifiers* and in healthcare data *DBC-code* information.

Data transformation (Access)

We have learned that MS Access is the most widely used software tool for data transformation. In Access it is possible to perform several modifications to the data such as the aggregation of events, renaming and filtering, as well as deleting undesired information using standard Access options. However, in order create a .MXML file from an .mdb file, an elaborate and complex method that is described by Mans [20XX] has to be used, which requires both time and some coding knowledge.

As a result, this pattern has not been tested during this research as we learned that the commercial software tool Nitro allows for a much simpler and faster creation of event logs in the .MXML format from a simple MS Excel file. Since the main focus of this thesis was not only on MS Access, it was chosen to incorporate the Nitro pattern, despite the fact that it has not been validated in practice yet (i.e. it is a proto-pattern). As a result of this decision, no additional patterns were created for pre-processing steps in MS Access. However, many of the MS Excel patterns are also (partly) applicable to MS Access.

Creating event log (ProMimport)

ProM*import* is an alternative to the Nitro tool and has the advantage that it is free and can deal with many different file formats as input. As was explained in the pattern *Data transformation (Access)*, we have decided to focus on the possibilities of Nitro as a tool to create event logs and as a result we refer to Mans [20XX] for additional information on the creating an event log with ProM*import*.

Event log

Despite the fact that the event log is not a real pattern (as it symbolizes no action but a static data file), due to its centrality in the pattern network (and future convenience) we have chosen to create a special pattern for the event log, as many of the following patterns are dependent on the event log.

Data transformation (Excel)

MS Excel was used by several researchers [Riemers, 2009; Ramos, 2009; Zanden, 2010] as a means to transform (part of) the dataset. In Excel, modifications of the data are performed easily (which will be elaborated on in other patterns) using simple functions and macros, as well as deleting or adding information to the data (such as adding missing timestamps or deleting the undesired tasks). Moreover, during the development of the patterns it became apparent that the Nitro tool, which requires simple Excel files as input, is much easier to use compared to the elaborate and complex MS

Access/ProM*import* approach to create event logs. Therefore, during this research we focussed on the data transformation in Excel mostly.

From the case studies, it became obvious that several parts of the data are of essential for process analysis in ProM: *timestamp*, *case identifier*, *activity*/*event*, *originator* and additionally *DBC-code* information in healthcare data. As a result, these data attributes should be retained in any study.

Testing on the GGzE data: The data was delivered as a .mdb file and we used the export function in Access to transform the data into a .csv file that could be loaded in Excel. The dataset contained over 50 columns (data objects/attributes) and well over 100.000 rows (ATEs). Moreover, certain abstraction for the events was available (three levels).

The first step was to reduce the number of data objects by creating three datasets; one for each event abstraction level. Initially, only the timestamp, event, case identifier, originator and DBC-codes were retained, as these were the most important data objects and this information was also available for the other two datasets.

Testing on the AMC data: The first step (and also relevant for this pattern) was to remove all irrelevant information from the dataset to reduce the size and complexity. Therefore, it was decided to only keep the following information: timestamp, case ID, activity name, originator (department) and the DBC-code. All other columns (and therefore information) were discarded. The remaining columns were checked for completeness and we identified for four ATEs for which the event and originator was missing. Subsequently, these ATEs have been deleted (leaving 43.611 ATEs). Since the data transformation did not have a significant effect on the data (other than removing 4 ATEs), no new process model was created.

Testing on the Italy data: The first step was to reduce the amount of data as there were many irrelevant data attributes. Eventually, only the following information was kept: timestamp, case ID, activity name, hospital code, event type and type task. Subsequently, all events with the type *complete* and *schedule* were deleted, reducing the number of ATEs to 2.838 and the number of event classes to 45. Unfortunately, there is no DBC-code information or information on the departments that executed the event. Mining on this log produced a process model with a fitness of -1,17. This result is worse than the result of the raw dataset. However, the former model was more understandable compared to the latter.

Creating event log (Nitro)

This pattern is solely based on experience gained throughout this research and information that was retrieved from the Fluxicon website, as it is a new software tool and no scientific research has reported its use as of June 2011 (to the best of our knowledge). However, some information on the tool can be found in Aalst [2011].

Nitro has significant advantages over ProM*import* due to its simplicity, accessibility and the fact that it accepts simple Excel files (.csv) as input. Moreover, Nitro immediately provides some information on the data such as the frequency of events, resources, different process patterns and additionally has the

ability to inspect the process of individual cases. As of July 2011, it is also possible to perform certain pre-processing steps (e.g. filtering and adding artificial start and end events) in Nitro itself. However, it is a commercial tool and therefore requires payment where the ProM*import* tool is free (which could be a drawback for some).

Testing: This pattern was tested with the GGzE, AMC and the Italian data. In Nitro, the five different data objects that are required for process mining are easily appointed (and even recognized automatically). As a result, creating a .MXML file with Nitro is only a matter of minutes at most.

Renaming of events (ProM)

As an alternative to renaming of events in MS Excel (for instance used by Riemers), the ProM tool was used by Mans *et al.* [2008, 2009] as a method to rename certain event classes. In ProM, with the *Remap Element Log Filter* it is easily specified which event class needs to be renamed and what the subsequent new name should be. However, this approach has a major downside as it is only possible to rename one event class at a time. For the AMC data this would mean 170 manual actions to rename all laboratory tasks (see *Renaming of events (Excel)*), which is both time consuming and prone to mistakes.

Testing on the AMC data: To compare both renaming patterns (ProM and Excel) we have used the AMC data that was also used by Mans *et al*. [2009]. It was concluded from the test (and was confirmed by Riemers) that renaming in Excel is preferred over renaming in ProM due to its speed and accuracy. Therefore, no additional tests have been performed with the *Remap Element Log Filter*.

Renaming of events (Excel)

This pattern is not explicitly mentioned in any of the reports (but confirmed by Riemers). However, it is a simple and convenient pre-processing pattern which that reduces the number of event classes and could also serve as an initial step for the aggregation of events. Due to the lack of details on the use of MS Excel as a pre-processing tool in any of the case studies, the coding in the pattern is based on prior Excel experience and information in Excel itself only.

In most reports, the laboratory tasks are candidate for renaming. Therefore, we make simple use of the originator (in this case the department) to identify and select activities for renaming. Using special codes we search for originators that match the specified name. Whenever such an originator is identified, the name of corresponding events is renamed into the predetermined new name. For instance, all laboratory tasks (e.g. (bio)chemical tests) executed by the originator *Laboratory* are renamed into the generic name *Laboratory*. As a result, the number of event classes in the dataset can be dramatically decreased. In healthcare data, many different laboratory tasks can be present (up to and over 100 different classes) and since they are not of interest to the main behaviour of the process, it makes sense to rename them.

Renaming in Excel proved to be easier than renaming in ProM, due to the fact that many events can be renamed at the same time (using clever coding) and little manual effort is required (which was also confirmed by Riemers during the interview).

Testing on the AMC data: We used the dataset obtained after the *Data transformation (Excel)* pattern. In the dataset, many laboratory tasks (such as *Glucose* and *Calcium*) were present (i.e. over a hundred different event classes). In order to reduce the number of event classes it was chosen to rename all laboratory tasks to the department that executed them, in this case *Algemeen Lab Klinische Chemie* (as was also performed by Mans *et al.* [2009] and Ramos [2009]). To identify events that needed to be renamed we made use of the originator, which in this case is *Algemeen Lab Klinische Chemie*. If the originator was positively identified, the name of the corresponding event would be transformed into *Algemeen Lab Klinische Chemie*. As a result, this pattern reduced the number of event classes from 417 to 294. The resulting process model however was still large and complex model with an EBP fitness of 0,49.

Additionally, we have also created an event log for which all events are renamed to the department that had executed them. As a result, only 28 event classes were retained (as there are 28 originators). Process mining led to a process model that was much smaller and less complex compared to the previous model, and it had a EBP fitness of 0,63. From here on, we will use both datasets to illustrate the development of the patterns.

Testing on the Italy data: There were only 45 event classes present in the dataset. Additionally, there was no information available on the originators, nor was it possible to determine the significance of the individual event classes (as there were no obvious insignificant events such as laboratory tasks). As a result, we have decided not to perform any renaming for the Italian dataset.

Aggregation of events (ProM)

Mans *et al.* [2008 & 2009] used the ProM tool for the aggregation of events and this is easily performed with the *Repetitions-to-Activity filter*. It automatically produces "*an event log where all direct repetitions of the same audit trail entry are replaced by one 'start' event with the time stamp from the first occurrence and a 'complete' event with the time stamp of the last occurrence in this sequence of repetitions*" [ProM 5.2 software].

Despite the ease with which events can be aggregated there are several potential drawbacks to this pattern. For instance, the filter considers all event classes for aggregation, i.e. it cannot be specified which event classes need to be aggregated. As a result, it is not possible to aggregate the laboratory tasks only (which is possible with the *Aggregation of events (Excel)* pattern). Second, the filter in ProM also aggregates events that directly follow each other in the event log, but occur on a different day (this is can be considered as an option in Excel). Third, ProM automatically creates both a start and end event type for each event in the event log. As a result, depending on the researcher's interest, these issues can be major drawbacks.

Testing on the AMC data: Testing this pattern on the AMC data resulted in 19.724 ATEs. The examct same result was obtained when we used the *Aggregation of events (Excel)* pattern with similar specifics for aggregation. Therefore, the *Aggregation of events (ProM)* pattern can be considered reliable, but it allows for much less freedom than the alternative pattern.

Aggregation of events (Excel)

In most of the case studies that have been considered during this thesis [Mans *et al*., 2008 & 2009; Riemers, 2009; Ramos, 2009; Zanden, 2010] it is mentioned that certain events need to be aggregated. Riemers has noted during the interview that he chose for aggregation in MS Excel over ProM (see *Aggregation of events (ProM)*) because of its simplicity and increased degree of freedom (as aggregation can be decided on many factors). However, there is little to no details provided on how the actual aggregation in Excel was performed.

As a foundation, we defined events that are candidate for aggregation as multiple identical events that occurred for a single patient on the same day. To achieve aggregation, we created a simple formula (see the actual pattern in Appendix I for this formula) to identify these "double" events. Subsequently, those events were marked for deletion and a macro[25] was found on the internet to delete the rows containing the marked events.

Testing on the AMC data: The event logs that resulted from the *Renaming of events (Excel)* pattern served as the input for the tests of this pattern. For the event log with all renamed events we applied aggregation for all event classes. The result was a reduction from 43.611 to 13.129 ATEs, indicating that over 30.000 double ATEs had been removed. The resulting process model was understandable and had a EBP fitness of -0,35. For the log with only the renamed laboratory events, aggregation of the laboratory events led to a reduction to 20.778 ATEs. The resulting process model was large and complex and had a EBP fitness of -0,04.

Testing on the Italy data: For the event log that was created after the *Data transformation (Excel)* pattern, we decided to aggregate for all "double" events that occurred for the same patient on the same day. The result was an event log with 2.475 ATEs and a process model with a EBP fitness of -1,54.

Export MXML

After renaming, aggregation, clustering or any other steps in ProM that have modified the event log, it can be important to save the modifications for future analysis. ProM allows to export the modified event log using the *Efficient MXML.GZ Export* under the Exports tab. Although not explicitly mentioned in any report, this pattern is an important tool for saving time and effort.

Attribute value filter

The *Attribute value filter* in ProM can be used as a method for filtering process instances that do not contain a specific value for a specific data attribute (such as a hospital or DBC code) and can therefore be used as a method to create a more homogenous event log. However, there are some major reliability problems with this filter (as was also stated by Mans during the interview). To illustrate, the filter did work in the Italian hospital event log when we filtered for hospital codes. In this case the filter produced similar results compared to the *Clustering (Excel)* pattern. However, in the AMC event log we were

[25] Refer to http://www.rondebruin.nl/delete.htm (retrieved 8 August 2011).

unable to filter for DBC-codes (or many of the other data attributes) using the *Attribute value filter*. Due to this unreliability we have decided not to create a pattern for this filter.

Clustering (Excel)

In many reports [Riemers, 2009; Ramos, 2009; Zanden, 2010; Gupta, 2007] clustering of patients is based on the DBC-code information in the data. Patients with similar DBC-codes are more likely to have similar processes and therefore clustering that is based on such characteristics increases the chances for more homogenous event logs and hence improved process models. In most hospital data, different DBC-codes can be found on all the DBC levels (e.g. diagnosis and treatment) and as a result allow the researcher to create clusters with different levels of abstraction.

Clustering based on data attributes is performed easily in Excel with the macro that was introduced in the pattern *Aggregation of events (Excel)*. Assuming that no additional preparation of the data is required, the researcher can simply specify the designated target column and the desired criterion for deletion (refer to the actual pattern in Appendix I).

It should be noted that before any clustering is performed, the researcher should confirm that the attribute that is used for clustering has the same format for all PIs. Additionally, it is recommended to avoid clustering in the early stages and perform renaming and aggregation of events first. This will avoid the effort of having to redo the same tasks for many clusters.

Testing on the AMC data: Information on the treatment programs (DBC) M11 to M16 was available in a separate column. With the pattern it was easy to extract the patients with similar DBC-codes (six in total). Each cluster contained a similar number of PIs compared to the method by Ramos [2009] with the *LTL checker* in ProM. The results for both the event log with only the renamed and aggregated laboratory tasks and the event log with all renamed and aggregated events are depicted in table F.1 as “a” and “b” respectively.

Testing on Italy data: In the case of the Italian hospital data, one of the goals was to visualize differences between the four hospitals [Mans *et al.*, 2008]. Therefore, to test this pattern we used the hospital code as input for clustering in Excel (moreover, there was no DBC-code information present). As explained, this pattern yielded the same results as the *Attribute value filter*. The results of the clustering are presented in table F.2. In addition, there were several PIs that did not have any information on the hospital code and as a result these were not included in any cluster.

	# PIs	# Events	# Event classes	EBP
M11a	134	4.255	111	0,52
M11b	134	2.452	18	0,31
M12a	8	168	58	0,41
M12b	8	118	14	0,15
M13a	296	10.172	215	0,04
M13b	296	6.593	25	-0,46
M14a	93	2.296	131	0,2
M14b	93	1.433	18	-0,32
M15a	12	223	48	0,55
M15b	12	144	10	0,49
M16a	139	3.658	119	-0,07
M16b	139	2.383	19	0,06

Table F.1 (left): Results of the Clustering (Excel) pattern for the AMC data.

	# PIs	# Events	# Event classes	EBP
Hospital 1	95	876	34	-0,52
Hospital 2	73	417	29	-0,70
Hospital 3	101	785	31	-0,60
Hospital 4	96	307	24	-1,43

Table F.2 (above): Results of the Clustering (Excel) pattern for the Italy data.

Add artificial start/end task

Common in many process mining studies, and always used by Weijters and Mans in their research, the option to add an artificial start and end task to a process model is an easy but effective method to introduce more structure in a normally chaotic process model. For instance, it allows for a much easier identification of the start and end of the process, a particular problematic task in larger and more complex processes. In addition, (as is presented below) these tasks can significantly increase the fitness measure.

Testing on the AMC data: When we added an artificial start and end task to the event log with all renamed and aggregated events we drastically improved the EBP fitness measure to 0,66. Moreover, this makes identification of the process flow more visible. For the event log with the renamed and aggregated laboratory tasks we achieved a fitness of 0,64. However, the process model was very large and complex (i.e. 294 event classes). For the M13 cluster with all renamed and aggregated events we achieved an increase in fitness from -0,46 to -0,09. For M13 with only renamed and aggregated laboratory tasks, an increase from 0,04 to 0,51 was obtained. For M16 with all events renamed and aggregated we achieved an increase from 0,06 to 0,51 and for M16 with only the laboratory tasks renamed and aggregated, an increase from -0,07 to 0,56.

Testing on the Italy data: After addition of the artificial start and end tasks to the aggregated global event log the fitness improved from -1,54 to 0,49 and produced a moderately but understandable process model.

For the hospital clusters, addition of artificial start and end events resulted in process models with a fitness of 0,13, 0,5, 0,48 and 0,64 for hospitals 1 up to 4 respectively. All process models were much easier to understand compared to the models without the start and end events.

Enhanced event log filter

Despite applying the previously developed process mining patterns, it is possible that a lot of different event classes still remain in the event log, thereby causing complex process models. In many cases,there

are numerous event classes with a high frequency in the total event log and/or as well as a high presence in the all process instances. However, there are also event classes with a low frequency in the total event log and/or do not occur in many process instances. In many cases, when an event class is highly frequent in the event log, it also occurs in many of the process instances, and low frequent events occur only in a small number of process instances (i.e. both frequencies are highly correlated). As a result, in several reports and by several process mining experts it is suggested to use the *Enhanced event log filter* to remove these low frequent event classes. With this option it is possible to filter for low frequent events, as well as events that are not in many PIs. There is no "optimal" setting, but a 1% minimum for occurrence in PIs is used by several researchers [Riemers, 2009; Ramos, 2009].

Testing on the AMC data: Note that all event logs now also feature the artificial start and end tasks. From the summary of the event log we identified many low frequent events in the process. Therefore, we filtered for events with an occurrence of less than 1% in the total number of process instances.

- For the event log with all renamed and aggregated events this resulted in an understandable process model with 21 events and a fitness of 0,66.

- For the event log with only renamed and aggregated laboratory events, we obtained a process model with 111 events (from 294) and a fitness of 0,63. However, this model was very large and complex. Therefore, we added the filter for relative frequency and set the parameter to 1%. As a result we obtained a process model with 17 event classes and a fitness of 0,78. Subsequently, the model was much easier to understand.

- For the DBC-code M13 with the renamed and aggregated laboratory events, filtering yielded a complex process model with 116 event classes and a fitness of 0,51. After filtering for the low total frequency (1%) the fitness improved to 0,62 and the model was more understandable (19 event classes). The log with all renamed and aggregated events yielded process models a EBP of -0,1 and 24 event classes. After both filters were applied the fitness increased to a EBP of 0,1 and the model contained 13 event classes. Both models were understandable.

- For the DBC M16 with renamed and aggregated laboratory events, filtering yielded a complex process model with 87 event classes and a fitness of 0,62. After additional filtering for the low total frequency (1%), the number of event classes decreased to 15 and the resulting process model had a fitness of 0,79. The log with all renamed and aggregated events resulted in a process model with 17 event classes and a fitness of 0,5.

Testing on the Italy data: When filtering for event classes that occur in less than 1% of the PIs (resulting in 37 event classes), the fitness of the global aggregated model was 0,46 (which is slightly lower than the model without filters). After additional filtering for the total frequency of less than 1%, we obtained a smaller (and understandable) model with 19 event classes and a fitness of 0,47.

For the hospital clusters we first filtered for events that occured in less than 1% of the PIs. However, it was observed that this did not result in any change in the fitness and process model layout for hospitals 1 and 4. Therefore, we also filtered for events that occurred less than 1% in total. This resulted in

process models with a fitness of 0,7, 0,49, 0,48 and 0,63 for hospitals 1 to 4 respectively. In addition, for hospitals 2 and 4, after the first filter we obtained models with fitness measures of 0,49 and 0,64 respectively. As can be concluded, in this case, filtering does not significantly change the fitness for three out of four hospitals (it does for hospital 1). In addition, all models were small to medium in size and reasonably understandable.

HeuristicsMiner

The *HeuristicsMiner* [Weijters *et al.*, 2006] is the most predominantly used mining algorithm for the discovery of a process model. One of the most important reasons to use the HM is the fact that it can deal with noise in the event log, which is an inevitable complication when real life data is used. Moreover, the algorithm can be used to mine the main behaviour in an event log and can deal with loops, hidden activities and long distance dependencies. Furthermore, the representation of the process model by the so called *HeuristicsNets* is easy to understand in comparison with for instance a Petri net [Murata, 1989], which features different representations for places and tasks in the model and requires more knowledge of the modelling language.

The HM plug-in in the ProM tool is featured with many different parameters (eight parameters and three additional options) that can be tweaked by the user to focus on specific frequent or non-frequent behaviour. During the multiple case study analysis we have learned that the HM is mostly used with the default parameter settings and researchers only incidentally took alternate settings of the HM into consideration. The report by Gupta [2007] is an exception to this problem. However, it must be noted that the focus of that research was to identify problems with the HM algorithm and therefore it differs from the other papers that have been analyzed (which focus more on the application of the general concept of process mining in a healthcare setting and its corresponding results).

As was learned from the case studies, the interview with Weijters and research papers on the HM, different parameter settings can have a significant effect on the resulting process model. During the development of the patterns we have so far only focussed on the default parameter settings. However, for the development of the HM pattern we have also investigated the approach that was suggested by Weijters (section 4.5), where the different parameter setting are taken into consideration and are used to identify event classes with few dependencies and show low frequent behaviour. In addition, we have also investigated the use of the positive observations in the approach by Weijters, with values of 1 and 10.

As we have not modified the event log since the previous pattern, for the *HeuristicsMiner* pattern we now only consider the approach by Weijters for testing. In the eventual pattern both details on the default use of the HM as well as the approach by Weijters are included.

Testing on the AMC data:

- For the event log with all renamed and aggregated events (with artificial start and end event AND filtering), the approach suggested by Weijters resulted in an event log with 14 event classes and a small process model with an EBP of 0,62. This implies that the model does not necessarily

improve anymore. When we specified the positive observations to 10, we obtained a model with a EBP of 0,30.

- When Weijters' approach was tested on the event log with all renamed and aggregated events, with artificial start and end tasks but without filtering by the Enhanced event log filter, we obtained a process model with 18 event classes and an EBP of 0,34. However, when the positive observations was set to 10, we obtained a model with an EBP of 0,71. This result again shows the difficulty of finding the right parameter settings.

- When tested on the event log with renamed and aggregated laboratory tasks but without filtering, we obtained a process model with all 294 event classes. As a result, deselecting all unconnected events (over 200) is an approach that is both time consuming and prone to mistakes. Therefore, it can be concluded that the approach suggested by Weijters is not convenient for event logs with a high number of event classes and the *Enhanced event log filter* is the preferred option in such a case.

- For the DBC M16 event log with all events renamed and aggregated, the Weijters approach with positive observations set to 1 produced a process model with 14 event classes and an EBP of 0,33. For positive observations set to 10, the EBP increased to 0,51. When the HM approach was tested on the filtered set for M16 we achieved process models with 0,32 and 0,51 for 1 and 10 positive observations respectively.

Testing on the Italy data: When the HM approach was tested on the global aggregated set, this resulted in a process model with 16 event classes and a fitness of -0,31. With positive observations set to 10 a process model with a fitness of 0,56 and 16 event classes was obtained.

Trace clustering

Trace clustering was used by Mans *et al*. [2009], Riemers [2009] and Ramos [2009], and recommended by Rozinat and Gunther as a method to create more homogenous sets of process instances, using certain mathematical algorithms as opposed to manual clustering based on predetermined data attributes (such as in *Aggregation of events (Excel)*). In all cases the self organising maps algorithm in combination with Euclidean distance have been used and only Ramos has reported the use of different clustering algorithms. However, there are several drawbacks to trace clustering.

First of all, when dealing with large (raw) event logs it takes a considerable amount of time to cluster all process instances (several hours up to several days). Additionally, it became apparent that clustering on large and raw event logs is not very useful as in most cases only a few very large clusters were observed (and additionally a few very small clusters). These clusters did not result in process models that were any better than the process models before trace clustering had been used. Accordingly, during this thesis we have learned that it makes more sense to first perform manual clustering based on specific data attributes as a means to obtain more homogenous datasets. Subsequently, if the researcher is interested in behaviour within a such a cluster obtained after manual clustering, it can be sensible to use additional algorithms such as trace clustering.

These findings also account for the other possible clustering algorithms in trace clustering. The problem is that the results are not very usable when applied to large event logs and certainly are not superior to the manual clustering based on data attributes, which is faster and more accurate. Additionally, with manual clustering it is apparent on what foundation the PIs have been clustered. With trace clustering, no specific characteristic on which clustering was based can be determined. In his research, Ramos [2009] has tested numerous trace clustering algorithms and only found a limited number of process models useful.

Testing on the AMC data:

- With KMEANS clustering and Euclidean distance, for the global event log with all event classes renamed and aggregated, three clusters were obtained with 172, 452 and 58 cases. However, the last cluster only had cases with one even class and therefore it was considered not significant to produce a process model. The remaining two clusters resulted in process models (after *Add artificial start/end task* and *Enhanced event log filter*) with an EBP of -0,2 (24 event classes) and 0,54 (16 event classes) respectively. Both models were small and understandable.

- Using SOM with Euclidean distance on the AMC event log for which all events were renamed and aggregated resulted in five clusters (after running for four hours):

    - Cluster 0 with 26 PIs and 7 event classes. Without artificial start and end tasks, the fitness was -0,54 and 0,68 with both tasks added. After filtering the fitness was 0,68.

    - Cluster 1 with 520 PIs and 28 event classes. Without artificial start and end tasks, the fitness was -0,32 and 0,69 with both tasks added. After filtering the fitness was 0,69.

    - Cluster 2 with 35 PIs and 3 event classes. Without artificial start and end tasks, the fitness was -0,11 and 0,96 with both tasks added. After filtering the fitness was 0,96.

    - Cluster 3 with 43 PIs and 1 event class. Without artificial start and end tasks, the fitness was -0,1 and 0,98 with both tasks added. After filtering the fitness was 0,98.

    - Cluster 4 with 58 PIs and 1 event class. Without artificial start and end tasks, the fitness was 0,98 and 0,98 with both task added. After filtering the fitness was 0,98.

Testing on the Italy data: During the previous patterns we have learned that the process model for hospital 1 had a low fitness. Therefore, we tried to improve the results for hospital 1 with trace clustering. Using SOM with Euclidian distance and default settings we obtained seven clusters. There were four clusters with only 1 PI, one cluster with 5 PIs, one cluster with 60 PIs and one cluster with 26 PIs. Since we are interested in the global behaviour of patients in hospital 1, we choose the last two clusters for additional analysis.

- For the cluster with 60 PIs (30 event classes), mining on the set without artificial tasks and filters resulted in a model with a fitness of -0,4, which is an improvement (from -0,52). After the

artificial tasks had been added, the fitness significantly improved to 0,44. After double filtering the fitness decreased to 0,22 (for 15 event classes).

- For the cluster of 24 PIs, a process model with a fitness of -0,76 was obtained. After the addition of artificial tasks, the fitness improved to -0,22. Filtering of events did not have a positive effect, as the fitness decreased to -0,28.

Trace clustering on the global aggregated set (with SOM and Euclidian distance) resulted in five clusters with 1, 1, 4, 16 and 347 PIs. Since we are interested in the global behaviour of patients, we only choose the largest cluster for additional analysis (as the remaining clusters represented too few patients to be interesting for global behaviour). Mining resulted in a process model with a fitness of -1,51 (45 event classes). After the addition of artificial events, the fitness improved to 0,5. However, the model is still quite large (as no event classes were removed) and therefore difficult to understand. After filtering, the fitness dropped to 0,47 (37 event classes). After double filtering the fitness remained 0,47 (17 event classes). Therefore, in the case of applying trace clustering on the global log, the results do not significantly improve. Moreover, the clusters represented patients from all four hospitals and therefore no clear characteristics could be identified that were common for patients that belonged to the same cluster (as would be the case for manual clustering)

Appendix G. Reproducing process mining results in SHARE

Reproducing the process mining results by Mans et al. [2009] and Ramos [2009]

Information on the dataset: The AMC global data that was available to us contained 682 cases, 43.615 ATEs, 417 event classes and 28 originators. Furthermore, many data attributes such as DBC-code information (there are six DBC-codes) were present. In our dataset, the administrative tasks had already been removed (this was mentioned for removal by Mans *et al*.). However, the data that was available to us complied only to the set that was used by Ramos, and it is an extension of the data that was used by Mans *et al.* Compared to the latter, not only was there an increased number of cases, also more event classes were present. As a result, the data that was available to us is more complex than the data that was used by Mans *et al*., however it is similar to the data used by Ramos.

Link to SHARE demo: http://is.ieis.tue.nl/staff/pvgorp/share/?page=ConfigureNewSession&vdi=XP-TUe10G-WithOfficeAndAcrobat_ProM_5.vdi

Links to screencasts:
Part 1: http://www.screencast.com/t/BqLqUqbm
Part 2: http://www.screencast.com/t/KSm4gCUA
Part 3: http://www.screencast.com/t/2uYvPdqU
Part 4: http://www.screencast.com/t/wpxMLN0bl
Part 5: http://www.screencast.com/t/sX8a5MoA5J
Part 6: http://www.screencast.com/t/ZgGRBHQGldPy

Reproducing the results in SHARE:

1. Using the *Export .MXML* pattern we transformed the .MXML file to a .csv file.

2. We have loaded the .csv file in MS Excel (*Data transformation (Excel)* pattern) via *import* and *from text*, and used *delimited* and *semicolon* and then pressed finish.

3. It is noted that the timestamp is presented in different formats. For the transformation of the Excel file to .MXML in Nitro, it is required that all timestamps have the same format. Therefore, we changed the format using the *text to columns* option, selected *delimited*, *space* and selected the option *DMY*. Accordingly we changed the column containing the timestamp to *short date* and subsequently to *MM/DD/YYYY HH:MM*.

4. There are some ATEs with empty tasks that need to be removed (i.e. 4 ATEs). In the SHARE demo we used manual filtering, but this could also be achieved with the *Clustering (Excel)* pattern (by removing cells that contain the value zero)

5. We checked for completeness of the data and noticed that the DBC-code column (J, diagnosecode) was not complete as there were only 25.380 values. The remaining 18.231 values are in column Y called *diagnosecode2*. To combine them into a new singe column we inserted a new column at D (called DBC). Using the formula *=IF(K2=0,Z2,K2)* for cell B2, we subsequently

pasted the formula for every cell in column B. Next, we pasted the new values over the old values in column E with *paste special* (as values).

6. Subsequently, we deleted all columns except *caseID*, *DBC*, *taskID*, *originator* and *timestamp*.

7. We renamed all events to the originator that executed the task (as Ramos did and Mans *et al.* did up to some part) by copying the originator column and paste it over the taskID column (it is also possible to use the pattern *Renaming of events (Excel)* for this).

8. We aggregated events according to the *Aggregation of events (Excel)* pattern (i.e. multiple event per patient per day) which left 13.129 ATEs.

9. The event log was created using the *Creating event log (Nitro)* pattern.

10. We added an artificial start and end task according to pattern *Add artificial start/end task*.

11. Using the *HeuristicsMiner* with default settings resulted in the moderately understandable process model presented in figure G.1, with an EBP fitness of 0,66. Compared to the model by Mans *et al.* (in figure G.2) our model is smaller, which is due to the slightly different pre-processing steps (which are not documented accurately enough by Mans *et al.* in order for us to exactly reproduce the results).

12. Trace clustering was used by both Mans *et al.* and Ramos as a method to create clusters of patients with more similar processes. Using the *Trace clustering* pattern with default settings (SOM), we obtained a different result compared to both researchers (see Appendix B) as we had five clusters with 26, 35, 43, 58 and 520 PIs. Again this difference can be contributed partly due to the fact that there is a difference between the pre-processing steps. Moreover, Ramos has only used care trajectories that had started within the timeframe of the event log, information that was not available to us (and therefore we used all cases).

13. When we used the *HeuristicsMiner* with default settings on the largest cluster (and after addition of artificial start and end tasks), we obtained a large process model (figure G.3) with a EBP fitness of 0,69. However, compared to the models of Mans *et al*. (figure G.4) and Ramos (figure G.5), our model is much larger and more complex. Once again, this is mainly due to differences in the pre-processing steps.

14. The *Fuzzy Miner* was briefly tested by both researchers. However, not much attention was devoted to it and therefore we did not investigate it any further.

15. Ramos has performed preliminary analysis for all six DBC-codes, but eventually used the M16 for further analysis. Since the process mining methodology for each DBC-code cluster is similar, we only focussed on M16 as well.

16. Using the *Clustering (Excel)* pattern (for the data that was obtained after step 8) we clustered for the M16 DBC-code which contained 2.384 ATEs.

17. Using the *Creating event log (Nitro)* pattern we created an event log and noticed that there were 139 cases left, compared to Ramos' 85 cases (which is most likely due to the fact that Ramos had only used closed care trajectories).

18. Using SOM with Euclidean distance (*Trace clustering* pattern) we obtained six clusters for the M16 patient group with 1, 5, 5, 6, 14, 108 PIs. Ramos obtained three clusters and used the largest one with 57 PIs for additional analysis. As a result, we decided to focus on the largest cluster as well. Using the *Export .MXML* pattern we created an separate event log for this cluster.

19. We added artificial start and end tasks (*Add artificial start/end task* pattern) to the M16 event log with 108 PIs and used the *HeuristicsMiner* to create a process model. However, as Ramos did, we changed the *AND-threshold* to 10 and the *L1L threshold* to 0,999. This resulted in the process model depicted in figure G.6 with an EBP of 0,90. Our model is slightly larger compared to the model of Ramos (figure G.7), but our fitness is slightly higher than his 0,84.

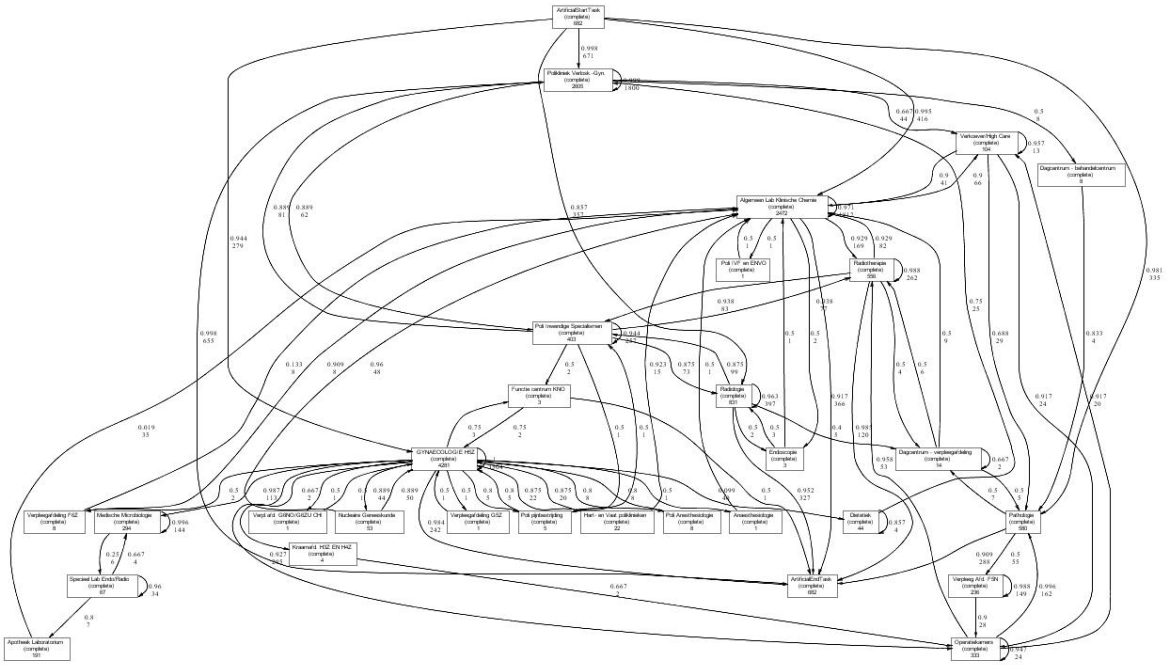


Figure G.1: Global process model of the AMC data.

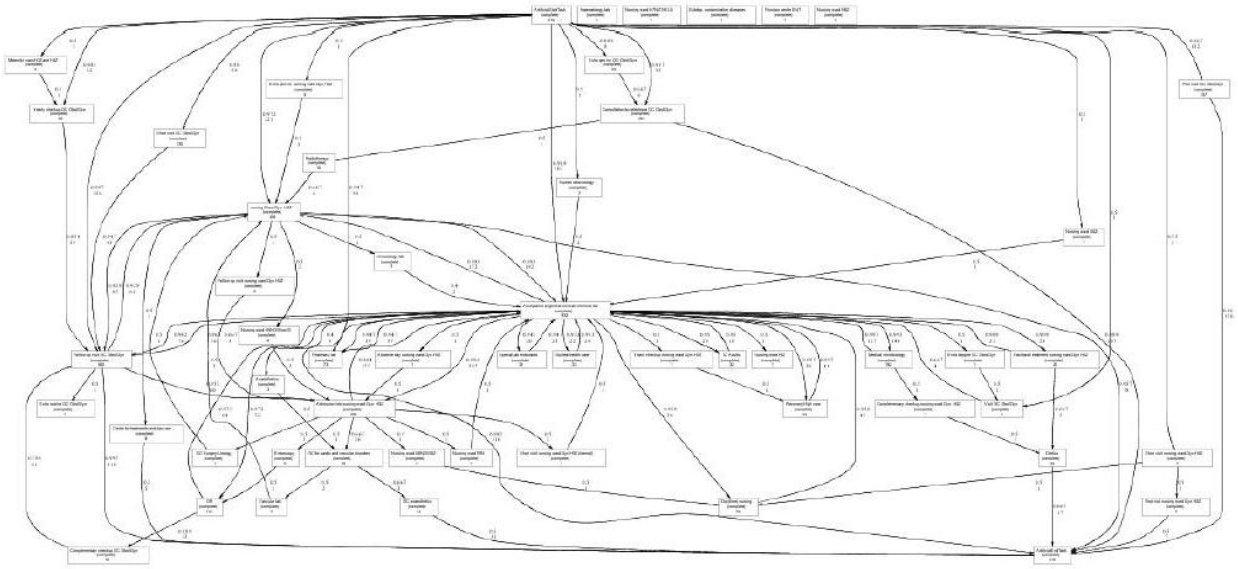


Figure G.2: Global process model of the AMC data by Mans et al. [2009].

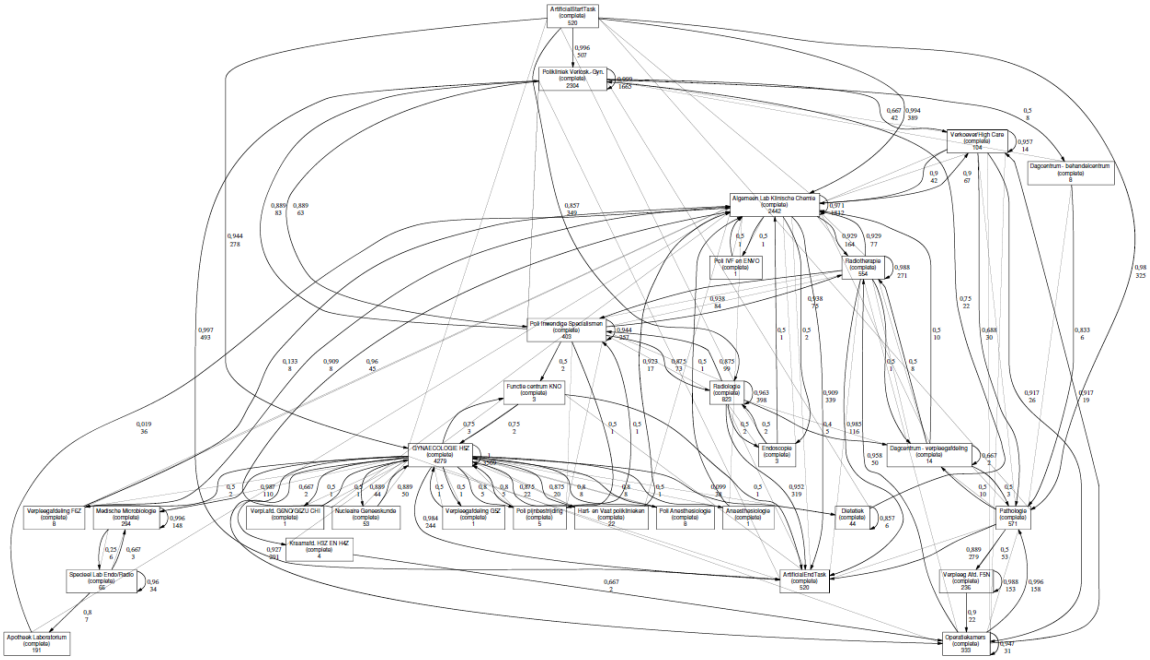


Figure G.3: Process model of a cluster of 520 PIs from the AMC data.

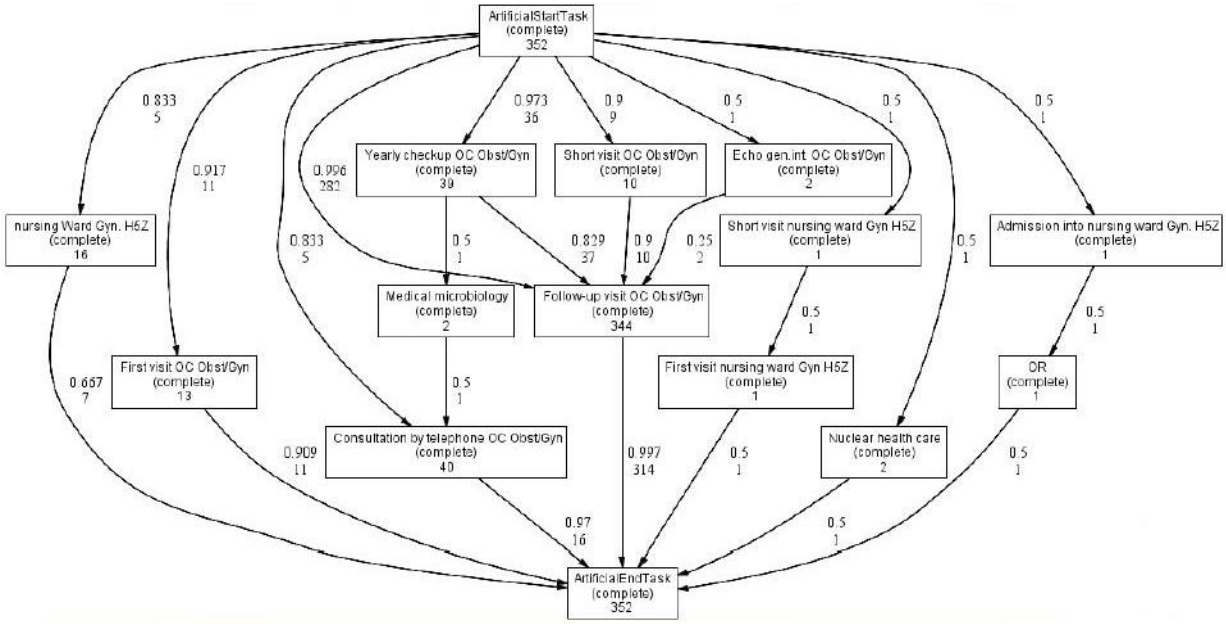


Figure G.4: Process model of a cluster of 352 PIs from the AMC data by Mans et al. [2009].

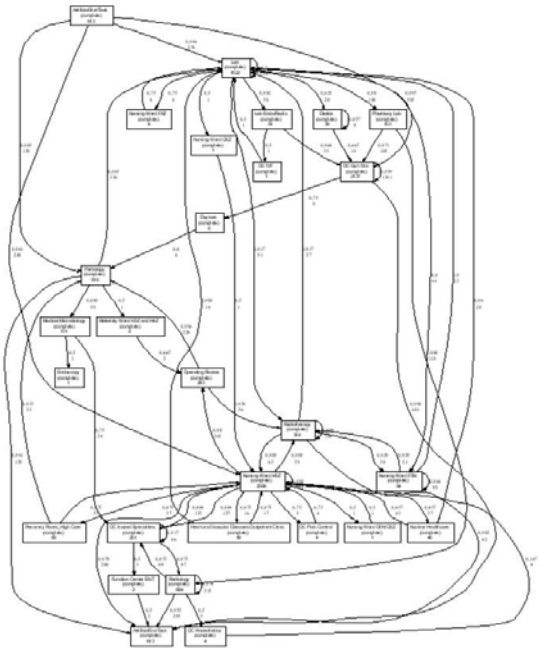


Figure G.5: Process model of a cluster of 613 PIs from the AMC data by Ramos [2009].

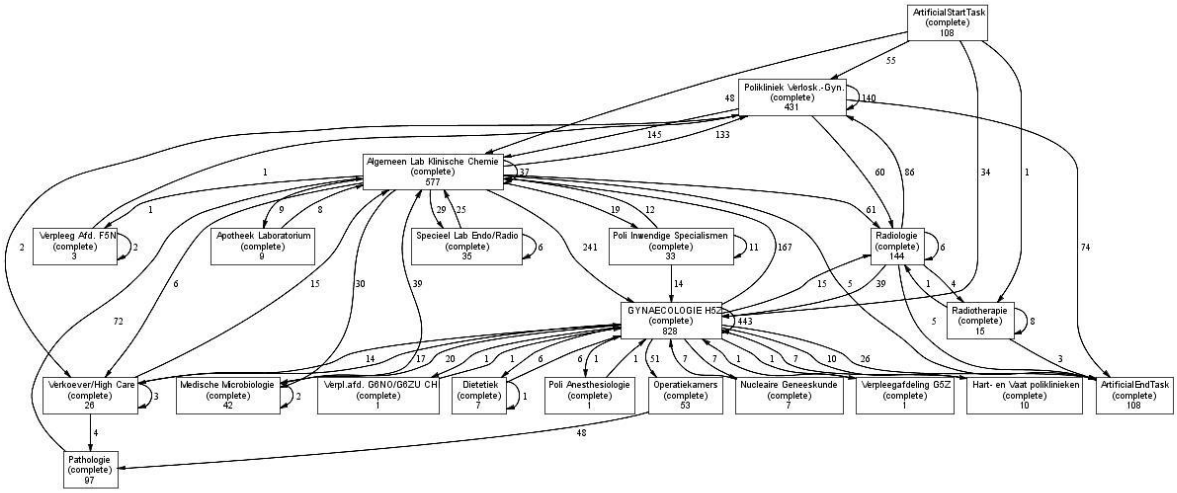


Figure G.6: Process model of the AMC M16 cluster with 108 PIs.

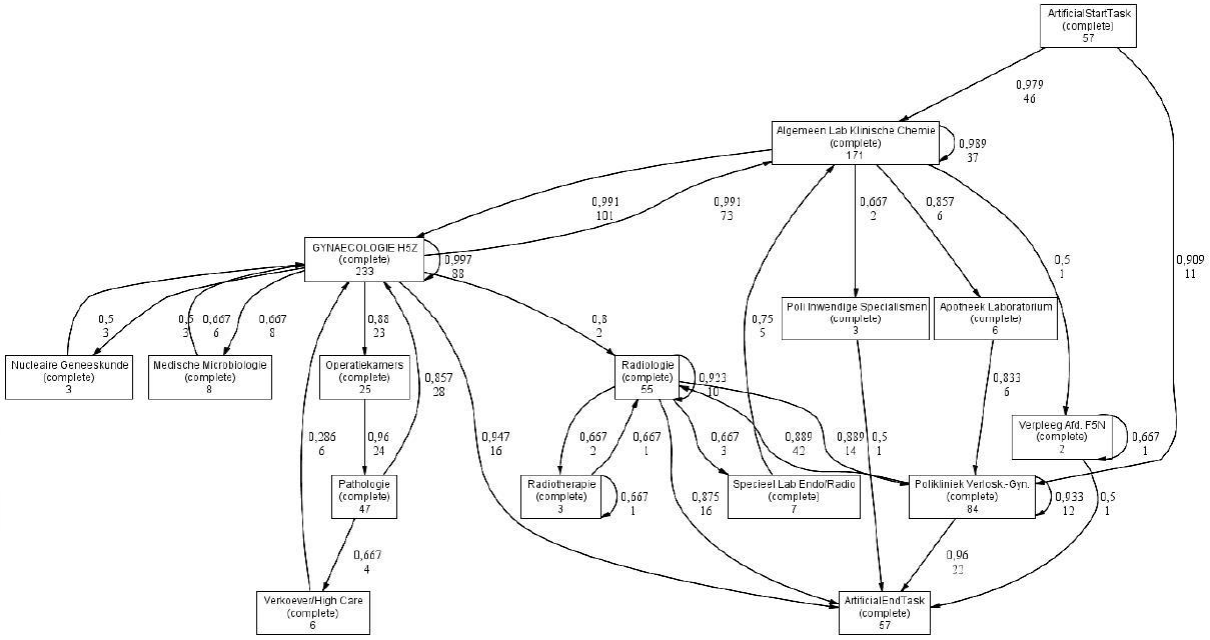


Figure G.7: Process model of the AMC M16 cluster with 57 PIs by Ramos [2009].

Reproducing the process mining results by Zanden [2010]

Information on the dataset: The GGzE data was available to us as a MS Access database. To test the process mining patterns, we exported the data as a .csv file that could be loaded into MS Excel. The data contained 83.920 ATEs and pertained to 3.512 patients. Considered by Zanden were the level 1 activities (8 event classes) and the level 3 activities (61 event classes). Additionally, it was also indicated whether the care trajectory had been finished or was still open (at the time of the data collection) and the dataset included many attributes on patient characteristics and originators. In the SHARE demo only a part of the dataset is presented due to privacy reasons. Moreover, we have anonymized the patient reference numbers.

Link to SHARE demo: http://is.ieis.tue.nl/staff/pvgorp/share/?page=ConfigureNewSession&vdi=XP-TUe10G-WithOfficeAndAcrobat_ProM_2_2.vdi

Links to screencasts:
Part 1: http://www.screencast.com/t/Cph8KpNz4Hi
Part 2: http://www.screencast.com/t/jpPMVi0d
Part 3: http://www.screencast.com/t/SdSI1HhVTiZr

Reproducing the results in SHARE:

1. The first step was to delete all irrelevant data from the dataset (*Data transformation (Excel)* pattern). This means, retaining only the *timestamp*, *reference number* (i.e. patient ID), *activity name_level_1* and *3* and *care trajectory final date*. The originators were not used by Zanden and therefore were removed as well. Before all irrelevant data was removed, it was confirmed that the relevant part of the data was complete (i.e. no missing data). The removal of the data attributes did not affect the resulting process model.

2. (SHARE demo starts here) In her research, Zanden only uses closed care trajectories. Therefore, we used the *Clustering (Excel)* pattern to filter for empty cells in the column *Care trajectory final date*. After filtering, only 50.351 ATEs and 2.087 cases were left.

3. Aggregation of the events was performed for each patient per activity per day using the *Aggregation of events (Excel)* pattern. This left 47.301 and 49.824 ATEs for levels 1 and 3 respectively. Additionally, the sets contained 8 and 57 event classes respectively.

4. Using the *Creating event log (Nitro)* pattern we created .MXML files for the level 1 and 3 activities.

5. Using the *HeuristicsMiner* pattern with default settings (as did Zanden) we first performed process mining on the level 3 activities event log. This resulted in the process model that is depicted in figure G.8. This process model is both large and very complex and not easily understood. This result is similar to the model that was obtained by Zanden, which is illustrated by figure G.9. Our model has an EBP fitness measure of -0,68 and unfortunately Zanden did not mention any fitness measure.

6. Using the *HeuristicsMiner* pattern with default settings we obtained the process model that is depicted in figure G.10. This model has the following fitness measures: Proper completion and StopSemantics of 0,0, ContinuousSemantics of 0,359, ImprovedContinuousSemantics of 0,597 and ExtraBehaviourPunishment of 0,572.

7. Zanden used the HM with default settings on the event log with the level 1 activities. This resulted in the process model that is illustrated in figure G.11 with the following fitness measures: Proper completion and StopSemantics of 0,0, ContinuousSemantics of 0,359, ImprovedContinuousSemantics of 0,583 and ExtraBehaviourPunishment of 0,559.

Figure G.8: Process model of the level 3 activities.

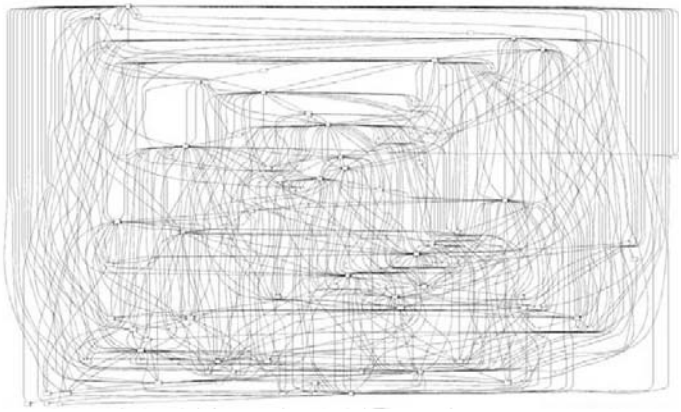


Figure G.9: Process model of the level 3 activities by Zanden [2010].

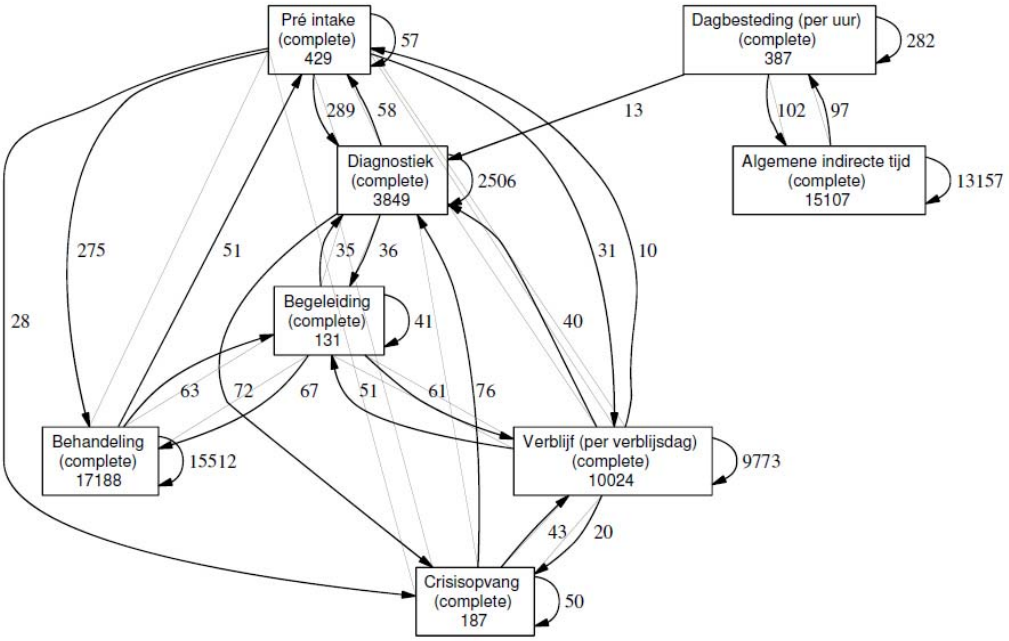


Figure G.10: Process model of the GGzE level 1 activities.

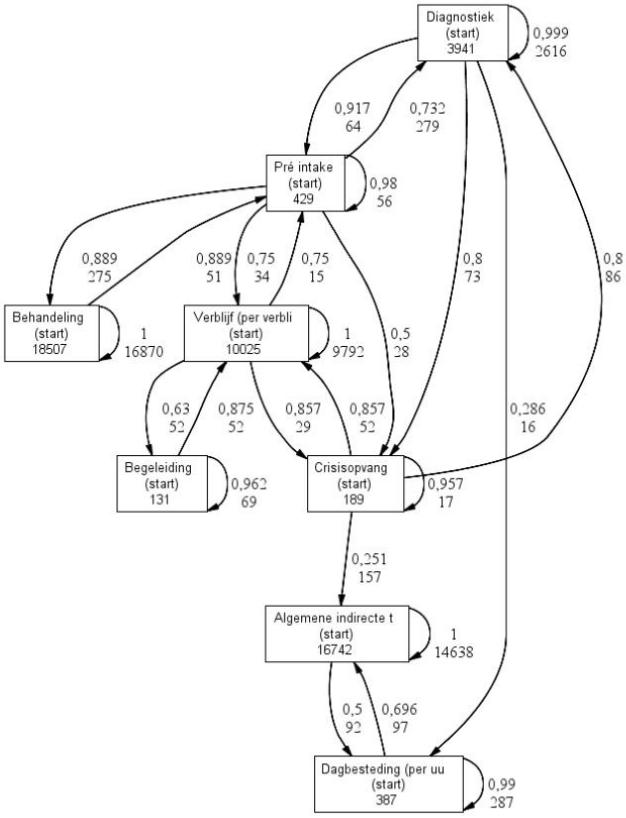


Figure G.11: Process model of the GGzE level 1 activities by Zanden [2010].

Recreating the process mining results by Gupta [2007]

Information on the dataset: Gupta used data originating from the Catharina hospital and four hospitals in the northern region in Italy. We have received all the data as ready-for-use .MXML files. The Catharina event logs contained:

- A log with complications with 576 cases, 183 event classes (although Gupta mentions 185) and 1.707 ATEs.
- A log with treatments with 2.711 cases, 255 event classes (although Gupta mentions 253) and 38.430 ATEs.

The Italian hospitals data contained:

- A log with both treatments and measurements, with 374 cases (one more than is mentioned by Gupta), 102 event classes and 11.790 ATEs. However, for the measurements we have one event that is named differently compared to the data used by Gupta; not *NIH* but *life_parameters*.
- A log with treatments with 369 cases (11 less than is mentioned by Gupta), 95 event classes (compared to the 35 that is mentioned by Gupta) and 6.484 ATEs.

Link to SHARE demo: http://is.ieis.tue.nl/staff/pvgorp/share/?page=ConfigureNewSession&vdi=XP-TUe10G-WithOfficeAndAcrobat_ProM_3.vdi

Links to screencasts:
Part1: http://www.screencast.com/t/5ZHwkZpgEmT
Part2: http://www.screencast.com/t/3Rkms1vP9
Part3: http://www.screencast.com/t/Psgv52S4D8wp
Part4: http://www.screencast.com/t/TusFxJfNQPX3
Part5: http://www.screencast.com/t/ufjOUvM4

Reproducing the results in SHARE:

1. First, the *HeuristicsMiner* with default settings was applied on the (Catharina) complications set with the addition of artificial start and end events (*Add artificial start/end task* pattern). For both the original researcher as the reproduced results in this thesis, the process models were very large and complex. Gupta achieved an ImprovedContinuousSymantic of -0,44, compared to our measure of 0,72.
2. We first applied the *Add artificial start/end task* pattern to the treatments log and used the *HeuristicsMiner* with default settings to obtain a very large and complex process model with an ImprovedContinuousSymantic of 0,63. Gupta achieved a similar complex process model with a fitness of -0,64.

3. Gupta tested the HM on both Catharina event logs. However, for most tests no parameter settings were mentioned. However, by applying the HM without the *all-activities-connected-heuristic* and artificial tasks, we achieved a similar unconnected process model compared to Gupta.

4. Gupta used the *Enhanced event log filter* to filter for events that occur less than 6,03% in the event log. In our case, this resulted in an event log with only two event classes left, as opposed to the process model with seven event classes for Gupta. Therefore, we were unable to reproduce these and some of the results that followed.

5. The next step was to mine on a subset of the complications log called *Uro-Genitaal*. We used the *Attribute value filter* (which worked properly this time, see section 4.4.3) in ProM to filter for events with the ComplicatieCategorie Uro-Genitaal. After we used the *Add artificial start/end task* pattern we applied the HM with default settings. This resulted in a process model that was much smaller compared to the result by Gupta. This difference could be the result of the differences in cases (99 in our event log compared to Gupta's 6) and ATEs (319 compared to 18). Changing the *dependency threshold* to 0,5 and the *L1L threshold* to 0,5 changed the model for Gupta, but this was not the case for our model.

6. Using the *DWS miner* with the same dataset and similar settings, we were unable to reproduce the original results, as we did not obtain the two clusters that are claimed by Gupta.

7. With the *Association Rule Miner* we were unable to reproduce the results by Gupta on the Uro-Genitaal subset. Gupta found 7 rules with the ARM but when following the same methodology, we did not obtain any rule at all (only when we added artificial start and end tasks we achieved four rules, each containing at least one artificial task).

8. Gupta tested the ARM on a larger event log which was not in our possession. However, as it appeared, the treatments event log was very similar (i.e. similar event classes). Unfortunately, using the *Apriori* algorithm with a population size of 100 (as did Gupta), we were unable to reproduce the original results as we found more rules (9 rules) compared to Gupta (5 rules).

9. Gupta used an unspecified event log for the clustering of PIs based on the association rules. Therefore, we used the treatments log to illustrate this effect. Using the ARM (default settings) resulted in several rules. However, the resulting clusters were very large (close to 2000 PIs for most rules) and therefore the resulting process models (obtained with the HM with default settings) were still very large and complex and could not be considered an improvement compared to the process model obtained by mining on the non-clustered event log. In contrast, Gupta was able to produce much more sensible process models

10. Using the *Enhanced event log filte*r we filtered for the seven measurement events in the Italian treatments and measurements event log. After addition of artificial start and end tasks and mining with the HM (default settings) we obtained a small and understandable model which was

comparable to the result by Gupta. However, there is a difference between the models due to different event names (i.e. life_parameters instead of NIH).

11. When we applied the ARM with the Apriori algorithm and a population size of 100 we only obtained 5 rules, compared to Gupta's 8. In addition, only 2 rules are similar between both results.

12. With the *PredictiveApriori* specified to 10 rules, we obtained only 7 rules compared to Gupta's 10. Again, only 2 rules were in both results.

13. Applying the HM (default settings) to the treatments log resulted in a complex and large model (comparable complexity to Gupta).

14. Using the ARM with the Apriori algorithm, a *population size* of 50 and a *confidence* of 0,5 we obtained only 11 rules compared to Gupta's 19.

15. With the default Apriori algorithm we obtained 6 rules, two more than Gupta. However, no similar rules were found in both results. In addition, when we clustered for any of the rules and applied the HM with default settings to the cluster, we were unable to create small and understandable process models that are comparable to the results by Gupta.

Appendix H. Validation of the best practices at Atrium and GGzE

Validation of the best practice results at Atrium

The datasets were collected as two separate .csv files, one for the *mammacare* and one for the *diabetic foot* (although not used in the validation) process. Both sets contained information on actual treatment events, appointment events and many different data attributes such as DBC-codes. Note that only the methodology and results are reported of the process models that have been verified by the Atrium.

Creating event logs:

1. In conclusion with the best practices, we loaded the *mammacare* dataset in MS Excel (*Data transformation (Excel)*).
2. Since the appointment data is not used by Riemers, we used the *Clustering (Excel)* pattern to remove the appointment data. As a result, 44.173 ATEs were left.
3. It was noted that the timestamp contained two different formats. By using the sort function, we identified a smaller group of events with an alternate date. Using the following formula we changed the date to the common format:

   =DATE(2000+RIGHT(A1;2);IF(MID(A1;4;3)="MAR";3;IF(MID(A1;4;3)="MAY";5;IF(MID(A1;4;3)="OCT";10)));LEFT(A1;2))

   By pasting this new column over the old timestamps, all dates were in the same format. We changed the whole column to short date and sorted for timestamp, patient ID and task ID.

4. It was noted that there were several originators that were named "\N". However, the department code was available and after contact with the Atrium to identify the departments, we renamed each originator according to list below.

   7201 = Radiologie
   7401 = Medische microbiologie
   8501 = Fysiotherapie
   8701 = Unknown
   9401 = Longziekten (longarts)
   9701 = MDL-arts (maag-darm-leverziekten).

5. There were many data attributes that were not significant during the recreation of the results. We decided to delete these for reasons of convenience (using simple select and delete options). We retained only the timestamp, patientID, event name, department, DBC diagnosis code, DBC treatment code in columns A to F respectively for each ATE.
6. We performed renaming of events for all events executed by the originators *Klinische chemie* and *Medische microbiologie* using the *Renaming of events (Excel)* pattern.
7. We aggregated all events *Klinische chemie* and *Medische microbiologie* per patient per day, according to the pattern *Aggregation of events (Excel)*. This left a dataset with 37.556 ATEs.
8. Using the *Clustering (Excel)* pattern, we clustered for DBC-codes 101 (*conservatief poliklinisch*) and 104 (*enkelvoudig poliklinisch conservatief*).

9. We created event logs for both clusters using the *Creating event log (Nitro)* pattern:
    a. The *conservatief poliklinisch* event log had 2020 cases (Riemers: 2000) and 74 event classes (Riemers: 95). This difference between numbers of event classes might be due to different renaming methods.
    b. The *enkelvoudig poliklinisch conservatief* event log had 1509 cases (Riemers 1500) and 34 event classes (Riemers 39).

Mining on *conservatief poliklinisch* 101:

10. **Result (1)** For the first mining run we only used the patterns *Add artificial start/end task* and *HeuristicsMiner*. This resulted in a large and complex process model (figure H.1), but it has a fitness of 0,785 (EBP). The process model by Riemers (figure H.2) had a very different fitness measure (unspecified) of <0,05.
11. **Result (2)** Using the event log obtained under 10, when we applied filtering for events that occur in less than 1% of the process instances using the *Enhanced event log filter* pattern, we retained 15 event classes (compared to Riemers' 17). After mining with default HM settings, we achieve a smaller and much more understandable model (compared to result 1) with a fitness of 0,795 (figure H.3), compared to Riemers' 0,05 (not specified which measure) for the model in figure H.4.
12. **Result (3)** Trace clustering (SOM, 4x4) on the event log obtained under 11 resulted in six clusters (compared to Riemers' sixteen clusters). Applying the HM resulted in a process model (figure H.5) with a BP fitness of 0,800. Unfortunately, Riemers has not specified any process models and/or fitness measures.

Mining on *enkelvoudig poliklinisch conservatief* 104:

13. **Result (4)** Using the *Add artificial start/end task* and *HeuristicsMiner* we obtained a large process model (figure H.6) with a fitness of 0,706.
14. **Result (5)** For the event log after 13, filtering for event classes that occur in less than 1% of the PIs, a much smaller process model (figure H.7) with seven event classes is created (EBP fitness of 0,599).

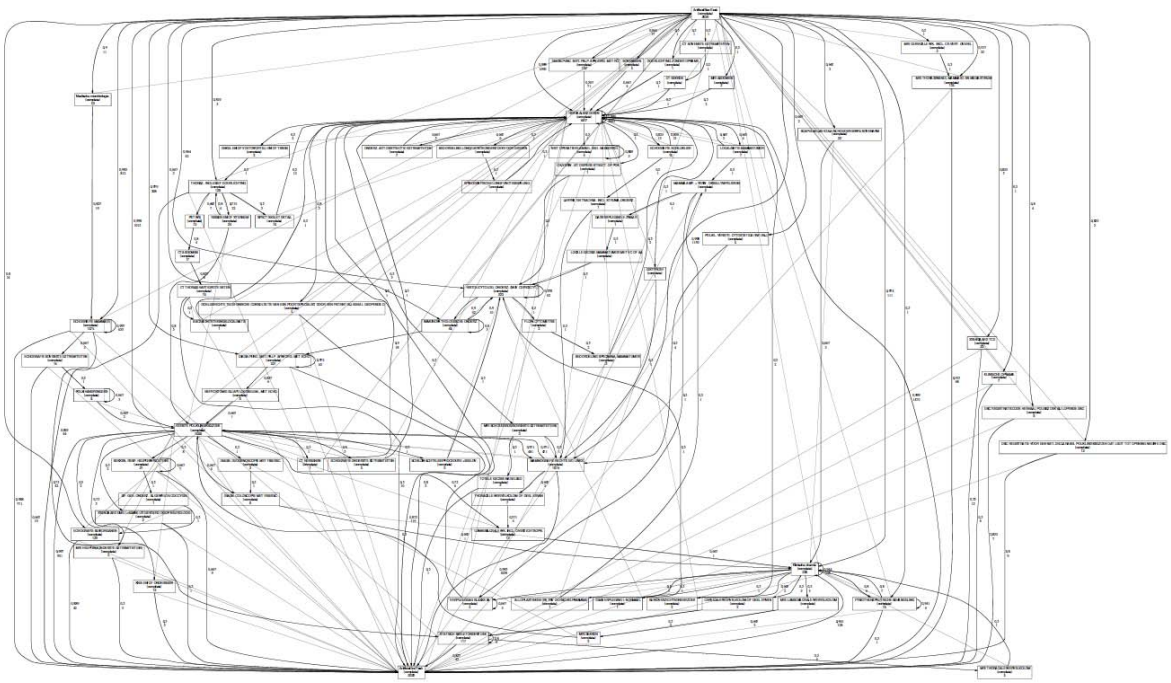


Figure H.1: Process model of Atrium conservatief poliklinisch (unfiltered).

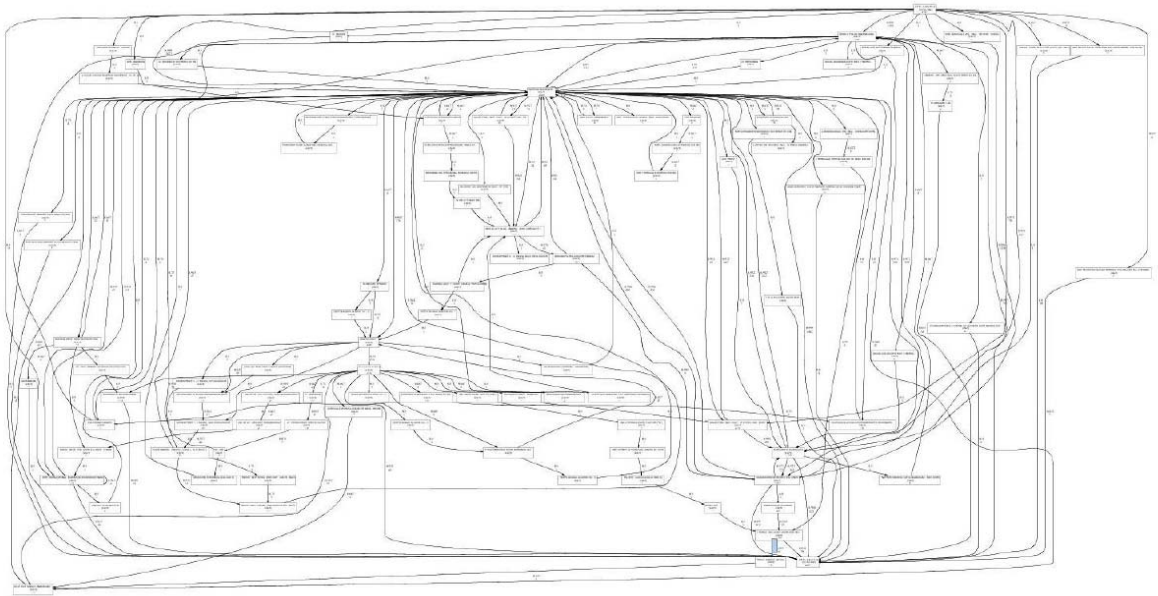


Figure H.2: Process model of Atrium conservatief poliklinisch (unfiltered) by Riemers [2009].

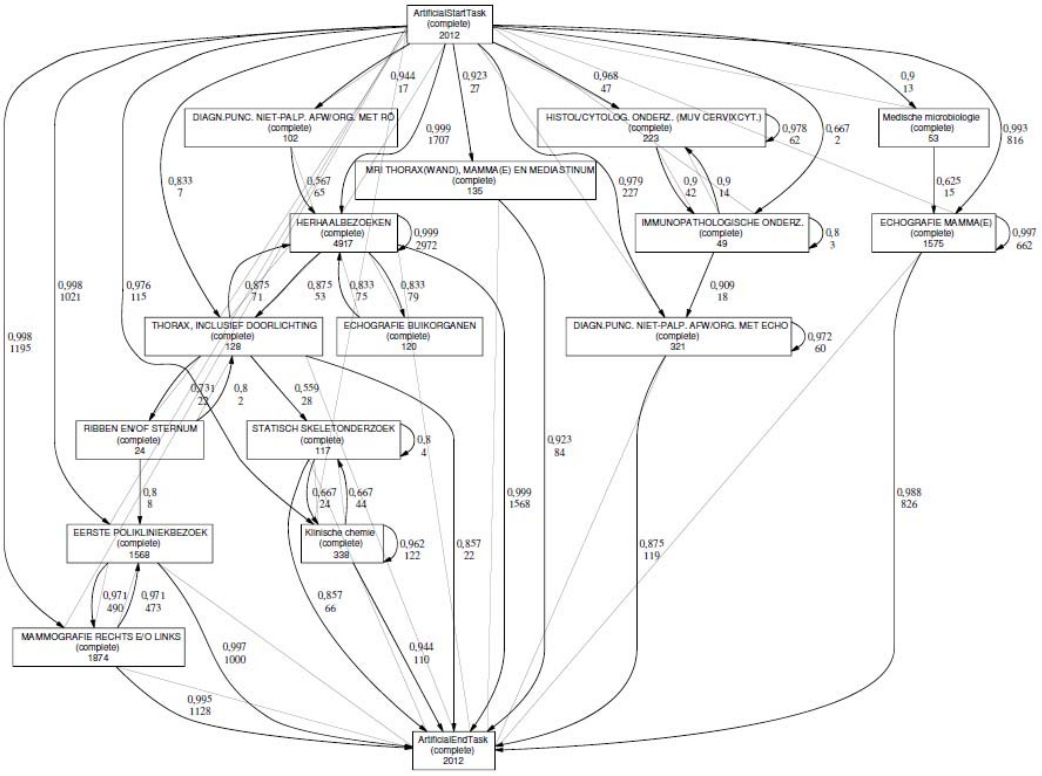


Figure H.3: Process model of Atrium conservatief poliklinisch (filtered).

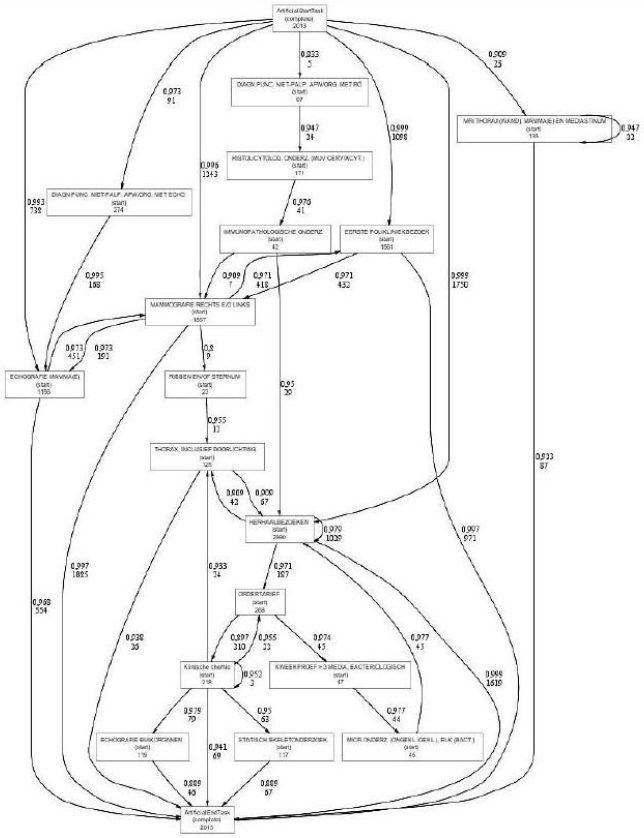


Figure H.4: Process model of Atrium conservatief poliklinisch (filtered) by Riemers [2009]

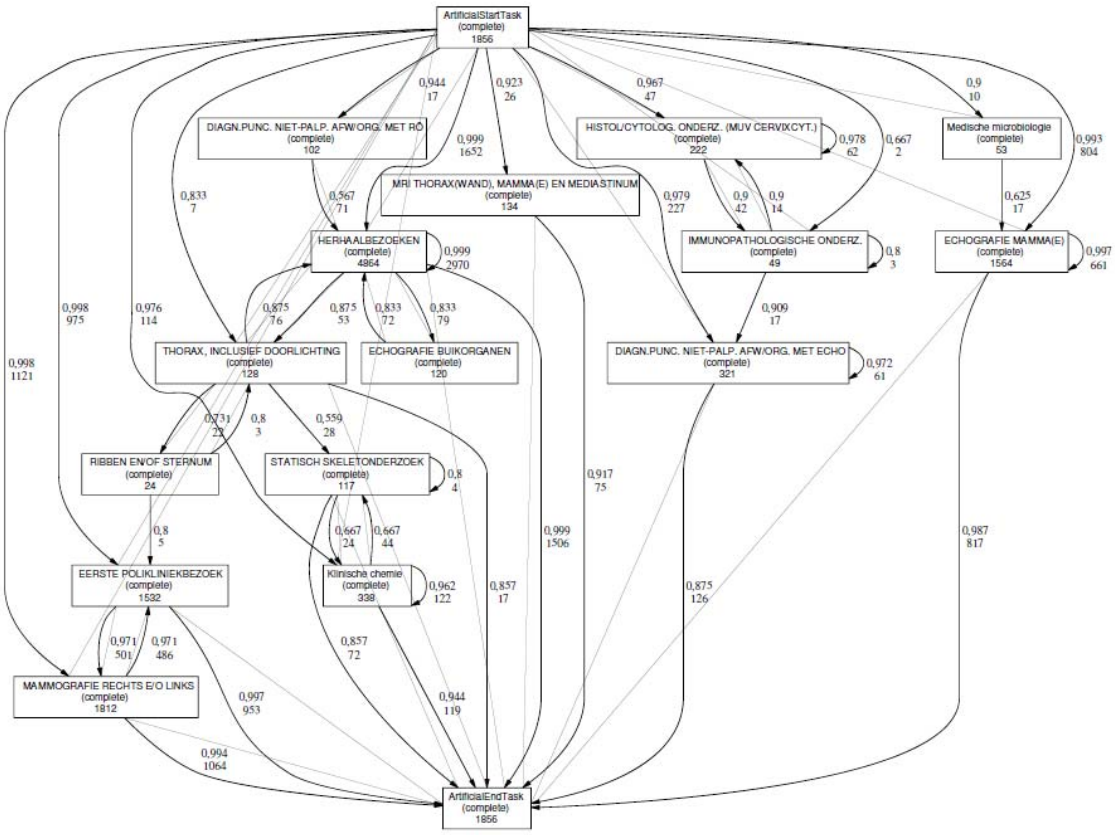


Figure H.5: Process model of the largest cluster in Atrium conservatief poliklinisch.

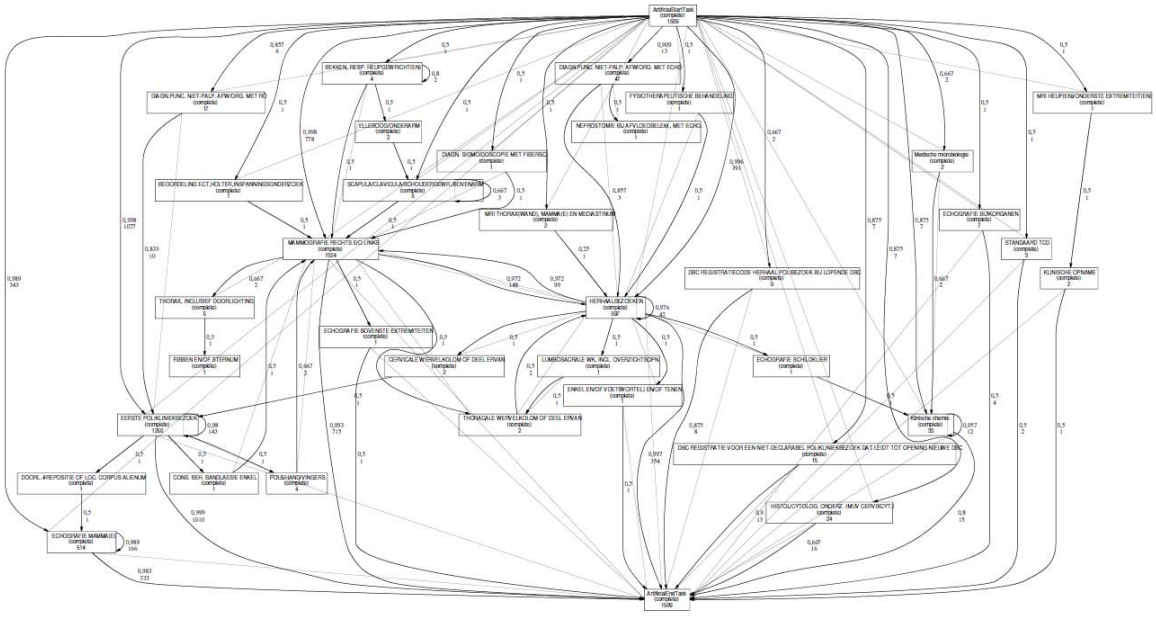


Figure H.6: Process model of Atrium enkelvoudig poliklinisch conservatief (unfiltered).

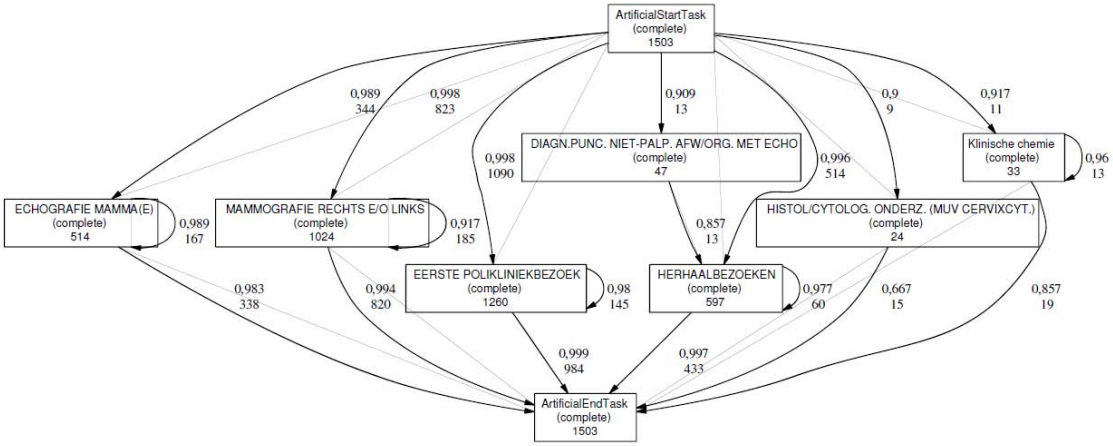


Figure H.7: Process model of Atrium enkelvoudig poliklinisch conservatief (filtered).

Validation of the best practice results at GGzE

For the validation of the best practices at GGzE we have used the same dataset that was used during the recreation of the results by Zanden [2010] in SHARE (Appendix G).

1. The original dataset was exported from MS Access to a .csv file for the *Data transformation (Excel)* pattern.

2. When the data was loaded in Excel, we first removed all data attributes that were not relevant for this study and as a result only retained the timestamp, reference number, the events on all three levels of detail, profession name level 3, care type code, care trajectory final date, diagnose description, diagnose element and diagnosis.

3. Using the *Clustering (Excel)* pattern we filtered for all the patients that had no closed care trajectory and thereby retained 50.351 ATEs.

4. There were some ATEs for which no diagnosis was specified and these were removed with the *Clustering (Excel)* pattern, leaving 49.909 ATEs.

5. Considering there were only 61 event classes we decided not to use the *Renaming of events (Excel)* pattern either. In addition, since there were only 525 double ATEs (on 49.909) we decided not to use the *Aggregation of events (Excel)* pattern to achieve the most accurate process model.

6. With the *Clustering (Excel)* pattern we created separate datasets for the diagnoses *Pervasieve ontwikkelingsstoornissen* (16.112 ATEs) and *Aandachtstekortstoornissen en gedragsstoornissen* (9.721 ATEs).

7. Using the *Clustering (Excel)* pattern we created a separate dataset for the diagnose element *Pervasieve ontwikkelingsstoornis NAO* (13.599 ATEs) from the *Pervasieve ontwikkelingsstoornissen* dataset.

8. From the dataset *Aandachtstekortstoornissen en gedragsstoornissen* we created a separate dataset for the diagnose element *Gecombineerde type* (3.914 ATEs).

9. Using the Creating event log (Nitro) pattern we created event logs for the datasets *Pervasieve ontwikkelingsstoornissen* (889 cases and 50 event classes) (from now on called **result 1**), *Pervasieve ontwikkelingsstoornis NAO* (738 cases and 48 event classes) (from now on called **result 2**) and *Gecombineerde type* (241 cases, 28 event classes) (from now on called **result 3**).

10. We used the *Enhanced event log filter* pattern to filter for event classes that did not occur in less than 1% of the PIs. This left 23, 23 and 18 event classes for results 1 to 3 respectively.

11. The *Add artificial start/end task* pattern was used to add artificial tasks to all three datasets.

12. We applied the *HeuristicsMiner* pattern with default settings to all three datasets which created process models with EBP fitness measures of 0,778, 0,709 and 0,533 for results 1 to 3 respectively. The resulting models are illustrated in figures H.8 to H.10 for results 1 to 3 respectively.

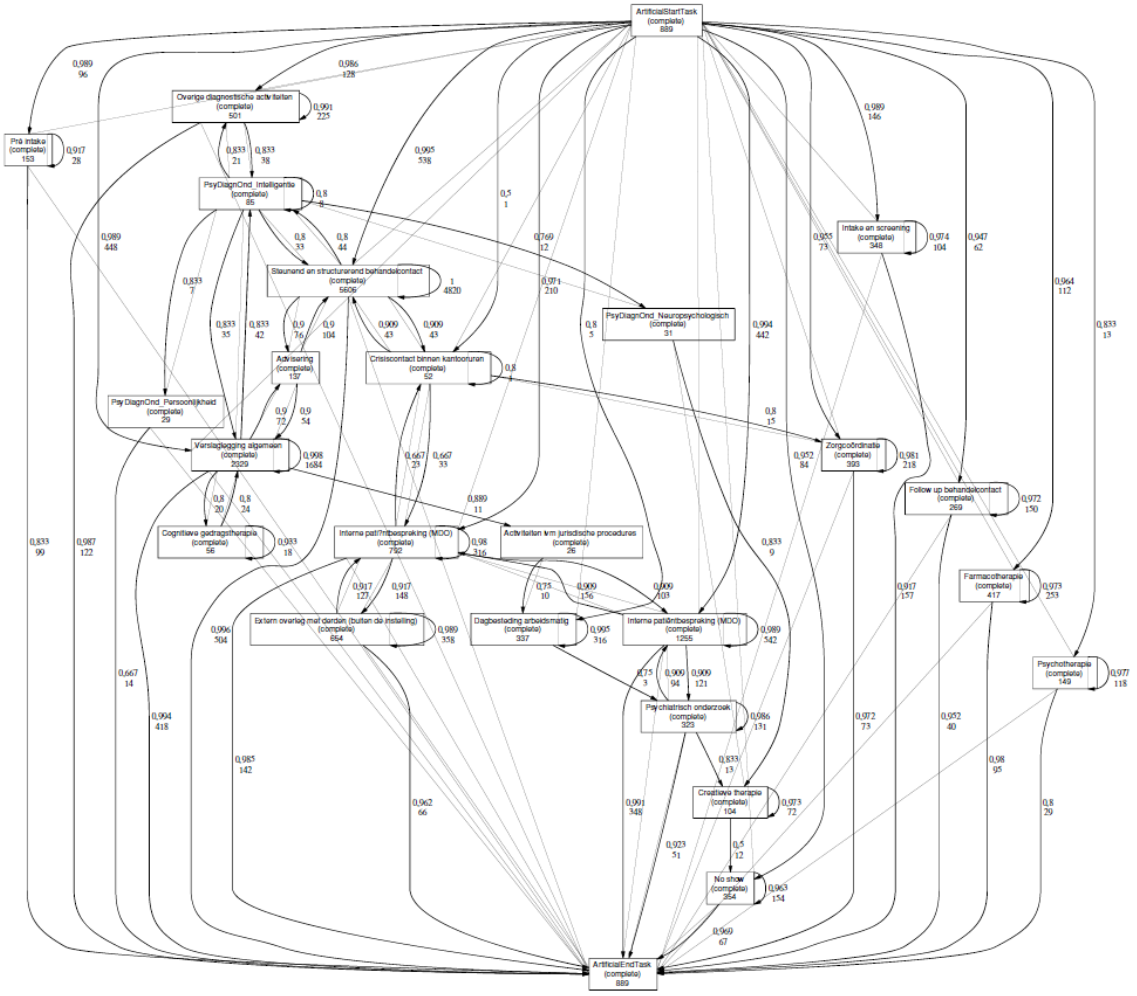


Figure H.8: Process model for the GGzE Pervasieve ontwikkelingsstoornissen dataset.

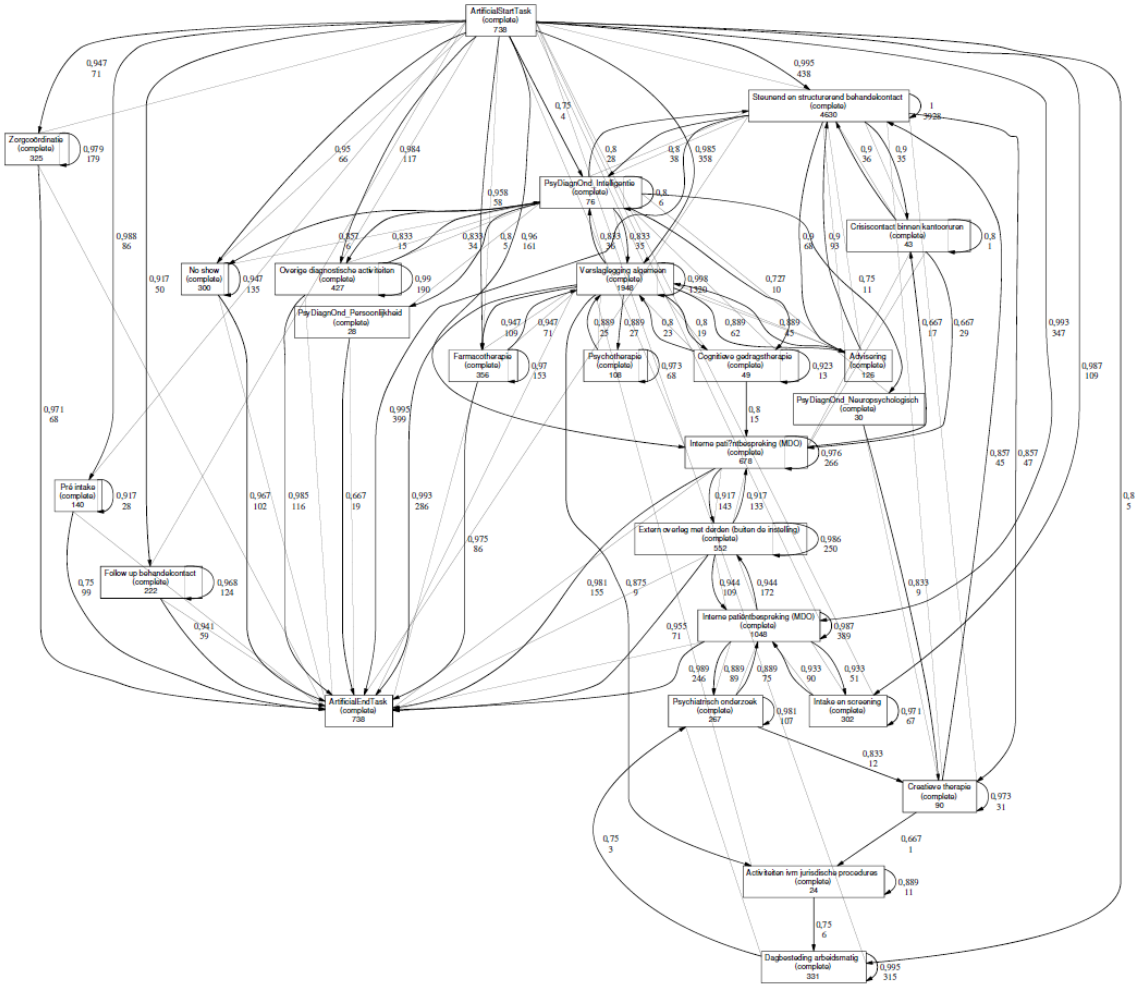


Figure H.9: Process model for the GGzE Pervasieve ontwikkelingsstoornis NAO dataset.

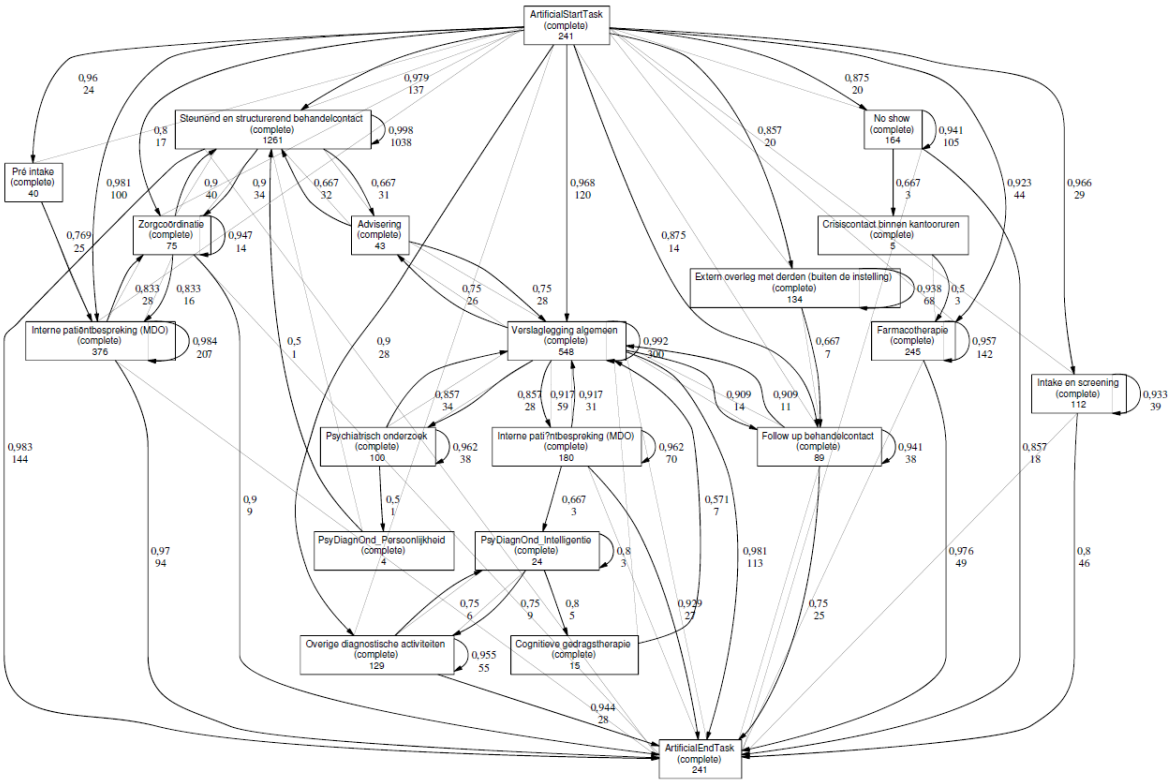


Figure H.10: Process model for the GGzE Gecombineerde type dataset.

Appendix I. Process mining patterns

List of process mining patterns

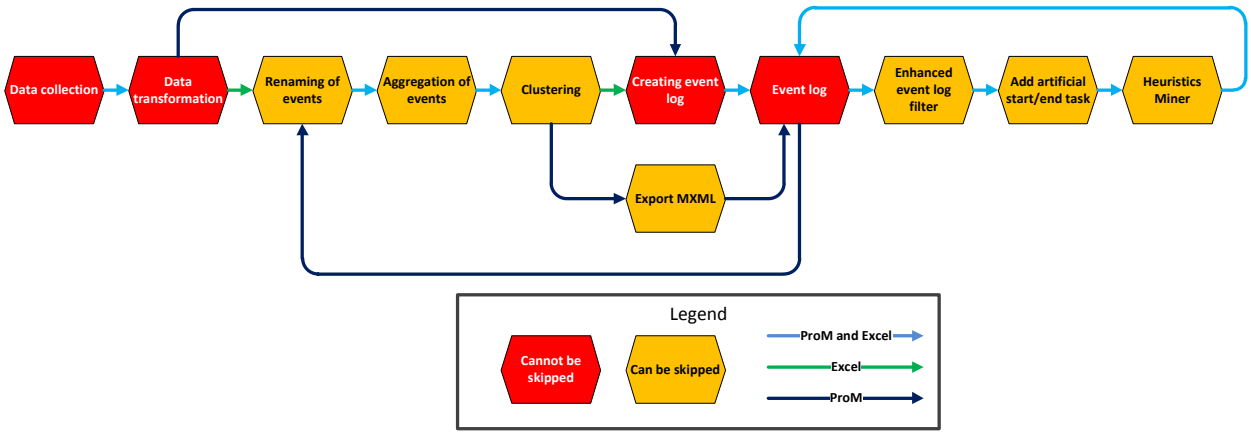


Figure I.1: Process mining pattern network.

Pattern Name: Add artificial start/end task

Intent: The goal of this pattern is to add an artificial start and end task to the process model, making it easier for the user to identify the start and end in the process model, which can be particularly problematic in larger process models.

Also known As: -

Author/date: R.J.P. Janssen, 8 August 2011

Consider These Patterns First: Enhanced event log filter

Alternatives: -

Implementation: ProM 5.2

Motivation: By adding an artificial start and end task to the process model, the user can more easily recognize the start and end of the process. This is especially convenient in larger and more complex processes. Moreover, they are easily added to and removed from the event log, and do not change the master file.

Involved data objects: -

Solution: The artificial start and end task are easily added to the event log by using the filter menu at the home dashboard. Choose the "*advanced*" options and click both the "*Add Artificial Start Event Log Filter*" and "*Add Artificial End Event Log Filter*". They can just as easily be removed again.

Screencast: http://screencast.com/t/tPtjj9Jv8u

Trade-offs:

Pros:

1. Easy to add and easy to remove
2. Makes the process much more easy to understand

Cons: -

Known Uses: Mans *et al.* [2008, 2009], Gupta [2007], Riemers [2009], Ramos [2009], Zanden [2010]

Consider Next: Heuristics Miner

Pattern Name: Aggregation of events

Intent: The goal of this pattern to aggregate common events that are executed multiple times (in a row) for most cases in the event log, thereby reducing the total number of events in the event log and resulting in less complex process models.

Also known As: -

Author/date: R.J.P. Janssen, 8 August 2011

Consider These Patterns First: Renaming of events

Alternatives: -

Implementation: MS Excel, ProM 5.2

Motivation: In datasets, it is possible many "double" events are present for a case on the same day. An example is the large amount of laboratory tests in hospital data. These are of no particular interest for the process and greatly increase the dataset's size and complexity. By aggregating these events, we reduce the total number of events (ATEs) in the data, reducing process model complexity.

Involved data objects: Timestamp, ID, events

Solution: The goal of this pattern is to aggregate similar events that are performed for a case at the same day, e.g. all activities X for patient Y on day Z. This is not required for all events but can be interesting for activities which are of minor interest for the main behaviour in a process, such as laboratory tests. In the event log in table I.1, there are many events AAA (let us say they are laboratory tests). These events could be considered for aggregation. When aggregating for all events AAA for a patient on a single day, in the new situation entire row 6 is deleted.

	A	B	C	D
1	Timestamp	ID	Activity	NEW
2	1-1-2000	1	BBB	BBB
3	1-1-2000	1	AAA	AAA
4	1-1-2000	1	CCC	CCC
5	1-1-2000	1	AAA	AAA
6	1-1-2000	1	AAA	deleted
7	2-1-2000	1	AAA	AAA
8	2-1-2000	2	AAA	AAA

Table I.1: Example event log

The result of the pattern is a smaller dataset with a smaller total number of events (ATEs), thereby reducing the size and complexity of the dataset.

There are multiple patterns for the aggregation of events, namely: *Aggregation of events (Excel)* and *Aggregation of events (ProM).*

Screencast: -

Trade-offs: -

Known Uses: Mans *et al.* [2009], Riemers [2009], Ramos [2009]

Consider Next: Clustering

Pattern Name: Aggregation of events (Excel)

Intent: The goal of this pattern to aggregate common events that are executed multiple times (in a row) for most cases in the event log, thereby reducing the total number of events in the event log and resulting in less complex process models.

Also known As: -

Author/date: R.J.P. Janssen, 8 August 2011

Consider These Patterns First: Renaming of events (Excel)

Alternatives: Aggregation of activities (ProM)

Implementation: MS Excel

Motivation: In datasets, it is possible many "double" events are present for a case on the same day. An example is the large amount of laboratory tests in hospital data. These are of no particular interest for the process and greatly increase the dataset's size and complexity. By aggregating these events, we reduce the total number of events (ATEs) in the data, reducing process model complexity.

Involved data objects: Timestamp, ID, events

Solution: The goal of this pattern is to aggregate similar events that are performed for a case at the same day, e.g. all activities X for patient Y on day Z. This is not required for all events but can be interesting for activities which are of minor interest for the main behaviour in a process, such as laboratory tests. In the event log in table I.2, there are many events AAA (let us say they are laboratory tests). These events could be considered for aggregation. When aggregating for all events AAA for a patient on a single day, in the new situation entire row 6 is deleted.

	A	B	C	D
1	Timestamp	ID	Activity	NEW
2	1-1-2000	1	BBB	BBB
3	1-1-2000	1	AAA	AAA
4	1-1-2000	1	CCC	CCC
5	1-1-2000	1	AAA	AAA
6	1-1-2000	1	AAA	deleted
7	2-1-2000	1	AAA	AAA
8	2-1-2000	2	AAA	AAA

Table I.2: Example event log

The first step is to identify all events that are candidate for aggregation. The second step is to remove these events from the data. Consider the following dataset (column A to C) for an example of this pattern:

In this case, we are interested in deleting double events AAA (column C). Row 6 contains a "double" event and can therefore be selected for deletion. To select events for aggregation in MS Excel we use the following formula:

=IF(AND((C2="AAA");(C1=C2);(A2=A1);(B2=B1));"delete";C2)

We paste this formula in a new column D (the above formula is for cell D2). It checks whether the event in cell C2 is indeed AAA, the preceding event C1 is AAA, both events are executed on the same day, and whether the case ID is similar for both ATEs. When these conditions are met, the value "delete" is returned in the cell D2 NEW. If any of the conditions is not met, the event in cell C2 is returned in the cell D2. In the example we see that row 6 receives the value "delete" and the other rows do not.

With slight modifications to the formula it is possible to select multiple events for aggregation, or aggregate based on different conditions.

To delete the rows that are marked “delete”, the following macro can be used:

```
Sub Loop_Example()
    Dim Firstrow As Long
    Dim Lastrow As Long
    Dim Lrow As Long
    Dim CalcMode As Long
    Dim ViewMode As Long

    With Application
        CalcMode = .Calculation
        .Calculation = xlCalculationManual
        .ScreenUpdating = False
    End With

    With ActiveSheet

        .Select

        ViewMode = ActiveWindow.View
        ActiveWindow.View = xlNormalView

        .DisplayPageBreaks = False

        Firstrow = .UsedRange.Cells(1).Row
        Lastrow = .UsedRange.Rows(.UsedRange.Rows.Count).Row

        For Lrow = Lastrow To Firstrow Step -1

            With .Cells(Lrow, "D")

                If Not IsError(.Value) Then

                    If .Value = “delete” Then .EntireRow.Delete
                    ‘ “” for strings only, numeral elements do not require it

                End If

            End With

        Next Lrow

    End With

    ActiveWindow.View = ViewMode
    With Application
        .ScreenUpdating = True
        .Calculation = CalcMode
    End With

End Sub
```

The previous macro can be used to remove rows that meet a specific criterion. In this case, column D is checked for values of *“delete”*. For every cell X in D that contains *“delete”*, the entire row that contains cell X will be deleted from the data.

This pattern results in a smaller dataset with a smaller total number of events, thereby reducing the size and complexity of the dataset.

Screencast: http://www.screencast.com/t/wgWsvzvt5

Trade-offs:

Pros:

1. It is possible to aggregate based on many different conditions and is not confined to a single method of aggregation, as is the alternative pattern
2. This pattern allows much more freedom compared to the alternative pattern.

Cons:

1. Some MS Excel code knowledge is required and therefore it is slightly more complex than the alternative.

Known Uses: The technique in this pattern is not explicitly mentioned but similar approaches have been used by Riemers [2009] and Ramos [2009].

Consider Next: Clustering (Excel)

Pattern Name: Aggregation of events (ProM)

Intent: The goal of this pattern to aggregate common events that are executed multiple times (in a row) for most cases in the event log, thereby reducing the total number of events in the event log and resulting in less complex process models.

Also known As: -

Author/date: R.J.P. Janssen, 8 August 2011

Consider These Patterns First: Renaming of events (ProM)

Alternatives: Aggregation of events (Excel)

Implementation: ProM 5.2

Motivation: In datasets, it is possible many "double" events are present for a case on the same day. An example is the large amount of laboratory tests in hospital data. These are of no particular interest for the process and greatly increase the dataset's size and complexity. By aggregating these events, we reduce the total number of events (ATEs) in the data, reducing process model complexity.

Involved data objects: Events

Solution: Aggregation of activities in ProM is very simple and straightforward. The first step is to load an event log in ProM. In the dashboard that subsequently pops-up, select the option "*Filter*" and within that menu select "*Advanced*". Search the list of filters for the "*Repetitions-to-Activity Filter*" and select it by either double clicking or clicking "*add selected filter*". No further actions are needed. The result is "*an event log where all direct repetitions of the same audit trail entry are replaced by one 'start' event with the time stamp from the first occurrence and a 'complete' event with the time stamp of the last occurrence in this sequence of repetitions*" [ProM tool]. Note that this counts for all event classes and can also aggregate activities that occur on different days (see Trade-offs)!

Screencast: http://screencast.com/t/b1U9S97enkh6

Trade-offs:
Pros:

1. Aggregation is simple and straightforward.
2. No code knowledge is required.

Cons:

1. It is not possible to specify which events need to be aggregated and the pattern counts for all events.
2. It is possible that direct repetitions, although on a different day, can also automatically be aggregated. This depends on the random ordering of activities on the same day by ProM. If activity A takes place on day 1 and 2, and the order in the event log is AA, then the two days are aggregated. If there is an activity B in between, so that we have ABA, the activities A are not aggregated.
3. There is considerable less freedom compared to the alternative pattern (see 1 and 2).

Known Uses: Mans *et al*. [2008, 2009]

Consider Next: Clustering (ProM)

Pattern Name: Clustering

Intent: The goal of this pattern is to cluster process instances with a similar data attributes, in order to create more homogenous datasets, thereby reducing dataset size and complexity.

Also known As: -

Author/date: R.J.P. Janssen, 8 August 2011

Consider These Patterns First: Aggregation of events

Alternatives: -

Implementation: MS Excel, ProM 5.2

Motivation: In large heterogeneous datasets, it is highly likely cases share similar attributes. For instance, hospital patients are either male or female. These attributes can be used to cluster cases to create a more homogenous dataset. The theory behind this widely used pattern is that a process model is more likely to be less complex when the underlying data is more homogenous.

Involved data objects: Any data attribute(s) can be used as a selection criterion for clustering of cases.

Solution: Basically there are two options for clustering, manual and automatic. In this former case, the researcher is the determining factor for clustering. For instance, the researcher decides to cluster based on patients gender. In the latter case, automated algorithms determine the clustering process (however the researcher can specify which data attributes to pay attention to).

Clustering (Excel) deals with clustering based on manual actions only. *Clustering (ProM)* features a wide range of both manual and automatic clustering patterns.

Screencast: -

Trade-offs: -

Known Uses: Gupta, [2007], Mans *et al*. [2008, 2009], Riemers [2009], Ramos [2009], Zanden [2010]

Consider Next: Creating event log (Nitro), Export MXML

Pattern Name: Clustering (Excel)

Intent: The goal of this pattern is to cluster process instances with a similar data attributes, in order to create more homogenous datasets, thereby reducing dataset size and complexity.

Also known As: -

Author/date: R.J.P. Janssen, 8 August 2011

Consider These Patterns First: Aggregation of events (Excel)

Alternatives: Clustering (ProM)

Implementation: MS Excel

Motivation: In large heterogeneous datasets, it is highly likely cases share similar attributes. For instance, hospital patients are either male or female. These attributes can be used to cluster cases to create a more homogenous dataset. The theory behind this widely used pattern is that a process model is more likely to be less complex when the underlying data is more homogenous.

Involved data objects: Any data attribute(s) can be used as a selection criterion for clustering of cases.

Solution: For this pattern we presume that the researcher is interested to cluster the cases in the dataset based on one specific data attribute (could be any attribute of any format). The researcher has to determine only two things, what attribute to use for clustering and the "value" that the attribute must contain. All process instances that contain the specific value for the attribute will be retained in the dataset, all other process instances will be deleted.

In MS Excel, suppose we want to cluster for data attribute X, which is located in column A. Attribute X contains values 1 and 2. We are interested in process instances that contain the value 1 in column A. The following macro can be used to delete all cases that contain the value 2 and thereby retaining only the cases with 1.

```
Sub Loop_Example()
    Dim Firstrow As Long
    Dim Lastrow As Long
    Dim Lrow As Long
    Dim CalcMode As Long
    Dim ViewMode As Long

    With Application
        CalcMode = .Calculation
        .Calculation = xlCalculationManual
        .ScreenUpdating = False
    End With

    With ActiveSheet

        .Select

        ViewMode = ActiveWindow.View
        ActiveWindow.View = xlNormalView

        .DisplayPageBreaks = False

        Firstrow = .UsedRange.Cells(1).Row
        Lastrow = .UsedRange.Rows(.UsedRange.Rows.Count).Row

        For Lrow = Lastrow To Firstrow Step -1
```

```
            With .Cells(Lrow, "A")
                If Not IsError(.Value) Then
                    If .Value = 2 Then .EntireRow.Delete
                End If
            End With
        Next Lrow
    End With

    ActiveWindow.View = ViewMode
    With Application
        .ScreenUpdating = True
        .Calculation = CalcMode
    End With

End Sub
```

In the previous macro, all cases that contain the value 2 in column A are deleted and thereby only the cases with value 1 in column A are retained. The first yellow coloured line specifies the column that is checked. The second yellow line holds the equation for which the column is checked. This equation can easily be changed with the standard MS Excel equations, e.g. to *if .Value = 2 Or .Value = 3 Then .EntireRow.Delete* to delete both values 2 and 3. By using *if .Value <> 2 Then .EntireRow.Delete*, we delete all rows that do not contain 2.

Screencast: http://www.screencast.com/t/cvW5LLEdaih

Trade-offs:

Pros:

1. This pattern is fast and accurate.
2. The user has much freedom and can use many different attributes for clustering.

Cons:

1. This pattern requires some knowledge on codes in MS Excel.
2. The researcher needs knowledge on the data attributes that are interesting to use as source for clustering.

Known Uses: This pattern is not explicitly mentioned in any report but similar techniques have been used by Riemers [2009] and Ramos [2009].

Consider Next: Creating event log (Nitro)

Pattern Name: Clustering (ProM)

Intent: The goal of this pattern is to cluster process instances with a similar data attributes, in order to create more homogenous datasets, thereby reducing dataset size and complexity.

Also known As: -

Author/date: R.J.P. Janssen, 8 August 2011

Consider These Patterns First: Aggregation of events (ProM)

Alternatives: Clustering (Excel)

Implementation: ProM 5.2

Motivation: In large heterogeneous datasets, it is highly likely cases share similar attributes. For instance, hospital patients are either male or female. These attributes can be used to cluster cases to create a more homogenous dataset. The theory behind this widely used pattern is that a process model is more likely to be less complex when the underlying data is more homogenous.

Involved data objects: Any data attribute(s) can be used as a selection criterion for clustering of cases.

Solution: In ProM, it is possible to use both manual and/or automatic clustering.

Manual clustering patterns: -

Automatic clustering patterns:

1. *Trace clustering*

Screencast: -

Trade-offs: -

Known Uses: Mans *et al.* [2009], Riemers [2009], Ramos [2009]

Consider Next: Export MXML

Pattern Name: Creating event log

Intent: The goal of this pattern is to transform the data in the data transformation software into an event log in the .MXML file format for analysis in ProM.

Also known As: Load (from ETL)

Author/date: R.J.P. Janssen, 8 August 2011

Consider These Patterns First: Clustering (Excel), Data transformation (Access)

Alternatives: -

Implementation: Nitro, ProM*import*

Motivation: The ProM software requires specific file formats as input and one cannot go directly from the database manager/transformation software (e.g. MS Access or MS Excel) to analysis in ProM. The database has to be transformed into a .MXML or .XES (ProM 6) format before the data can be analyzed in ProM.

Involved data objects: .MXML

Solution: There are two methods to create an event log from a database. *Creating event log* (*Nitro)* uses a .csv file from MS Excel and *Creating event log* (*ProMimport)* requires a MS Access .mdb file input.

Screencast: -

Trade-offs: -

Known Uses: Mans *et al*. [2008, 2009], Riemers [2009], Zanden [2010]

Consider Next: Event log

Pattern Name: Creating event log (Nitro)

Intent: The goal of this pattern is to transform the data in the data transformation software into an event log in the .MXML file format for analysis in ProM.

Also known As: Load (from ETL)

Author/date: R.J.P. Janssen, 8 August 2011

Consider These Patterns First: Clustering (Excel)

Alternatives: Creating event log (ProM*import*)

Implementation: Nitro

Motivation: The ProM software requires specific file formats as input and one cannot go directly from the database manager (e.g. MS Access or MS Excel) to analysis in ProM. The database has to be transformed into a .MXML or .XES format by Nitro before the data can be loaded into ProM.

Involved data objects: Nitro requires a .csv file format as input. The minimal requirements for the data are at least a timestamp, an activity and case ID. Furthermore, additional data objects such as originators and data attributes can be useful during analysis in ProM.

Solution: The transformation of the database to an event log with Nitro is straightforward. The first step is to load the database into the Nitro software. The software will show the user part of the database (all its columns and only a fraction of the rows). The next step is to specify the column that contains the timestamp, activity, case ID and originator (optional). Additionally, data attributes such as case characteristics can be specified.

Screencast: http://screencast.com/t/GFogUyplleQ

Trade-offs:

Pros:

1. This pattern is much easier to use compared to the alternative pattern, as no elaborate and complex database construction is necessary.
2. Simple .csv files can be used as input.
3. Nitro provides some analytics on the event log (such as frequencies) straightaway.

Cons:

1. Nitro is commercial software and therefore it is not for free use (although it is possible to obtain a free license as a student or scientific researcher).

Known Uses: -

Consider Next: Event log

Pattern Name: Creating event log (ProMimport)

Intent: The goal of this pattern is to transform the data in the data transformation software into an event log in the .MXML file format for analysis in ProM.

Also known As: Load (from ETL)

Author/date: R.J.P. Janssen, 8 August 2011

Consider These Patterns First: Data transformation (Access)

Alternatives: Creating event log (Nitro)

Implementation: MS Access and ProM*import*

Motivation: The ProM software requires specific file formats as input and one cannot go directly from the database manager/transformation software (e.g. MS Access or MS Excel) to analysis in ProM. The database has to be transformed into a .MXML or .XES (ProM 6) format before the data can be analyzed in ProM.

Involved data objects: ProM*import* can feature a wide range of inputs, among them the MS Access .mdb file format.

Solution: Creating an event log with ProM*import* and MS Access is complex and time consuming and cannot easily be explained. Therefore we refer to the user manual by Mans [20??].

Screencast: -

Trade-offs:

Pros:

1. ProM*import* is free to use.
2. ProM*import* can handle wide a range of data formats as input

Cons:

1. Creating an event log with this pattern is considerably more complex and time consuming compared to alternative pattern

Known Uses: Mans *et al*.[2008, 2009], Riemers [2009], Zanden [2010]

Consider Next: Event log

Pattern Name: Data collection

Intent: The goal is to gather data from information systems so it can be transformed into usable formats for the data analysis software. The key for any analysis is a dataset and therefore this is an important pattern which serves as the starting point for any research.

Also known As: Extract (from ETL)

Author/date: R.J.P. Janssen, 8 August 2011

Consider These Patterns First: -

Alternatives: Data is an obvious and absolute prerequisite for data analysis and therefore the data collection is an important pattern. The only alternative to this pattern is to use a pre-existing dataset (which in essence requires a certain level of collecting as well).

Implementation: Various ERP/CRM/HI systems can be used as a data source. However, past research has proved billing systems to be useful sources of data for process mining in healthcare, as they contain all executed events with the required additional data (see involved data objects).

Motivation: When one is interested in data analysis, an early and obvious step in the process is the collection of useful data that can be analyzed. This is true for process mining as well. Data can be collected from a wide range of information systems and depending on the researcher's interests, can involve different data characteristics.

Involved data objects: Depending on the researcher's intentions, different data objects may be of interest for the analysis. However, the following data objects are a requirement for process mining analysis: *timestamp*, *case identifier* and *activity/event*. Additionally, data such as *originators,* DBC-code information or attributes such as case characteristics (e.g. date of birth, gender etc.) can be collected for future convenience during filtering and clustering patterns. Note that it is always possible to discard data later on in the analysis and therefore it is better to collect too much information as opposed to too little. Depending on the source of the data, data can be collected in various formats such as .txt, .csv or .mdb, which can easily be loaded in software tools such as MS Access or MS Excel, where the data can be transformed.

Solution: This pattern is highly dependent on the data source that provides the input. Therefore, no system specific detailed information on the extraction of data from the source can be provided. However, it is important to consider the type of information the researcher is after before he commits to extraction. The type of information is dependent on the desired results of the analysis. As a result, certain data sources are more fitting for the intended goals than others. Subsequently, depending on the type of source, certain software is required in future patterns. In any case, the result of this pattern should at least be a dataset that can be loaded preferably in data manipulation software tool, preferably MS Excel or MS Access. Additionally, the dataset should contain the required data objects.

Screencast: -

Trade-offs: -

Known Uses: Mans *et al*. [2008, 2009], Riemers [2009], Zanden [2010], Gupta [2007]

Consider Next: Data transformation

Pattern Name: Data transformation

Intent: The goal of this pattern is to transform a raw dataset that was extracted from a data source, into a file format that can be transformed into the .MXML file format for analysis in ProM. Furthermore, it is essential to discard insignificant information to reduce data size and avoid distraction by too many details.

Also known As: Transform (from ETL), pre-processing

Author/date: R.J.P. Janssen, 8 August 2011

Consider These Patterns First: Data collection

Alternatives: -

Implementation: MS Excel, MS Access

Motivation: Raw data from that is extracted from a data source usually contains undesired or incomplete data. Furthermore, it is possible that the data is not in the right format for transformation into .MXML files that are required for analysis in ProM. Therefore, the raw data needs to be transformed in the correct software tools before it can be loaded into ProM.

Involved data objects: Depending on the researcher's intentions, different data objects may be of interest for the analysis. However, the following data objects are a requirement for process mining analysis: *timestamp*, *case identifier* and *activity/event*. Additionally, data such as *originators,* DBC-code information or attributes such as case characteristics (e.g. date of birth, gender etc.) can be retained for future convenience during filtering and clustering patterns.

Solution: This pattern is useful to downsize a dataset by getting rid of unwanted. Besides size reduction (which contributes to processing speed), it also reduces the complexity of the dataset and allows the researcher to focus on the most significant information. Depending on the researcher's intentions, several choices with regard to the transformation of the dataset can be made.

The first step is loading the raw dataset into a software tool such as MS Access (*Data transformation (Access)*) or MS Excel (*Data transformation (Excel)*) (database manager or spreadsheet application). It is possible (and likely) that the raw data contains many columns with many different data attributes (which represent the data objects such as events, timestamps and case IDs) and rows (which represent individual ATEs). Therefore, for the convenience of the researcher, it is recommended to rearrange the columns such that the most significant data objects (i.e. timestamp, activity, case ID, originator and DBC-codes) are placed as the first few columns.

The second step is to discard all information that is not desired (i.e. insignificant for the analysis). For process mining, additional data attributes such as patient characteristics and insurance information, is usually of much lesser interest and can therefore be discarded. Furthermore, it is up to the researcher to decide whether to include only activities of the process and discard events such as the administrative tasks. This step can easily be performed by selecting and deleting the undesired columns (data objects) and rows (ATEs) (make use of the sort ability).

Additionally, steps such as the implementation of artificial timestamps can be executed when necessary. Furthermore, it is possible the dataset contains a certain hierarchy for events. As a result, it is recommended to create several subsets for each level, making future process mining more convenient.

The result of this step should be a pre-processed set of data that contains only the information of interest to the researcher. The following list presents the useful parts of the data that should be considered part of the dataset for hospital information:

1. Timestamp
2. Case identifier
3. Activity
4. Originator (can be both a department and/or person, as well as requesting as acting originator)
5. DBC code information
6. Note: It is recommended to retain process related information only and to discard administrative tasks.

Subsequently, the researcher can perform additional pre-processing steps (see “Consider next”). Take note that it always possible to return to this pattern as the researcher’s interest for the data changes during the course of the project. It is therefore recommended to always keep a master file of the raw data.

Screencast: -

Trade-offs: -

Known Uses: Mans *et* al. [2008, 2009], Riemers [2009], Ramos [2009], Zanden [2010]

Consider Next: Renaming of events (Excel), Creating event log (ProM*import*)

Pattern Name: Data transformation (Access)

Intent: The goal of this pattern is to transform a raw dataset that was extracted from a data source, into a file format that can be transformed into the .MXML file format for analysis in ProM. Furthermore, it is essential to discard insignificant information to reduce data size and avoid distraction by too many details.

Also known As: Transform (from ETL), pre-processing

Author/date: R.J.P. Janssen, 8 August 2011

Consider These Patterns First: Data collection

Alternatives: Data transformation (Excel)

Implementation: MS Access

Motivation: Raw data from that is extracted from a data source usually contains undesired or incomplete data. Furthermore, it is possible that the data is not in the right format for transformation into .MXML files that are required for analysis in ProM. Therefore, the raw data needs to be transformed in the correct software tools before it can be loaded into ProM.

Involved data objects: Depending on the researcher's intentions, different data objects may be of interest for the analysis. However, the following data objects are a requirement for process mining analysis: *timestamp*, *case identifier* and *activity/event*. Additionally, data such as *originators,* DBC-code information or attributes such as case characteristics (e.g. date of birth, gender etc.) can be retained for future convenience during filtering and clustering patterns.

Solution: This pattern is useful to downsize a dataset by getting rid of unwanted. Besides size reduction (which contributes to processing speed), it also reduces the complexity of the dataset and allows the researcher to focus on the most significant information. Depending on the researcher's intentions, several choices with regard to the transformation of the dataset can be made.

The first step is loading the raw dataset into MS Access. It is possible (and likely) that the raw data contains many columns with many different data attributes (which represent the data objects such as events, timestamps and case IDs) and rows (which represent individual ATEs). Therefore, for the convenience of the researcher, it is recommended to rearrange the columns such that the most significant data objects (i.e. timestamp, activity, case ID, originator and DBC-codes) are placed as the first few columns.

The second step is to discard all information that is not desired (i.e. insignificant for the analysis). For process mining, additional data attributes such as patient characteristics and insurance information, is usually of much lesser interest and can therefore be discarded. Furthermore, it is up to the researcher to decide whether to include only activities of the process and discard events such as the administrative tasks. This step can easily be performed by selecting and deleting the undesired columns (data objects) and rows (ATEs) (make use of the sort ability).

Additionally, steps such as the implementation of artificial timestamps can be executed when necessary. Furthermore, it is possible the dataset contains a certain hierarchy for events. As a result, it is recommended to create several subsets for each level, making future process mining more convenient.

The result of this step should be a pre-processed set of data that contains only the information of interest to the researcher. The following list presents the useful parts of the data that should be considered part of the dataset for hospital information:

1. Timestamp
2. Case identifier
3. Activity
4. Originator (can be both a department and/or person, as well as requesting as acting originator)
5. DBC code information
6. Note: It is recommended to retain process related information only and to discard administrative tasks

Subsequently, the researcher can perform additional pre-processing steps (see "Consider next"). Take note that it always possible to return to this pattern as the researcher's interest for the data changes during the course of the project. It is therefore recommended to always keep a master file of the raw data.

The result of this pattern is a dataset in the .mdb file format.

Screencast: -

Trade-offs:

Pros:

1. Depending future patterns may be simpler to use this pattern than the alternative pattern.

Cons:

1. It requires a series of complex steps to go from a MS Access file to a .MXML file.
2. Some MS Access knowledge is required.

Known Uses: Mans *et* al. [2008, 2009], Riemers [2009], Ramos [2009], Ramos [2009], Zanden [2010]

Consider Next: Creating event log (ProM*import*)

Pattern Name: Data transformation (Excel)

Intent: The goal of this pattern is to transform a raw dataset that was extracted from a data source, into a file format that can be transformed into the .MXML file format for analysis in ProM. Furthermore, it is essential to discard insignificant information to reduce data size and avoid distraction by too many details.

Also known As: Transform (from ETL), pre-processing

Author/date: R.J.P. Janssen, 8 August 2011

Consider These Patterns First: Data collection

Alternatives: Data transformation (Access)

Implementation: MS Excel

Motivation: Raw data from that is extracted from a data source usually contains undesired or incomplete data. Furthermore, it is possible that the data is not in the right format for transformation into .MXML files that are required for analysis in ProM. Therefore, the raw data needs to be transformed in the correct software tools before it can be loaded into ProM.

Involved data objects: Depending on the researcher's intentions, different data objects may be of interest for the analysis. However, the following data objects are a requirement for process mining analysis: *timestamp*, *case identifier* and *activity/event*. Additionally, data such as *originators,* DBC-code information or attributes such as case characteristics (e.g. date of birth, gender etc.) can be retained for future convenience during filtering and clustering patterns.

Solution: This pattern is useful to downsize a dataset by getting rid of unwanted. Besides size reduction (which contributes to processing speed), it also reduces the complexity of the dataset and allows the researcher to focus on the most significant information. Depending on the researcher's intentions, several choices with regard to the transformation of the dataset can be made.

The first step is loading the raw dataset into MS Excel. It is possible (and likely) that the raw data contains many columns with many different data attributes (which represent the data objects such as events, timestamps and case IDs) and rows (which represent individual ATEs). Therefore, for the convenience of the researcher, it is recommended to rearrange the columns such that the most significant data objects (i.e. timestamp, activity, case ID, originator and DBC-codes) are placed as the first few columns.

The second step is to discard all information that is not desired (i.e. insignificant for the analysis). For process mining, additional data attributes such as patient characteristics and insurance information, is usually of much lesser interest and can therefore be discarded. Furthermore, it is up to the researcher to decide whether to include only activities of the process and discard events such as the administrative tasks. This step can easily be performed by selecting and deleting the undesired columns (data objects) and rows (ATEs) (make use of the sort ability).

Additionally, steps such as the implementation of artificial timestamps can be executed when necessary. Furthermore, it is possible the dataset contains a certain hierarchy for events. As a result, it is recommended to create several subsets for each level, making future process mining more convenient.

The result of this step should be a pre-processed set of data that contains only the information of interest to the researcher. The following list presents the useful parts of the data that should be considered part of the dataset for hospital information:

1. Timestamp
2. Case identifier
3. Activity
4. Originator (can be both a department and/or person, as well as requesting as acting originator)
5. DBC code information
6. Note: It is recommended to retain process related information only and to discard administrative tasks

Subsequently, the researcher can perform additional pre-processing steps (see "Consider next"). Take note that it always possible to return to this pattern as the researcher's interest for the data changes during the course of the project. It is therefore recommended to always keep a master file of the raw data.

For future convenience (i.e. creating the event log in Nitro), it is recommended to save the dataset as a .csv (comma separated value) format.

Screencast: http://www.screencast.com/t/0AYjseRx

Trade-offs:

Pros:

1. Data transformation in MS Excel is more convenient than in MS Access, due to the simplicity of the transformation from a MS Excel file to a .MXML file.
2. It is relatively easy to manipulate data in MS Excel.

Cons:

1. Some MS Excel code knowledge is required.

Known Uses: Riemers [2009], Ramos [2009], Zanden [2010]

Consider Next: Renaming of events (Excel)

Pattern Name: Enhanced event log filter

Intent: The goal of this pattern is to filter the event log for event classes with a low global frequency and/or tasks that occur in a small amount of process instances, in order to reduce the number of event classes and thereby complexity of the event log.

Also known As: Filtering

Author/date: R.J.P. Janssen, 8 August 2011

Consider These Patterns First: Event log

Alternatives: -

Implementation: ProM 5.2

Motivation: It is possible the event log contains event classes that have a low frequency, and/or occur only in a relatively small amount of process instances. It may therefore be interesting to filter for these event classes, in order to focus more on the general behaviour in the process instead of specific, non-frequent behaviour. The Enhanced event log filter is a simple method to filter for these low frequent event classes.

Involved data objects: Events

Solution: The Enhanced event log filter can be found under the advanced filters in the home dashboard in ProM. After clicking the filter, the researcher can specify the conditions of the filter.

First, the top left filter can be used to filter for tasks for which the frequency (in percentages) in the event log is less than the percentage that is specified. The specified percentage is dependent on the interest of the researcher, but a widely used measure is 1%. Additionally, the right filter can be used to filter for tasks for which the frequency (in percentages) of in how many different process instances they occur, thereby also focussing on more global behaviour. In most cases, the global frequency and PI frequency show some correlation. Furthermore, it is possible to select either one or both filters with the top right option AND/OR.

The researcher can also decide to focus on the less frequent behaviour, by inverting the selection of tasks. Additionally, tasks can also manually be selected or deselected. This last option could be useful if process experts indicate certain task that would have normally be filtered, are indeed important, or common tasks are not that interesting at all.

Screencast: http://screencast.com/t/KeIPuhBhY

Trade-offs:
Pros:
1. Easy to add and easy to remove.
2. Makes the process much easier to understand.

Cons: -

Known Uses: Riemers [2009], Ramos [2009], Gupta [2007]

Consider Next: Add artificial start/end task

Pattern Name: Event log

Intent: The event log is the basis for analysis of data in ProM.

Also known As: -

Author/date: R.J.P. Janssen, 8 August 2011

Consider These Patterns First: Creating event log, Export MXML

Alternatives: -

Implementation: ProM a specific file format as input and one cannot go directly from the database manager (e.g. MS Access or MS Excel) to analysis in ProM. The database has to be transformed into a .MXML or .XES format by Nitro before the data can be loaded into ProM.

Involved data objects: The event log has the .MXML file format (for ProM 5.2 and lower) or the .XES file format for ProM 6.

Solution: The event log (figure I.2) is a prerequisite for analysis of data in ProM. From this event log, a process model can be extracted, process performance analysis can be performed and other process details can be discovered. Moreover, it is possible to modify the event log to zoom in on certain parts of the process or highlight points of interest using the many different algorithms in ProM.

For use in ProM, the event log should at least contain a *timestamp*, *event name* and a *case identifier*. Additional information such as *originators* and *DBC-information* (in healthcare) is convenient.

case id	activity id	originator	time stamp
case 1	activity A	John	9-3-2004:15.01
case 2	activity A	John	9-3-2004:15.12
case 3	activity A	Sue	9-3-2004:16.03
case 3	activity B	Carol	9-3-2004:16.07
case 1	activity B	Mike	9-3-2004:18.25
case 1	activity C	John	10-3-2004:9.23
case 2	activity C	Mike	10-3-2004:10.34
case 4	activity A	Sue	10-3-2004:10.35
case 2	activity B	John	10-3-2004:12.34
case 2	activity D	Pete	10-3-2004:12.50
case 5	activity A	Sue	10-3-2004:13.05
case 4	activity C	Carol	11-3-2004:10.12
case 1	activity D	Pete	11-3-2004:10.14
case 3	activity C	Sue	11-3-2004:10.44
case 3	activity D	Pete	11-3-2004:11.03
case 4	activity B	Sue	11-3-2004:11.18
case 5	activity E	Clare	11-3-2004:12.22
case 5	activity D	Clare	11-3-2004:14.34
case 4	activity D	Pete	11-3-2004:15.56

Figure I.2: Example event log [Weijters et al., 2006].

Screencast: -

Trade-offs: -

Known Uses: Mans *et al*. [2008, 2009], Riemers [2009], Ramos [2009], Gupta [2007], Zanden [2010]

Consider Next: Enhanced event log filter, Renaming of events (ProM)

Pattern Name: Export MXML

Intent: The goal of this pattern is to export an event log which has been filtered, clustered or modified in any possible way in ProM, so that it can be used in the future without having to redo all the steps.

Also known As: -

Author/date: R.J.P. Janssen, 8 August 2011

Consider These Patterns First: Clustering (ProM)

Alternatives: -

Implementation: ProM 5.2

Motivation: It is possible to modify the event log in ProM, by for instance filtering events/instances, clustering data or renaming events. These steps can aid in the extraction of a more simple and better process model. To allow the researcher to reuse the modified event log in the future, it is possible to save the modified event log, and thereby also retaining the original file in which the modifications have been applied.

Involved data objects: .MXML file

Solution: A modified event log is easily exported in ProM as a new event log. Select the filtered event log under the tab "*Exports*" and export as "*Efficient MXML.GZ Export*". The result is a new event log with all the modifications.

Screencast: http://screencast.com/t/ej2KN7NSl

Trade-offs: -

Pros:

1. Exporting a modified event log allows the researcher to separate the modifications from the master event log.

Cons: -

Known Uses: No report explicitly mentions the export function.

Consider Next: Event log

Pattern Name: HeuristicsMiner

Intent: The goal of this pattern is to create a process model from an event log, represented by the so called HeuristicsNet, which shows the different events and the order in which they are executed, as well as their frequencies and relations.

Also known As: -

Author/date: R.J.P. Janssen, 8 August 2011

Consider These Patterns First: Add artificial start/end task

Alternatives: -

Implementation: ProM 5.2

Motivation: The HM is one of the most used algorithms in ProM and can be found in much process mining research. The most important reason to use the HM is the fact that it can deal with noise in the event log, which is an inevitable complication when real life data is used. Moreover, the algorithm can be used to mine the main behaviour in an event log and can deal with loops, hidden activities and long distance relationships. Furthermore, the representation of the process model by the so called "HeuristicsNets" is easy to understand in comparison with for instance a Petri net, which features different representations for places and tasks in the model and require more knowledge of the modelling language for interpretation.

Involved data objects: The process model is created using the following data objects: timestamp, case ID, events.

Solution: The Heuristics Miner is an excellent method to extract a process model from an event log. However, it features a set of eight parameters, and three additional options. Using the default settings on a raw event log usually results in a complex "spaghetti-like" process model. Therefore, such an approach should only be used on a pre-processed event log (e.g. renaming, aggregation, filtering). Using default settings on a carefully pre-processed dataset can result in understandable process models, and should not necessarily require modification of the parameter settings. Unfortunately, there is no optimal general setting, but researchers are encouraged to test different settings for their event log to obtain even better process models. As the parameters are not easily explained, <LINK HERE> will provide more information on the parameters.

In addition to the previous "default" mining approach, the "Ton Weijters" approach can be used to extract a process model as well. This method should also result in process models that exceed that quality of process models extracted from raw event logs:

1. Remove all filters.
2. *Add* artificial *start/end tasks*.
3. Set the *Positive observations* to 1.
4. Set the *Dependency*, *Length-one-loops*, *Length-two-loops* and *Long distance thresholds* to 0,95.
5. Switch the *all-activities-connected-heuristic* **off** and press *start mining.*

6. identify all non-connected and low frequent events.
7. Use the *Enhanced event log filter* to filter for the activities under 6.
8. Mine on the filtered log (obtained after 7) with the *all-activities-connected-heuristic* (and *Positive observations* set to 1) switched **on**.

Screencast: http://screencast.com/t/RdgEEqEpS

Trade-offs:

Pros:

1. The heuristics miner produces an easy to understand process model.
2. The "Ton Weijters" approach leads to better process models.

Cons:

1. The "Ton Weijters" approach is not suitable for event logs with many events, as this can result in much manual labour.
2. Changing the parameter settings slightly can drastically change the process model, and therefore it can be difficult to determine the optimal settings for each event log.

Known Uses: Mans *et al.* [2008, 2009], Gupta [2007], Riemers [2009], Ramos [2009], Zanden [2010].

Consider Next: Event log

Pattern Name: Renaming of events

Intent: The goal of this pattern is to reduce the total number of event classes and thereby reducing process model size and complexity, by the technique of renaming events.

Also Known As: -

Author/date: R.J.P. Janssen, 8 August 2011

Consider These Patterns First: Data transformation

Alternatives: -

Implementation: MS Excel, ProM

Motivation: In this context we consider events/activities, such as laboratory tasks, that are uninteresting to the main behaviour of the process, and due to their frequency in numbers and variants, contribute significantly to the complexity of the resulting process model. Therefore, by renaming these events, a smaller amount of event classes can be obtained (note that renaming only affects the number of event classes, not the total number of events/audit trail entries in the event log), and thereby reducing dataset complexity. The renamed events can then be selected as candidates for aggregation in future patterns.

Involved data objects: Events/activities

Solution: Event classes that are insignificant to the main behaviour of the process can be considered for renaming. Frequently, these events are executed by a certain originator, or have some other data attribute in common. These common attributes are key to renaming the event classes. For instance, all (for the main behaviour of the process insignificant) event classes A to F are executed by originator 1. Subsequently, all event classes executed by originator 1 are renamed to one new name G. As result, the number of event classes is reduced with five, contributing to the understandability of the process model that is extracted from the data. In the end, there are numerous options upon which renaming can be based, but it is recommended to use the originators as a foundation for renaming in hospital data.

For the renaming of events, different approaches can be considered, each with its own trade-offs. The following patterns can be considered for renaming of events: *Renaming of events (Excel)* and *Renaming of events (ProM)*.

Screencast: -

Trade-offs: -

Known Uses: Mans *et al*. [2008, 2009], Riemers [2009], Ramos [2009]

Consider Next: Aggregation of events

Pattern Name: Renaming of events (Excel)

Intent: The goal of this pattern is to reduce the total number of event classes and thereby reducing process model size and complexity, by the technique of renaming events.

Also Known As: -

Author/date: R.J.P. Janssen, 8 August 2011

Consider These Patterns First: Data transformation (Excel)

Alternatives: Renaming of events (ProM)

Implementation: MS Excel

Motivation: In this context we consider events/activities, such as laboratory tasks, that are uninteresting to the main behaviour of the process, and due to their frequency in numbers and variants, contribute significantly to the complexity of the resulting process model. Therefore, by renaming these events, a smaller amount of event classes can be obtained (note that renaming only affects the number of event classes, not the total number of events/audit trail entries in the event log), and thereby reducing dataset complexity. The renamed events can then be selected as candidates for aggregation in future patterns.

Involved data objects: Events/activities, originators

Solution: In this context we consider events that need to be renamed in order to decrease the number of event classes and create candidates for aggregation. Table I.3 will serve as an example throughout this pattern. In the original dataset we only have columns A and B.

First, it is necessary to create an additional column C where the renamed activities are placed (named "New activity"). Furthermore, this column will also display the activities that do not need renaming. In this specific context we make use of the department name (Laboratory) as a new name for the renamed activities, but in practice, any name can be used by the researcher. *The goal is to rename the activities in column A, that are executed by Laboratory in column B, to the name Laboratory in column C.*

The second step is to identify candidates for renaming. In this example, we select events (column A) based on the originator (e.g. department name). In practice it is possible to select activities based on any data attribute. For each event, we check whether it is executed by a specific originator (column B), if true, the event is selected for renaming, if false, we will copy the original event name. The generic formula for this step is *=if(target cell="criterion";"new name";target cell)*, and applied to the example example we use the following function *=if(target cell="department name X";"department name X";target cell)* in each cell in column C (below row 1), which in terms compares the target cell to a pre-specified department name X and returns if true (the second part), the pre-specified department name X, if false (the third part), the target cell (i.e. the original event name). Of course, any event can be selected for renaming and formulas can be stacked to rename multiple events in only one formula.

To illustrate, the formula has been modified to the example below. In cell C2 we paste =if(B2="Laboratory";"Laboratory";B2), in cell C3 we use B3 instead of B2 etc. Column C now returns the renamed and original event names, depending on whether the criterion is met. As we observe, there is no change to row 2. However, the events in row 3 and 4 have been renamed into the new name Laboratory.

	A	B	C
1	Activity	Department	New activity
2	X-Ray	Radiology	X-Ray
3	Glucose	Laboratory	Laboratory
4	Calcium	Laboratory	Laboratory

Table I.3: Example event log

The final step is to *cut* row C, select row A, paste special as *number*. The list in column A is now replaced by the list in column C. The result of this pattern is a dataset with renamed and original events.

Screencast: http://www.screencast.com/t/HYacXawtpU8f

Trade-offs:

Pros:

1. Many event classes can be renamed at the same time, requiring little manual effort.
2. It is possible to rename based on many different data attributes.
3. The pattern allows much more freedom than the alternative pattern, which can be of value for advanced MS Excel users.

Cons:

1. Some MS Excel code knowledge is required.

Known Uses: Riemers [2009]

Consider Next: Aggregation of events (Excel)

Pattern Name: Renaming of events (ProM)

Intent: The goal of this pattern is to reduce the total number of event classes and thereby reducing process model size and complexity, by the technique of renaming events.

Also known As: -

Author/date: R.J.P. Janssen, 8 August 2011

Consider These Patterns First: Event log

Alternatives: Renaming of events (Excel)

Implementation: ProM 5.2

Motivation: In this context we consider events/activities, such as laboratory tasks, that are uninteresting to the main behaviour of the process, and due to their frequency in numbers and variants, contribute significantly to the complexity of the resulting process model. Therefore, by renaming these events, a smaller amount of event classes can be obtained (note that renaming only affects the number of event classes, not the total number of events/audit trail entries in the event log), and thereby reducing dataset complexity. The renamed events can then be selected as candidates for aggregation in future patterns.

Involved data objects: Events/activities

Solution: Renaming events in ProM is very simple and straightforward. The first step is to load an event log in ProM. In the dashboard that subsequently pops-up, select the option "*Filter*" and within that menu select "*Advanced*". Search the list of filters for the "*Remap Element Log Filter*" and select it by either double clicking or clicking "*add selected filter*".

On the left hand side of the table, the user is required to specify the event class that needs to be renamed. On the right hand side the user can specify the new name for the event class.

The result is an event log where all the specified event classes are renamed to a newly specified name.

Screencast: http://screencast.com/t/4Xb4UtrM40UB

Trade-offs:

Pros:

1. No code knowledge required.
2. The steps are straightforward and simple.

Cons:

1. Renaming is much more labour intensive compared to the alternative pattern, as it is only possible to rename one event class at a time.
2. There is much less freedom compared to the alternative pattern.

Known Uses: Mans *et al*. [2008, 2009]

Consider Next: Aggregation of events (ProM)

Pattern Name: Trace clustering

Intent: The goal of this pattern is to cluster process instances based on a number of features, in order to create more homogenous subsets of process instances, thereby reducing dataset size and complexity.

Also known As: -

Author/date: R.J.P. Janssen, 8 August 2011

Consider These Patterns First: Aggregation of events (ProM)

Alternatives: Clustering (Excel)

Implementation: ProM 5.2

Motivation: In large heterogeneous datasets, it is highly likely that some process instances are more similar to each other than others. Using trace clustering, it is possible to extract similar process instances, thereby creating several more homogenous subsets of process instances. Subsequently, process models extracted from these subsets are more likely to be less complex and smaller in size, making it easier to understand the process model.

Involved data objects: Any data attribute(s) can be used as a selection criterion for clustering of cases.

Solution: Trace clustering is an automated clustering algorithm that features a large set of parameter settings, different distance metrics and clustering algorithms. However, in many cases, the Self Organizing Maps with Euclidian distance is used, usually combined with the remaining settings on default.

The result of this pattern is a number of subsets with process instances that are closer together than to the process instances in the other clusters. Subsequently, these subsets can be used for process mining to obtain smaller and less complex process models.

Screencast: http://screencast.com/t/PXDo1Up3joj

Trade-offs:

Pros:

1. The clustering is completely automatic.
2. Smaller clusters of process instances are obtained.

Cons:

1. There are many different parameter settings, clustering and distance measures to choose from.
2. Results tend to vary greatly.

Known Uses: Mans *et al.* [2009], Riemers [2009], Ramos [2009]

Consider Next: Export MXML