
RWTH AACHEN UNIVERSITY

This work was submitted to:

**Chair of Process and Data Science (PADS - Informatik 9), RWTH Aachen University**

---

# Thesis hhh

---

## Master's/Bachelor's Thesis

Author: **Author**

Student ID: **Student ID**

Supervisor: Dr. Supervisor

Examiners: Prof. One
Prof. Two

Registration Date: YYYY-MM-DD

Submission Date: YYYY-MM-DD

# Abstract

The abstract should be a concise summary of what you have done. It should cover, in roughly 1 to 2 sentences per topic:

- The domain, i.e., explain the application domain in which your thesis is applicable.
- The problem, i.e., motivate what the problem is you are solving.
- The method, i.e., describe the method you have proposed, and, compare it, briefly & high level to other approaches. In particular, focus on the benefits of your approach versus the other approach(es)
- The evaluation, i.e., how did you evaluate your method, what results did you obtain and what do the results indicate?

# Contents

# Chapter 1

# Introduction

In this section, you introduce the work to your audience. Even though this is the first chapter of your thesis, you should not necessarily write this chapter first. It is often easier to write this chapter at the end of the thesis writing. In this part of your introduction, i.e., the part before the Motivation section (Section 1.1), you write a paragraph (not just a sentence) for each of the bullet points mentioned in the abstract:

- Domain, i.e., explain the application domain in which your thesis is applicable.
- Problem, i.e., motivate what the problem is you are solving.
- Method, i.e., describe the method you have proposed, and, compare it, briefly & high level to other approaches. In particular, focus on the benefits of your approach versus the other approach(es)
- Evaluation, i.e., how did you evaluate your method and what do the results indicate?

Optionally, you can finalize this part with a little overview of what is still to come in this chapter: "The remainder of this chapter is structured as follows. In Section 1.1, we motivate the need for . . . In Section 1.2, we provide a formal problem statement. . . . "

Before we dive into the motivation section, a few general guidelines for writing:

1. Write *active* and in the *present tense*.
2. Avoid "optionality" as much as possible, i.e., usage of the following words should (pun intended) be minimized:
    - could
    - would
    - should
    - may
    - might
    - can
    - will

- ...

Good: We present a method that ...
Bad: We will present a method that ...
Good: We use function $f$ and compute its inverse.
Bad: We can use the function $f$ and will compute its inverse.

3. Connect mathematical operators using the { }-operators in math mode ($ $).
   Good: $x \in X$
   Bad: $x \in X$
   Good: $f\colon X \to Y$
   Bad: $f\colon X \to Y$
   If using { } prevents your mathematics to break at the end of the line, use `\allowbreak` in math mode.
4. Do not use \\ to generate line breaks

   Do it like

   This
5. Position the caption of a Figure below the figure.
6. Position the caption of a Table above the table.
7. Write Section Headings and Titles Like This
   Good: Approximation Bias in Unstable Systems
   Bad: Approximation bias in unstable systems
8. Be consistent in your references.
   - Make sure that the name of the same author is always the same (using DBLP helps for this)
   - Titles should be consistent. Either use the style previously described, i.e., Approximation Bias in Unstable Systems or Approximation bias in unstable systems (this is allowed in the references, opposed to your own section headings and titles, however, *BE CONSISTENT*).

## 1.1 Motivation

In this section, you explain to the reader why:

1. The problem you are solving is relevant to be solved.
2. The existing solutions do not solve the problem and/or have significant problems/shortcomings when doing so.

Note that parts of this section are already highlighted in both the abstract and the introduction. However, in this section, you dive a bit deeper. In a good motivation, you show a (simple) example on which current methods fail, yet, the method that you are going to describe in this thesis actually yields a better result.

For example, assume that your thesis describes a new *process discovery* algorithm that is able to handle noise, incomplete behavior, and, on top of that, is able to apply label-splitting. You can take an (example) event log and show that existing algorithms result in models that are of suboptimal quality. Finally, you show a model discovered by your fancy algorithm, and, you explain why this model is so much better.

## 1.2 Problem Statement

In this section, you introduce the problem that you are solving. A good problem statement is a concise, more general statement of the example motivation that you have used in the motivation section.

For example, in the case of our previous example:

"Real event data contains infrequent and incomplete behavior. Furthermore, different recordings of the same activity may refer to conceptually different contextual executions. Existing, state-of-the-art process discovery algorithms cannot discover process models of adequate quality, given event data of the previously described form."

## 1.3 Research Questions

The research questions you pose, are questions you need to (largely) answer in order to solve the problem. Basically, the combined set of answers to the questions you pose, allow you to solve the research problem.

From the book of Justin Zobel, "Writing for Computer Science" [1] (which we highly recommend you to read before writing):

" A hypothesis or research question should be specific and precise, and should be unambiguous; the more loosely a concept is defined, the more easily it will satisfy many needs simultaneously, even when these needs are contradictory. And it is important to state what is not being proposed—what the limits on the conclusions will be. "

In the context of our example, we can define many relevant research questions:

1. What are typical prominent noise patterns in real event data?
2. How to detect noise patterns in event data?
3. How to detect infrequent behavior in event data?
4. How to detect contextually different executions of the same activity?
5. How to balance between detected noise patterns, infrequent behavior and imprecise labels?
6. ...

Typically, your thesis consists of 3-5 research questions (often multiple more questions can be defined).

## 1.4 Research Goals

In this section, you present your research goals. A research goal is a concise statement that s.t., achieving the goal helps you in (partially) solve a research question. Hence, the research questions and goals are often very related. Furthermore, it should be obvious for the reader what goal answers what problem.

In the context of our previously mentioned questions, some example goals are:

- Conduct a systematic literature review on noise patterns in real event data.
- Design a noise detection algorithm.
- Formalize the notion of incompleteness in an event log.
- ...

## 1.5 Contributions

In this section, you list the contributions that your thesis makes to our wonderful world (of science). Again, there is a strong link to the previous section. Usually, you have achieved your research goals. Hence, the contributions are concrete statements of the goals you have achieved. Additionally, your evaluation (most likely also stressed as a goal) is a contribution. Any implementation or prototype can also be quantified as a contribution.

Some examples:

- A systematic literature review covering 35 articles on noise patterns in real event data
- A noise detection algorithm based on dynamic programming and symbolic linking
- ...

## 1.6 Thesis Structure

In this section, you outline the remainder of the thesis.

The remainder of this thesis is structured as follows. In chapter 2, we discuss related work. In chapter 3, we present basic mathematical preliminaries, used throughout the thesis. ...

# Chapter 2

# Related Work

In this section, you are going to list all the related work. In this template, the related work section is located directly after the introduction chapter, however, this is not always the best/most logical location. As a general guideline, the position of the related work section can be:

- *Directly after the introduction*; Position the related work section here, in case you do not need to refer to any detailed (mathematical) concepts. Furthermore, you do not need to refer to any part of your solution. Positioning the related work section here is most commonly done if you are really uncovering "new terrain", i.e., you are doing something *very novel*.
- *Directly after preliminaries*; In this scenario, you swap the related work and preliminaries, i.e., w.r.t. this thesis template. Position this here when you do need to refer to some of the (mathematical) concepts (typically presented in the preliminaries), yet, you do not need to use any properties of your own solution when comparing your work to the related literature.
- *Directly before the Conclusion (typically after the Evaluation)*; Position the related work section here, in case you need to compare the related work against properties of your algorithm and/or the results you have obtained.

In some cases, you can introduce sub-sections in this chapter, i.e., if a division of the related work eases the readability.

Note that the related work is not intended for you to "Convince your supervisors that you have studied a lot of papers". Rather, you simply describe in 1, max 2 lines what the core idea of a paper is, how it relates to your work and why it is different from your work. In some cases, it is possible to discuss the aforementioned points for a selection of papers rather than a single paper. Try to present the paper along logical lines, i.e., define categories of work and introduce subsections for this. Never present a long 2-page list of randomly/chronologically structured work. Always add a decent structure!

# Chapter 3

# Preliminaries

In the preliminaries you present well-understood mathematical concepts that you need in your thesis. For example, you can define the natural numbers as $\mathbb{N}=\{0,1,2,...\}$, and, correspondingly $\mathbb{N}^{+}=\mathbb{N}\backslash\{0\}$. A preliminary notion is either a well-defined commonly understood mathematical notion, e.g., sets, multisets, graphs, sequences, Petri nets, ..., or, it is a concept clearly defined in another paper, i.e., you just adopt the notation (or a slight variation thereof). *Any concept you use should be defined in your thesis.* You should never write: "We use Workflow nets, a definition of these can be found here [X]". If you use it, explain it.

Concepts that are unique to your approach are not part of the preliminaries, i.e., they are described in the approach section itself.

Some useful tips:

- When introducing a complex concept, use the following structure (always works):
  - Explain the concept informally.
  - Provide a formal definition of the concept.
  - Provide an example, using the formal definition, of the concept.

  In your examples, try to be *as visual as you can*, often, an image says more than 5 pages of text.
- use commands, e.g., `\newcommand{\naturals}{\ensuremath{\mathbb{N}}}`

# Chapter 4

# Method

Describe the method/idea/algorithm in this chapter. Rename this chapter to a more meaningful name, given your topic.

The best possible Method chapter is fairly easy, it has the following sections:

1. Overview; You provide an overview of the approach. You do so by showing a graphical example of the main steps (usually 2-3 steps). For each step, you clearly indicate the inputs and the outputs of the step. Let's assume we have Step 1, 2 and 3; guess what the next sections are...
2. Step 1 (obviously a fancier name)
3. Step 2 (obviously a fancier name)
4. Step 3 (obviously a fancier name)

Describing the steps is very similar to the difficult mathematical concepts. First describe an informal description of what the step does. Then provide supporting definitions, theorems and proofs. Then show how this works on an example. In some cases, it helps to adopt a "running example" that you use throughout this section to clarify each step.

# Chapter 5

# Implementation

This section is only needed if there are non-trivial parts of your approach that require clarification. You do not need to show the design of your code, or, pseudo code. Only focus on non-trivial aspects.

For example, the Inductive Miner [2] clearly describes the requirements for the "cuts" that it finds. How you actually compute these cuts, i.e., combined with correctness proofs w.r.t. the requirements posed, is a good example of what can be added here.

Ideally, your previous chapter is so clear, that you do not need this chapter :-), i.e., implementing it is simply clear from the description.

# Chapter 6

# Evaluation

In this chapter, you present your evaluation. You should have extensively discussed with your supervisor what you are evaluating and why...

Some typical sections that you need here:

## 6.1 Experimental Setup

Describe the setup of your experiments. If you apply a pipeline of different techniques, show the pipeline. Present the parameters of your experiments in a table, and, describe them. Typical Elements:

- Input Data Set(s) Used
- Algorithm(s) Used
- Parameter(s) Used
- ...

## 6.2 Results

Show the results. Try to present your results as structured as you can. A good result (sub)section, follows the following structure:

- Describe what results you are going to Show
- Present an initial hypothesis about the results, e.g., We expect the quality metric M to behave like this, conditional to this parameter P.
- Show the Results
- Confirm the hypothesis.
- If there are results that are not according to the hypothesis, you have to be able to explain why this is the case!!!

## 6.3 Threats to Validity

Here you discuss any element of your experiments (usually depending on the setup), e.g., data sets used, parameters assessed, that might have implications for the validity of your results. For example, if you have not considered a specific range of parameter values, it is unclear how the algorithm will behave for these settings. If you have excluded certain data, this might have its implications for the generalizability of your results. In a way, this is the discussion section of your thesis.

# Chapter 7

# Discussion

In this section, you discuss your general approach. What design decisions did you make, and, what are the implications of this? Maybe your approach computes an exact intermediary result, which is costly in terms of time. Does a suitable approximation approach exist? What are the implications for that approach. You can also dive deeper in very related work and explain why it does not work in certain cases.

# Chapter 8

# Conclusion

Your conclusion actually follows the same structure as your abstract. However, you present what the reader has seen, again, using the same concepts as you have already used in your abstract.

**8.0.0.0.1 Future Work** Furthermore, you present interesting directions for future work!

# Bibliography

[1] Justin Zobel. *Writing for Computer Science*. Springer, 2014. ISBN 978-1-4471-6638-2. doi: 10.1007/978-1-4471-6639-9. URL https://doi.org/10.1007/978-1-4471-6639-9.

[2] Sander J. J. Leemans, Dirk Fahland, and Wil M. P. van der Aalst. Discovering block-structured process models from event logs - A constructive approach. In José Manuel Colom and Jörg Desel, editors, *Application and Theory of Petri Nets and Concurrency - 34th International Conference, PETRI NETS 2013, Milan, Italy, June 24-28, 2013. Proceedings*, volume 7927 of *Lecture Notes in Computer Science*, pages 311–329. Springer, 2013. doi: 10.1007/978-3-642-38697-8_17. URL https://doi.org/10.1007/978-3-642-38697-8_17.

# Appendices

# Appendix A

# Some Appendix

Try to avoid appendices. If you can't this is a place to show all kinds of results that are supportive, yet, not critical for your thesis.

# Acknowledgments

At first, I would like to express my gratitude to the awesome supervisor that gave me this template.