
EXPERIMENTAL CS THESIS TITLE

A GUIDE FOR CONTENT & STYLE

STUDENT NO 1, STUDENT ID 1

STUDENT NO 2, STUDENT ID 2

STUDENT NO 3, STUDENT ID 3

STUDENT NO 4, STUDENT ID 4

MY GROUP (IF APPLICABLE)

MASTER'S THESIS

February 2025

Advisor: Niels Olof Bouvin



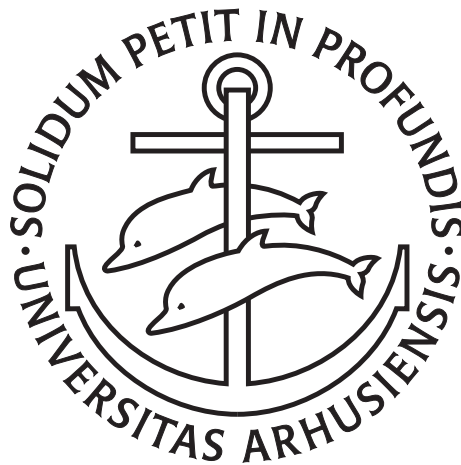
AARHUS
UNIVERSITY

DEPARTMENT OF COMPUTER SCIENCE

EXPERIMENTAL CS THESIS TITLE

A Guide for Content & Style

STUDENT 1, 2, 3 & 4



Master's Thesis

Department of Computer Science
Faculty of Natural Sciences
Aarhus University

February 2025

“Begin at the beginning,” the King said, very gravely, “and go on till you come to the end: then stop.”

Alice's Adventures in Wonderland
LEWIS CARROLL

ABSTRACT

Short summary of the contents in English. . .

CONTENTS

I The Proper Structure of a Thesis

1	Introduction	7
1.1	What Makes a Good Hypothesis?	7
1.2	Writing a Thesis Worth Reading	8
1.3	(MANDATORY) Generative AI Declaration	9
1.3.1	Using Generative AI in Your Thesis	9
2	Related Work	11
2.1	A Small Note on References	12
2.2	Frameworks and Technologies	12
2.3	Summary	12
3	Analysis	15
4	Design of Architecture and Evaluation	17
4.1	Design the System as Well as the Evaluation	17
4.2	Communicating the Design	18
5	Implementation	21
6	Evaluation	23
6.1	Some Notes on Tables, Graphs, and Figures	25
7	Conclusion	27

II Some Kind of Manual

8	Introduction	33
8.1	Organization	35
8.2	Style Options	36
8.3	Customization	37
8.4	Issues	38
8.5	Future Work	39
8.6	Beyond a Thesis	39
8.7	License	39

III The Showcase

9	Examples	43
9.1	A New Section	43
9.1.1	Test for a Subsection	43
9.1.2	Autem Timeam	44
9.2	Another Section in This Chapter	44
9.2.1	Personas Initialmente	44
9.2.2	Linguistic Registrare	45
10	Math Test Chapter	47
10.1	Some Formulas	47
10.2	Various Mathematical Examples	48

iv Appendix

A	Appendix Test	51
A.1	Appendix Section Test	51
A.2	Another Appendix Section Test	51
	Bibliography	53

LIST OF FIGURES

Figure 2.1	Getting good BibTeX references	13
Figure 4.1	Profibus states and transitions	18
Figure 4.2	A sequence diagram example	19
Figure 6.1	A graph with error bars	24
Figure 9.1	Tu duo titulo debitas latente	46

LIST OF TABLES

Table 2.1	Summary of systems	14
Table 6.1	An auto-generated table	24
Table 9.1	Autem timeam deleniti usu id	45
Table A.1	Autem usu id	51

LISTINGS

5.1	A small C program	22
A.1	A floating example (listings manual)	51

ACRONYMS

API	Application Programming Interface
DRY	Don't Repeat Yourself
JPG	Joint Photographic Experts Group
LLM	Large Language Model
PDF	Portable Document Format
PNG	Portable Network Graphics
UML	Unified Modeling Language

Part I

THE PROPER STRUCTURE OF A THESIS

The following reflects what I believe to be a good structure for a report or a thesis in experimental computer science. It contains a natural progression from the general to the specific, and from the work of others to the work by the authors, each chapter forming the foundation of the next.

Niels Olof Bouvin

SOME THOUGHTS ON TOOLING

As can be gleaned from the very existence of this guide, I very much favour PDF \LaTeX as the best way to format a thesis. Once it has been properly setup and configured, it is unparalleled in consistent quality of output. While excellent online editors exist, notably Overleaf¹, I would hesitate to recommend their use for a whole thesis, as I find that dedicated text editors, such as GNU Emacs, Vim, Sublime Text, or Visual Studio Code are vastly superior. They are mature text editing platforms, and provide excellent support, not only for \LaTeX itself, but also for versioning, and thus for collaboration.

If you cannot resist the convenience of Overleaf, I would recommend opting for a paid account during the months you are writing your thesis. There are a number of benefits, including git² and Dropbox integration, and it is not very expensive.

If you prefer a more visual tool, there are specialised \LaTeX editors, such as LyX³, which is available for Linux, Windows, and macOS; T \LaTeX nicCenter⁴ for Windows, which this very document comes prepared for; and T \LaTeX studio⁵, available for Linux, Windows, and macOS.

My own setup has for decades consisted of GNU Emacs with the packages AU \LaTeX TeX⁶ combined with Ref \LaTeX TeX⁷ (both easily installed from inside Emacs: Options → Manage Emacs Packages) and I have yet to see anything getting close to the power of that combination.

INSTALLING \LaTeX

Installing and maintaining \LaTeX used to be a bit daunting, but is today quite straightforward using T \LaTeX Live⁸ for Windows; Mac \LaTeX ⁹ for macOS; and your favourite package manager for your flavour of Linux (search for ‘texlive’).

¹ <https://www.overleaf.com/> accessed 2019/3/27.

² This alone is worth the price of admission.

³ <https://www.lyx.org/> accessed 2018/6/27.

⁴ <http://www.texniccenter.org/> accessed 2018/6/27.

⁵ <http://texstudio.sourceforge.net> accessed 2018/7/3.

⁶ <https://www.gnu.org/software/auctex/> accessed 2018/6/27.

⁷ <https://www.gnu.org/software/auctex/reftex.html> accessed 2018/6/27.

⁸ <https://tug.org/texlive/> accessed 2018/6/27.

⁹ <https://tug.org/mactex/> accessed 2018/6/27.

HANDLING BIBLIOGRAPHIES

`BIBTEX` is indispensable when it comes to handling references. I have included a starting set of references with this guide, but you will obviously need to add your own as your work progresses. That is very much aided by tools—I use the excellent Bibdesk¹⁰ (part of the Mac`TEX` distribution mentioned above) on macOS to handle my bibliographies, and a Windows alternative could be the Mendeley¹¹ desktop client, which can export to `BIBTEX`. If you find yourself with references in another format than the one you need, bibutils¹² may come in handy.

Once you have your references, proper tools, such as Ref`TEX` mentioned above, make inserting references a breeze. The best place to find references in general is Google Scholar¹³, which, just as most of the sites referenced by it, can export to `BIBTEX`. A ML powered alternative or supplement is Elicit¹⁴, which can be very useful to discover research that you might not know the right keywords for.

As you create your entries, you should take care to ensure that all fields required for others to find the referenced work are filled out correctly. See Section 2.1 for a more detailed discussion on this.

If you find an interesting paper, it is wise to download it for future reference. The above mentioned Bibdesk can help with filing the many papers you might accrue.

While Google Scholar or Elicit is fine for finding references, you will occasionally need to retrieve papers from ACM, IEEE, or some other paywalled site. The AU Library has accounts with these and many other publishers, so if you access the sites from within AU (or through the AU VPN), you can usually download without restriction.

CREATING FIGURES

`PDFLATEX` can include figures in many formats, notably [PDF](#), [PNG](#), and [JPG](#), but it is also possible to create quite sophisticated native `LATEX` figures using, e.g., Tikz¹⁵. See Section 6.1 for more details.

¹⁰ <https://bibdesk.sourceforge.net/> accessed 2018/6/27.

¹¹ <https://blog.mendeley.com/2011/10/25/howto-use-mendeley-to-create-citations-using-latex-and-bib/> accessed 2018/6/27.

¹² <https://sourceforge.net/p/bibutils/home/Bibutils/> accessed 2018/7/4.

¹³ <https://scholar.google.dk/> accessed 2018/6/27.

¹⁴ <https://elicit.com> accessed 2024/11/22.

¹⁵ <http://www.texample.net/tikz/examples/area/computer-science/> accessed 2018/6/27.

GETTING HELP

The organisation of the files comprising this template is explained in [Section 8.1](#). A good starting point for getting help would be the \LaTeX wiki books¹⁶, which does an admirable job of covering material for beginners and advanced users alike. <https://tex.stackexchange.com/> is an excellent resource for tricky \LaTeX related questions. If you prefer your reference material in hard copy, the two seminal works are *\LaTeX : A Document Preparation System: User's Guide And Reference Manual* [12] by Lamport, the creator of \LaTeX , and *The \LaTeX Companion* [13] by Mittelbach et al.

FOLDERS AND FILES: THE ORGANISATION OF THIS DOCUMENT

See [Section 8.1](#) for an explanation of the folder and file structure of this document.

THIS DOCUMENT IS DUPLEX

It may seem obvious, but just to be clear: this document style is intended to be printed duplex. You should *not* start trying to delete seemingly “empty” pages!

¹⁶ <https://en.wikibooks.org/wiki/LaTeX> accessed 2018/6/27.

1

INTRODUCTION

“Curiouser and curiouser!” cried Alice.

Alice’s Adventures in Wonderland
LEWIS CARROLL

The purpose of [Chapter 1](#) is make a short (2–6 pages) argument that should cover

- A. What this thesis is about?
- B. Why it is interesting or important?
- C. What are the central hypotheses that will be investigated?
- D. How will the work be done?

This is the place where the reader (who will be a computer scientist, but might not be a domain expert) should be convinced that not only is the topic interesting and important, the authors have also identified central questions/hypotheses pertaining to the topic, and have a clear plan and methodology to address it.

1.1 WHAT MAKES A GOOD HYPOTHESIS?

A hypothesis must be a “good question” in the sense that the question is an interesting one and that the answer should not be obvious to a computer scientist (or CS/IT student). If you know the answers in advance, you really should be asking different questions.

For the purposes of a report or thesis, it is wise to concentrate on research questions and hypotheses that are decidable or quantifiable. E.g., it is better to state that “method A is better than method B under circumstances C” or “combining method A with architecture B improves on standard approach C” than “we can build a system that does X”. This is why it is always a good idea to include baselines in your work, i.e., established methods or architectural choices that can be used for comparison. If you do not have baselines yourself, you should at least be ready and able to compare your results with the published results of others.

If your thesis work is exploring so-called a “wicked problem¹”, the validation of your work will rely on other criteria than quantifiable

¹ https://en.wikipedia.org/wiki/Wicked_problem accessed 2024/11/22

measurements and rejections of hypotheses. Research through design is a young field and quality criteria are currently debated and developed. You may work with Zimmerman, Forlizzi, and Evenson [19], draw upon Gaver [8] and work with artefacts as theory nexus, or as annotated portfolios. Alternatively, you can approach HCI research as problem solving, as suggested by Oulasvirta and Hornbæk:

A research problem in HCI is a stated lack of understanding about some phenomenon in human use of computing, or stated inability to construct interactive technology to address that phenomenon for desired ends. [Definition 1: [14]]

The hypotheses should address central aspects of the work, so that *if* these hypotheses are met, the overall work gains in credibility, or alternatively (and just as valid), if a hypothesis *cannot* be confirmed, it illustrates why and how the assumptions behind the work were flawed, and, hopefully, how they can be improved.

1.2 WRITING A THESIS WORTH READING

The purpose of the thesis is to be read as a whole in one sitting, and with this in mind it should be written, even if, in reality, it is authored over a period of months. The reader does not naturally understand the flow and process of the work involved (this understanding, hopefully, belongs to the authors), and must therefore be guided by the authors through the work. In order to accomplish this, the readers should at all times have a ready answer in their mind to the following questions:

- Why am I reading this?
- What comes next?
- How does this build upon what I just read?

So, why is something there? What is its purpose? How will it be used later? Vice versa, later in the text, refer back to things established earlier (this also supports readers that do not necessarily read linearly). While a text grows piecemeal, it is most often read as a whole, and should appear as such, lest the reader loses interest.

It is to that end a good idea to finish the introduction with a description of how the hypotheses are to be investigated, and how this is reflected in the structure of the thesis.

1.3 (MANDATORY) GENERATIVE AI DECLARATION

You should here state *clearly*, how Generative AI has been used in your thesis work, whether it has been as a programming aid, tool to clean-up your text, or tool to generate text or figures. You should throughout the thesis adhere to the advice given below, when it comes to references and attributions.

If you have *not* at *any* point during your work used Generative AI, you should state so here—a declaration is necessary and mandatory.

1.3.1 Using Generative AI in Your Thesis

It is now permitted to use GenAI at AU², but you must treat this kind of tool with caution as you would any outside help. *You* are, after all, the one writing the thesis, *not* some machine!

GOOD USES FOR GENAI Generative AI, e.g., ChatGPT, can be excellent tools for learning about new topics or summarising some area of interest. As such, it can be a powerful supplement to ordinary search engine use and literature perusal. You should however always keep in mind that GenAI is prone to hallucinations, and thus verify any claims made. A Large Language Model (LLM) may seem confident in its output, but it is just an algorithm generating texts probabilistic based on a training corpus, and thus produces texts *likely* rather than *true*.

Generative AI can be very useful, when it comes to correcting grammar, or cleaning up text. Ideally, you should use GenAI as a tutor providing critique—using its suggestions to improve your own writing, so that you, over time, do not require its assistance. An effective way of doing this is to instruct the LLM to behave in a specific way, e.g., preface the text you wish improved with something like this:

“I wish you to critique the following Master’s Thesis abstract as a seasoned Computer Science professor and thesis supervisor would. You are not to offer a corrected version of the text, but you are to point out strengths and weaknesses, so that the recipient can correct the text themselves.”

Voilà: a thesis supervisor in a box!

If you use generative AI to alter and improve your text, you must state so in the introduction.

Incidentally, ChatGPT can also be a great help, when it comes to figuring out how to do something in L^AT_EX—here, the machine’s ability to ingest vast amounts of documentation, and answer questions based

² <https://studerende.au.dk/en/gai>

on it becomes *very* useful (as long as you understand that it can still occasionally hallucinate, inventing fictitious commands).

QUESTIONABLE USES FOR GENAI More problematic is using such tools to generate actual thesis content from prompts. If you do that, you *must* indicate that this is machine generated text and provide the prompt used. This is just like providing a reference to a quotation from a scientific publication: including material without attribution is *plagiarism*, i.e., exam fraud.

While machine-generated text can efficiently produce content, relying solely on such output may compromise the integrity and authenticity of the work. Crafting original text allows for the infusion of personal insights, nuances, and creativity, tailored to the specific context and audience. This process fosters deeper engagement and understanding, facilitating the conveyance of complex ideas with clarity and precision. Moreover, writing one's own text encourages critical thinking, problem-solving, and mastery of the subject matter, nurturing intellectual growth and fostering a sense of ownership and pride in the work. Therefore, while machine-generated text may serve as a useful tool for generating ideas or assisting in drafting, the act of writing one's own text remains invaluable for cultivating expertise, expression, and scholarly contribution. [ChatGPT V4, 2024-05-08: "Write a short, succinct paragraph on why it is better to write your own text rather than relying on machine generated ditto in the style of a scientific publication"]

2 | RELATED WORK

The time has come,' the Walrus said,
To talk of many things:
Of shoes — and ships — and sealing-wax —
Of cabbages — and kings —
And why the sea is boiling hot —
And whether pigs have wings.'

The Walrus and the Carpenter
LEWIS CARROLL

Whereas the purpose of [Chapter 1](#) was to entice and convince the reader that the work reported is interesting, that the author is asking the right questions about it, and that reading about the answers will be worthwhile, the purpose of [Chapter 2](#) is to demonstrate that the author possesses a fine overview and keen understanding of the overall topic of the work. Note, that while the title of the chapter is “Related Work”, it might as well be called “*Relevant Work*” in that you should only include work that are directly useful or relevant to your purpose.

Writing about others’ work can be challenging—it is easy to succumb to just writing condensed summaries, which are just as tedious to read as they are to write. A better method is to gain an overview over the field of inquiry, and then establish in the first section the aspects or dimensions that are crucial to systems or methodologies such as the ones described. This demonstrates to the reader that the author has understanding and judgement. Having done this, every paper or work covered can then be described in those established terms. This makes for easier and far more structured writing, and it also helps the reader differentiate the systems and works reported on. If there are multiple works that cover approximately the same area (e.g., using the same technique), you may cite several, but only go into detail with the most significant or representative one.

The chapter can then be concluded with a table (see [Table 2.1](#) on [page 14](#)) summarising all the work reported on using the aspects defined in the introduction of the chapter.

One aspect to keep in mind when reading others’ work is how they performed their evaluation. This can both be a source of inspiration for your own evaluation in [Chapter 6](#), and a point of critique, if their evaluation is weak. It will also often be the case that different papers will have different kinds of evaluation, making a direct comparison tricky (and potentially something worth investigating).

A crucial element of this chapter is that it concerns the work of others and *only* that. While the selection of aspects or dimensions

described above is a reflection of your own focus, that should be the extent to which your own work and plans influence this chapter. Your own judgement comes in the next chapter.

2.1 A SMALL NOTE ON REFERENCES

It is essential for any scientific work to cite its sources. This can be done relatively painlessly using BibTeX as outlined on [page 4](#). It is important that you include all necessary information in the bibliography for others to correctly identify and locate the referenced work, and the easiest way to do that is to copy the BibTeX reference straight from ACM, as seen in [Figure 2.1](#). Other publishers usually also have BibTeX references available. If you cite one work, it should be done thus [11], if you cite a specific page in a work, it should be done thus [3, p. 410], and if you cite multiple works, it should be done thus [9–11, 13]. I generally reserve the bibliography for works that have been properly published and/or peer reviewed. References to Web pages are best handled through footnotes¹ (or marginalia, if they are very short), though some things, such as RFCs and other standard documents, belong in the bibliography proper. If you use a figure from another work, you *must* give attribution—otherwise it counts as plagiarism, which is a serious offence.

<http://lyx.org/>

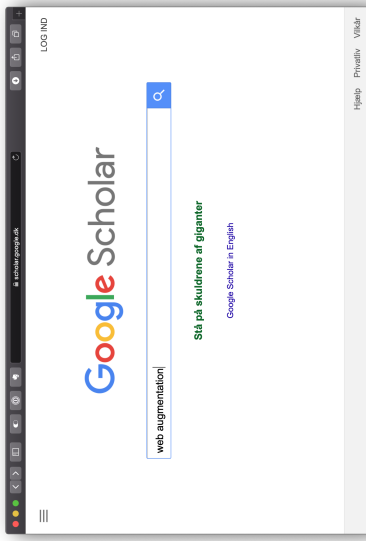
2.2 FRAMEWORKS AND TECHNOLOGIES

Related work need not only be published academic work. In many cases, it is also relevant to describe crucial frameworks and technologies that will be used or are relevant for the thesis. This does not mean that all employed technologies should be described in detail, but frameworks and technologies that are unusual (for lack of a better word) could be described here. E.g., there is no need to describe an ordinary network stack, but if the work involves GPU programming, a description of the chosen architecture might well be relevant, as it informs all the following chapters. In short, if it is something a CS generalist should be familiar with, there is no need to describe it.

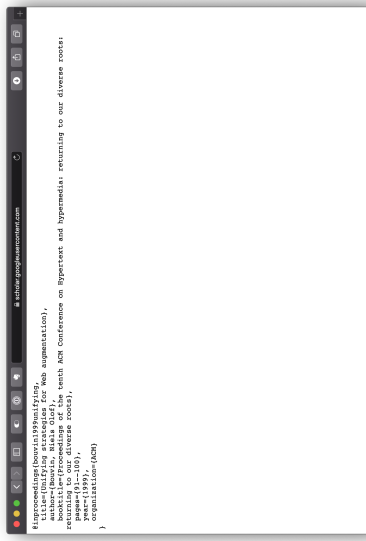
2.3 SUMMARY

As described above, ending the chapter with a figure like [Table 2.1](#) summarising the findings can be a great way to remind the reader of the results, as well as laying the foundation for [Chapter 3](#).

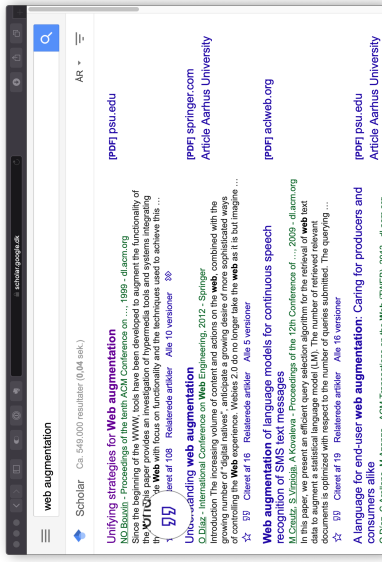
¹ <https://www.tug.org/applications/pdftex/> accessed 2018/6/27.



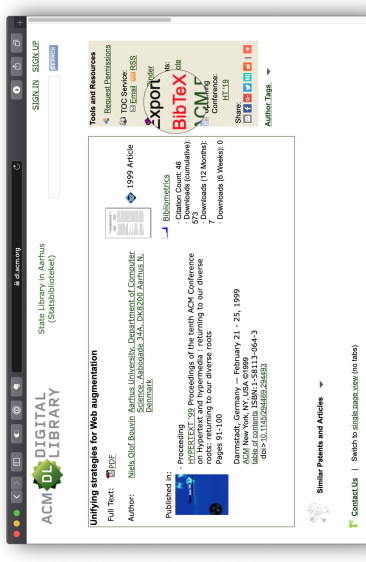
(a) Google Scholar.



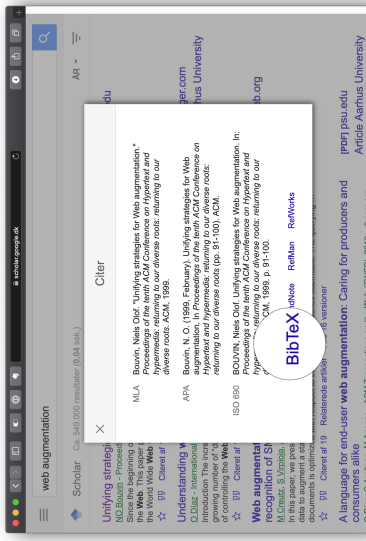
(d) Copy the reference.



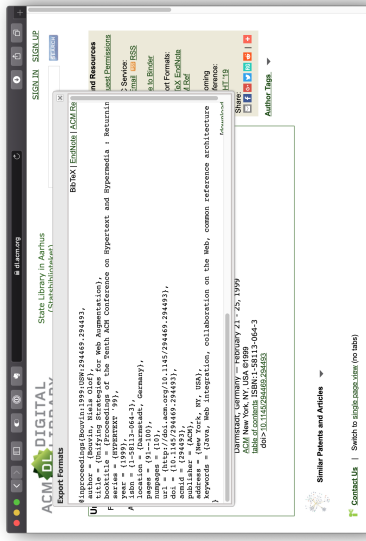
(b) Press "'".



(e) Follow the ACM link and press BibTeX.



(c) Press 'BibTeX'.



(f) Copy the much better reference.

Figure 2.1: Two different ways to get good BibTeX references. The ACM method is recommended as it provides a much higher quality reference, whereas the Google Scholar reference is merely scraped.

SYSTEM	SUPER ASPECT			SUPER ASPECT		ASPECT
	ASPECT	ASPECT	ASPECT	ASPECT	ASPECT	
Foo	Y	(Y)	N	(Y)	N/A ^a	43.32
Bar	(Y)	Y	(N)	N	(Y) ^b	7.43
Baz	(Y)	N	N/A	Y	Y	−9.87
Quux	N	(Y)	(Y)	(Y)	Y ^c	20 293.6

Table 2.1: The systems and papers described in [Chapter 2](#). The systems have been rated along the chosen aspects. Note, how this description is very much longer than what appears in the List of Tables. Using both a short and long description is a good idea—it leaves a tidy index and provides rich information where it is needed. This table is rotated (see the source to see how)—not because it is strictly needed, but to show how it might be done with wider figures. Numbers can be easily be aligned and typeset using `siunitx`.

^a Not Applicable

^b Only for positive integers, and not on Thursdays

^c Not tested on ducks

3

ANALYSIS

"Who in the world am I? Ah, *that's* the great puzzle!"

Alice's Adventures in Wonderland
LEWIS CARROLL

This is where the authors can answer the question of what use we can derive from all the works described in [Chapter 2](#). Ideally, the summary of the related work will show that there is room unexplored for what the authors have in mind. If there are differences between the included works on key aspects in the approach to be taken, this is where this should be identified, and a decision reached.

It might well be that some decision must be made at this point between one technical approach versus another, e.g., does a particular framework or technique work for the intended purpose? These things should be decided upon before the experimental system can be designed, and this chapter is the place to do so. Such decisions should be backed by experimental data, demonstrating one solution's superiority over the others. At other times, the choice can be made part of the hypotheses stated in [Section 1.1](#), and then the experiments must be postponed until the evaluation in [Chapter 6](#).

Having written the analysis, the authors have all the tools and arguments needed to complete [Chapter 4](#).

4

DESIGN OF ARCHITECTURE AND EVALUATION

"Would you tell me, please, which way I ought to go from here?"

"That depends a good deal on where you want to get to," said the Cat.

"I don't much care where—" said Alice.

"Then it doesn't matter which way you go," said the Cat.

Alice's Adventures in Wonderland
LEWIS CARROLL

Many of the other natural sciences have labs with equipment that has to be configured correctly to experimentally test stated hypotheses. Such experiments must be meticulously planned and designed in advance to work properly and provide valid and trustworthy results.

As computer scientists, we usually do not work in labs, and our experiments do not live in Petri dishes. Still, we have hypotheses to test, and thus, experiments to plan. This planning phase is the design of the experiment, where the authors describe the system intended to test the hypotheses posed in the introduction.

It is far easier to reason about the design of a system on a whiteboard than it is to change what has already been committed to code, and it is wise to carefully consider all aspects of your design, before starting to try to build it.

4.1 DESIGN THE SYSTEM AS WELL AS THE EVALUATION

Some hypotheses can be investigated wholly *in vitro*, testing, e.g., one algorithm against another. Other hypotheses require us to investigate further, involving, e.g., potential users or domain experts to properly evaluate our assumptions. It is therefore crucial to consider not only *what* you wish to create, but *how* you propose to evaluate it. For your study to have validity, the design of the evaluation is every bit as crucial as the design of the object being evaluated. How can the hypotheses described in [Chapter 1](#) be investigated? What is going to be measured? How will this be accomplished? If the reader is given a clear plan here, it should also be clear what must be implemented in order to perform the outlined experiments. The credibility of your study also relies on your ability to communicate the process with which you have reached your design.

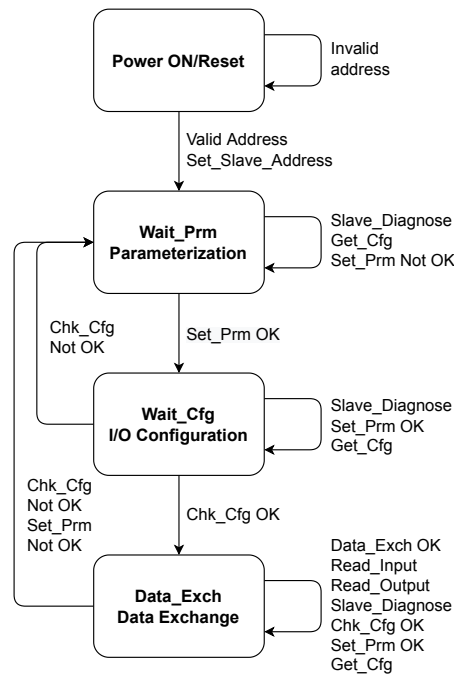


Figure 4.1: Profibus states and transitions ([15] used with permission)

4.2 COMMUNICATING THE DESIGN

If you have a protocol, or a system that can alternate between several states in a well-defined manner, a state machine is a very concise way to communicate this—consider Figure 4.1 and how much text that would have required to elucidate.

Another efficient way of communicating systems design is through the use of Unified Modeling Language (UML) diagrams, both structurally using class diagrams, and behaviourally using sequence diagrams, see Figure 4.2, both of which can be created without any external applications. These diagrams can later be referred to, or modified in the next chapter, as there may well be some differences between your design and your implementation.

Indeed, a luxury of this chapter is that the design may well go further than solely the confirmation or refutation of the hypotheses. If you are building a system, this is where you show that you know how to design one, even if you will actually not be implementing all of it. If you had sufficient time and resources, *this* is how you would make your system.

However, before we come to that, it is necessary to investigate whether the required hypotheses are valid. If they are not, the design must be reconsidered, and there is only one way to test them, namely through implementation, and subsequent evaluation.

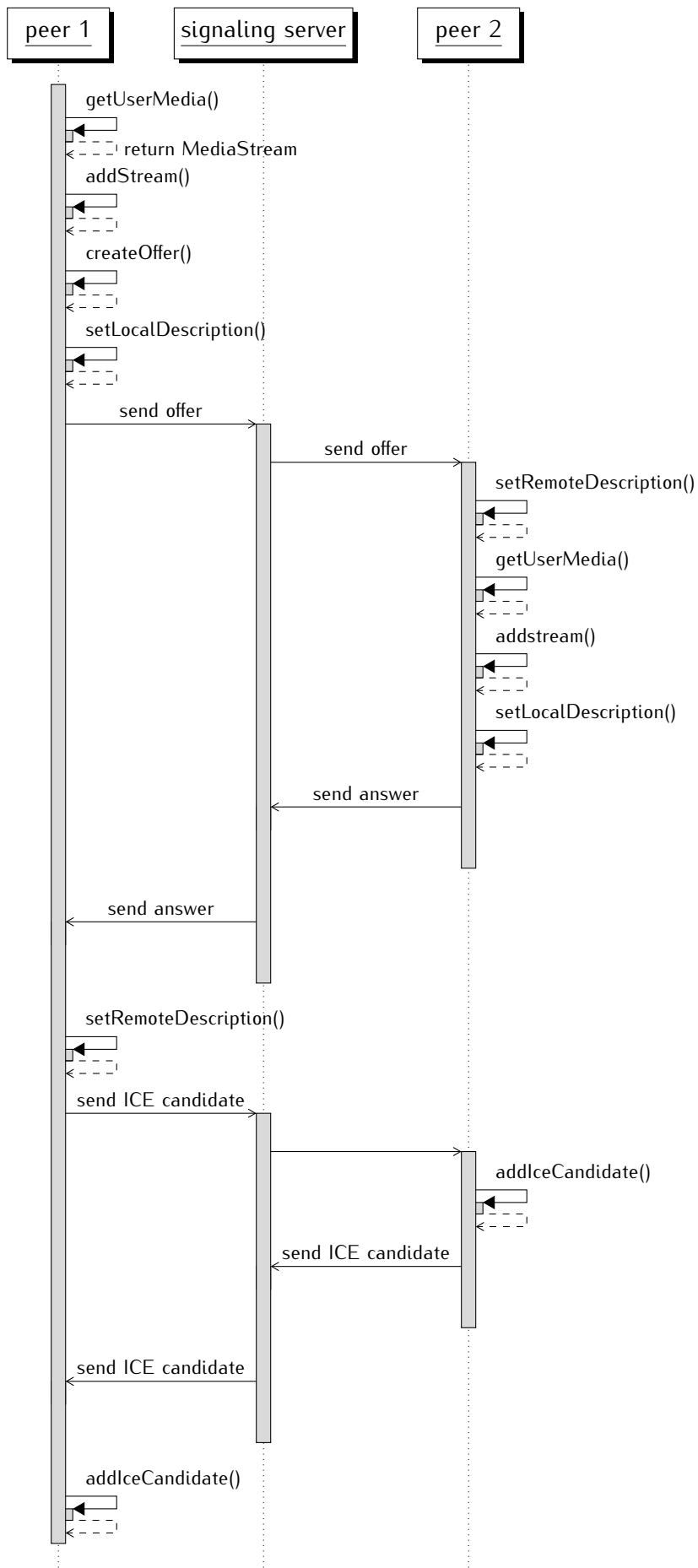


Figure 4.2: A sequence diagram can communicate the interaction between components efficiently (adapted from [17] with permission)

5

IMPLEMENTATION

"The rule is, jam to-morrow and jam
yesterday—but never jam to-day."

Through the Looking-Glass
LEWIS CARROLL

Whereas the previous chapter concerned itself with the overall plan in the abstract, this is where the actual experiment in the form of an implementation is taking form. It is not the purpose of the implementation to fully realise the design described in the previous chapter. It is the exclusive purpose of the implementation (a subset of the design) to either validate or refute the hypotheses put forth in the introduction. This, and nothing else. If it does less, you have posed questions you are not prepared to answer; if it does more, you should be coding less or asking additional questions.

The primary purpose of this chapter is to clearly communicate what has been built, and how it works. This can, e.g., include architectural diagrams, software and hardware overviews and specifics.

If it illustrates core aspects, e.g., the inner working of a particular important algorithm or function, code segments are welcome in this chapter, as long as they are short, to the point, well-commented and -formatted. For algorithms, pseudo code is often clearer than actual code, and for, e.g., [API](#) examples the reverse holds true. It is also a good idea to provide the reader with a general overview of the structure of the code, as well as how communication between various parts takes place. As in the previous chapter, I recommend using [UML](#) for this purpose. The complete code (as well as your data) should be included separately with your report in the form of a zip-file, USB-stick, or link to a repository.

Overall, the implementation is the computer scientist's equivalent of lab equipment carefully arranged into an experimental setup, and just as the validity of an experimental investigation will be judged in part on the craftsmanship of the setup, so will the quality of your implementation. It is therefore important to clearly communicate how your system works and how it was built, so that the reader may have confidence in your evaluation and conclusions.

Listing 5.1: A small C program

```
1  #include <stdio.h>
2  #include <stdlib.h>
3
4  int main(int argc, char *argv[])
5  {
6      FILE *fp;
7
8      if (argc != 2) {
9          fprintf(stderr, "Usage: %s <file>\n", argv[0]);
10         exit(EXIT_FAILURE);
11     }
12
13     fp = fopen(argv[1], "r");
14     if (fp == NULL) {
15         perror(argv[1]);
16         exit(EXIT_FAILURE);
17     }
18
19     /* Other code omitted */
20
21     fclose(fp);
22     exit(EXIT_SUCCESS);
23 }
```

6

EVALUATION

"I daresay you haven't had much practice,"
said the Queen. "When I was your age, I
always did it for half-an-hour a day. Why,
sometimes I've believed as many as six
impossible things before breakfast."

Through the Looking-Glass
LEWIS CARROLL

Having built the equivalent of a experimental setup, it is time to use the implementation to test the hypotheses.

It is a good idea at this point to remind the reader of the original hypotheses, so that you can communicate clearly how evaluating these hypotheses can be broken down into stages and subquestions manifested as a series of experiments.

One structured approach to performing and reporting on these experiments is to follow the following pattern for every single experiment:

1. What is the purpose of the experiment?
2. What is the expected outcome?
3. What are the parameters under which the experiment takes place?
4. What are the results?
5. How do the results align with the expected outcome? If they do not align, why is that so?

Results should be presented summarised. For quantitative experiments, this will usually be in the form of tables and graphs. Remember to note the number of times experiments were repeated, as well as averages, and standard deviations (in percent of the mean). There is much more to the proper statistical evaluation of experimental data than can be expounded upon here, but I turn the reader's attention to [5], which is freely available and written with CS students in mind.

If your results are of a qualitative nature, the summaries will depend on the type of investigation you have done. It can be carefully annotated recordings of specific incidents of the system in use; analysis with quotes from interviews; or results from questionnaires and other investigations. Whole transcripts belong in an appendix, but excerpts are fine in the main text.

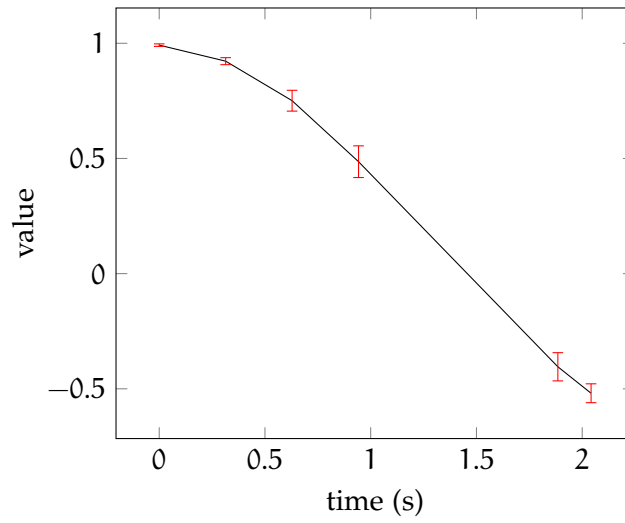


Figure 6.1: Generating figures directly in \LaTeX is possible, and they can be very consistent stylewise with the rest of the document.

TIME	VALUE	$\sigma\%$
0.00	0.99	0.6
0.31	0.92	1.7
0.63	0.75	6.0
0.94	0.49	14.2
1.88	-0.40	15.1
2.04	-0.52	7.9

Table 6.1: This table has been generated from a `.csv` file, which sometimes can be very handy and a great timesaver. Note, how numbers have been automatically normalised and aligned properly.

6.1 SOME NOTES ON TABLES, GRAPHS, AND FIGURES

It is paramount that you clearly and concisely present your results, and that usually entails documentation in the form of tables or figures. Such things are more informative if they are clearly formatted and legible.

There are many tools with which you can plot graphs of your data. I would suggest that you use something that can be scripted, so you can tweak your parameters, run the evaluation, and see the fresh results, again and again. Having to, e.g., copy and paste or import data into an application will slow you down, and might well lead to confusion—is this the new data or not? A small investment upfront in scripting everything will lead to savings in both time and frustration later on.

If you are fluent in Python, Matplotlib¹ is very good, and can be used interactively in Jupyter². Gnuplot³ is another powerful choice, and probably easier to get started with. If you know R⁴, ggplot2⁵ is the obvious route. For web-based systems, D3.js⁶ is quite popular.

As long as your tool of choice can generate PDF, you should be golden, as those are easy to include in PDF/L^AT_EX and scale well. It is a very good idea to try to print pages with figures on them—you may *think* that legends and labels are legible when looking at a zoomed-in PDF, but a paper copy might well tell a different story.

If you output your data to comma or tab separated files, you can even generate your plots and tables directly in L^AT_EX using Pgfplots⁷ (Overleaf has a gentle introduction⁸): Figure 6.1 was generated using Pgfplots and Table 6.1 using PgfplotsTable⁹. Doing this can quickly save you time, as especially L^AT_EX tables are tedious and error prone to edit. For more help on tables, see the L^AT_EX Wiki books¹⁰.

Booktabs [7]¹¹ is a package by Fear for creating publication ready tables, and it is used throughout this document.

¹ <https://matplotlib.org/> accessed 2018/6/27.

² <https://jupyter.org/> accessed 2018/6/27.

³ <http://gnuplot.info/> accessed 2018/6/27.

⁴ <https://www.r-project.org/> accessed 2018/6/27.

⁵ <http://ggplot2.org/> accessed 2018/6/27.

⁶ <https://d3js.org/> accessed 2018/6/27.

⁷ <http://pgfplots.sourceforge.net/> accessed 2018/6/27

⁸ https://www.overleaf.com/learn/latex/Pgfplots_package accessed 2018/6/17

⁹ <http://pgfplots.sourceforge.net/pgfplotstable.pdf> accessed 2018/6/27.

¹⁰ <https://en.wikibooks.org/wiki/LaTeX/Tables> accessed 2018/6/27.

¹¹ <http://mirror.ctan.org/macros/latex/contrib/booktabs/> accessed 2024/5/6.

7

CONCLUSION

"Tut, tut, child!" said the Duchess.
"Everything's got a moral,
if only you can find it."

Alice's Adventures in Wonderland
LEWIS CARROLL

This, then, is the grand summary of what you have accomplished. You may well imagine that many readers will read [Chapter 1](#), and then skip right to [Chapter 7](#), and if, and only if, those two parts are interesting, might be tempted to read the rest. A consequence is that you should ensure that the reader will gain a good overall understanding of what you have done by reading only the conclusion. Thus, this is a place to summarise all that has gone before, before finally concluding on the results of your experiments and the validity of your hypotheses. It is also important to ensure that [Chapter 1](#) (which in all likelihood was written first) still aligns closely with the work done and the conclusions reached.

Apart from reporting on your results, i.e., your product, this is also a good place to reflect on your process, and to discuss the wider consequences or ramifications of your work. What would you like to impart to others going down your path of inquiry? What would you have liked to have known, when you first began your work?

If you so desire, this is also where you might add a section on Future Work, where you point in the directions that should be followed to complete the work you have already accomplished. This should, however, probably be the very last thing on your to-do list.

Part II

SOME KIND OF MANUAL

*We have seen that computer programming is an art,
because it applies accumulated knowledge to the world,
because it requires skill and ingenuity, and especially
because it produces objects of beauty.*

— Donald E. Knuth [9]

ACKNOWLEDGMENTS

Put your acknowledgments here.

Many thanks to everybody who already sent me a postcard!

Regarding the typography and other help, many thanks go to Marco Kuhlmann, Philipp Lehman, Lothar Schlesier, Jim Young, Lorenzo Pantieri and Enrico Gregorio¹, Jörg Sommer, Joachim Köstler, Daniel Gottschlag, Denis Aydin, Paride Legovini, Steffen Prochnow, Nicolas Repp, Hinrich Harms, Roland Winkler, Jörg Weber, Henri Menke, Claus Lahiri, Clemens Niederberger, Stefano Bragaglia, Jörn Hees, and the whole L^AT_EX-community for support, ideas and some great software.

Regarding L_YX: The L_YX port was initially done by *Nicholas Mariette* in March 2009 and continued by *Ivo Pletikosić* in 2011. Thank you very much for your work and for the contributions to the original style.

¹ Members of GuIT (Gruppo Italiano Utilizzatori di T_EX e L^AT_EX)

8

INTRODUCTION

This bundle for L^AT_EX has two goals:

1. Provide students with an easy-to-use template for their Master's or PhD thesis. (Though it might also be used by other types of authors for reports, books, etc.)
2. Provide a classic, high-quality typographic style that is inspired by Bringhurst's *"The Elements of Typographic Style"* [2].

The bundle is configured to run with a *full* MiK_TE_X or T_EXLive¹ installation right away and, therefore, it uses only freely available fonts. (Minion fans can easily adjust the style to their needs.)

People interested only in the nice style and not the whole bundle can now use the style stand-alone via the file `classicthesis.sty`. This works now also with "plain" L^AT_EX.

As of version 3.0, `classicthesis` can also be easily used with L_YX² thanks to Nicholas Mariette and Ivo Pletikosić. The L_YX version of this manual will contain more information on the details.

This should enable anyone with a basic knowledge of L^AT_EX 2_ε or L_YX to produce beautiful documents without too much effort. In the end, this is my overall goal: more beautiful documents, especially theses, as I am tired of seeing so many ugly ones.

The whole template and the used style is released under the GNU General Public License.

If you like the style then I would appreciate a postcard:

André Miede
Detmolder Straße 32
31737 Rinteln
Germany

The postcards I received so far are available at:

<http://postcards.miede.de>

So far, many theses, some books, and several other publications have been typeset successfully with it. If you are interested in some typographic details behind it, enjoy Robert Bringhurst's wonderful book.

A well-balanced line width improves the legibility of the text. That's what typography is all about, right?

¹ See the file `LISTOFFILES` for needed packages. Furthermore, `classicthesis` works with most other distributions and, thus, with most systems L^AT_EX is available for.

² <http://www.lyx.org>

IMPORTANT NOTE: Some things of this style might look unusual at first glance, many people feel so in the beginning. However, all things are intentionally designed to be as they are, especially these:

- No bold fonts are used. Italics or spaced small caps do the job quite well.
- The size of the text body is intentionally shaped like it is. It supports both legibility and allows a reasonable amount of information to be on a page. And, no: the lines are not too short.
- The tables intentionally do not use vertical or double rules. See the documentation for the booktabs package for a nice discussion of this topic.³
- And last but not least, to provide the reader with a way easier access to page numbers in the table of contents, the page numbers are right behind the titles. Yes, they are *not* neatly aligned at the right side and they are *not* connected with dots that help the eye to bridge a distance that is not necessary. If you are still not convinced: is your reader interested in the page number or does she want to sum the numbers up?

Therefore, please do not break the beauty of the style by changing these things unless you really know what you are doing! Please.

YET ANOTHER IMPORTANT NOTE: Since classicthesis' first release in 2006, many things have changed in the L^AT_EX world. Trying to keep up-to-date, classicthesis grew and evolved into many directions, trying to stay (some kind of) stable and be compatible with its port to L^AX. However, there are still many remains from older times in the code, many dirty workarounds here and there, and several other things I am absolutely not proud of (for example my unwise combination of KOMA and titlesec etc.).

*An outlook into the
future of
classicthesis.*

Currently, I am looking into how to completely re-design and re-implement classicthesis making it easier to maintain and to use. As a general idea, classicthesis.sty should be developed and distributed separately from the template bundle itself. Excellent spin-offs such as arsclassica could also be integrated (with permission by their authors) as format configurations. Also, current trends of microtype, fontspec, etc. should be included as well. As I am not really into deep L^AT_EX programming, I will reach out to the L^AT_EX community for their expertise and help.

³ To be found online at <http://mirror.ctan.org/macros/latex/contrib/booktabs/>.

8.1 ORGANIZATION

A very important factor for successful thesis writing is the organization of the material. This template suggests a structure as the following:

You can use these margins for summaries of the text body...

- `Chapters/` is where all the “real” content goes in separate files such as `Chapter01.tex` etc.
- `FrontBackMatter/` is where all the stuff goes that surrounds the “real” content, such as the acknowledgments, dedication, etc.
- `gfx/` is where you put all the graphics you use in the thesis. Maybe they should be organized into subfolders depending on the chapter they are used in, if you have a lot of graphics.
- `Bibliography.bib`: the BibTeX database to organize all the references you might want to cite.
- `classicthesis.sty`: the style definition to get this awesome look and feel. Does not only work with this thesis template but also on its own (see folder `Examples`). Bonus: works with both L^AT_EX and pdfL^AT_EX...and L_XY.
- `ClassicThesis.tcp` a T_EXnicCenter project file. Great tool and it's free!
- `ClassicThesis.tex`: the main file of your thesis where all gets bundled together.
- `classicthesis-config.tex`: a central place to load all nifty packages that are used.

Make your changes and adjustments here. This means that you specify here the options you want to load `classicthesis.sty` with. You also adjust the title of your thesis, your name, and all similar information here. Refer to [Section 8.3](#) for more information.

This had to change as of version 3.0 in order to enable an easy transition from the “basic” style to L_XY.

In total, this should get you started in no time.

8.2 STYLE OPTIONS

There are a couple of options for `classicthesis.sty` that allow for a bit of freedom concerning the layout:

...or your supervisor might use the margins for some comments of her own while reading.

- General:
 - `drafting`: prints the date and time at the bottom of each page, so you always know which version you are dealing with. Might come in handy not to give your Prof. that old draft.
- Parts and Chapters:
 - `parts`: if you use Part divisions for your document, you should choose this option. (Cannot be used together with `nochapters`.)
 - `nochapters`: allows to use the look-and-feel with classes that do not use chapters, e.g., for articles. Automatically turns off a couple of other options: `eulerchapternumbers`, `linedheaders`, `listsseparated`, and `parts`.
 - `linedheaders`: changes the look of the chapter headings a bit by adding a horizontal line above the chapter title. The chapter number will also be moved to the top of the page, above the chapter title.
- Typography:
 - `eulerchapternumbers`: use figures from Hermann Zapf's Euler math font for the chapter numbers. By default, old style figures from the Palatino font are used.
 - `beramono`: loads Bera Mono as typewriter font. (Default setting is using the standard CM typewriter font.)
 - `eulermath`: loads the awesome Euler fonts for math. Palatino is used as default font.
 - `pdfspacing`: makes use of `pdftex`' letter spacing capabilities via the `microtype` package.⁴ This fixes some serious issues regarding math formulæ etc. (e.g., “ß”) in headers.
 - `minionprospacing`: uses the internal `textssc` command of the `MinionPro` package for letter spacing. This automatically enables the `minionpro` option, overriding `pdfspacing`.
- Table of Contents:
 - `tocaligned`: aligns the whole table of contents on the left side. Some people like that, some don't.
 - `dottedtoc`: sets `pagenumbers` flushed right in the table of contents.

⁴ Use `microtype`'s `DVIoutput` option to generate DVI with `pdftex`.

- `manychapters`: if you need more than nine chapters for your document, you might not be happy with the spacing between the chapter number and the chapter title in the Table of Contents. This option allows for additional space in this context. However, it does not look as “perfect” if you use `\parts` for structuring your document.
- Floats:
 - `listings`: loads the `listings` package (if not already done) and configures the List of Listings accordingly.
 - `floatperchapter`: activates numbering per chapter for all floats such as figures, tables, and listings (if used).
 - `subfig(ure)`: is passed to the `tocloft` package to enable compatibility with the `subfig(ure)` package. Use this option if you want use `classicthesis` with the `subfig` package.

The best way to figure these options out is to try the different possibilities and see what you and your supervisor like best.

In order to make things easier, `classicthesis-config.tex` contains some useful commands that might help you.

8.3 CUSTOMIZATION

This section will show you some hints how to adapt `classicthesis` to your needs.

The file `classicthesis.sty` contains the core functionality of the style and in most cases will be left intact, whereas the file `classicthesis-config.tex` is used for some common user customizations.

The first customization you are about to make is to alter the document title, author name, and other thesis details. In order to do this, replace the data in the following lines of `classicthesis-config.tex`:

```

1 % *****
2 % 2. Personal data and user ad-hoc commands
3 % *****
4 \newcommand{\myTitle}{A Classic Thesis Style\xspace}
5 \newcommand{\mySubtitle}{An Homage to...\xspace}

```

*Modifications in
classic-
thesis-config.tex*

Further customization can be made in `classicthesis-config.tex` by choosing the options to `classicthesis.sty` (see [Section 8.2](#)) in a line that looks like this:

```

1 \PassOptionsToPackage{eulerchapternumbers,drafting,listings,subfig,  
   eulermath,parts}{classicthesis}

```

Many other customizations in `classicthesis-config.tex` are possible, but you should be careful making changes there, since some changes could cause errors.

*Modifications in
classicthesis.sty*

Finally, changes can be made in the file `classicthesis.sty`, although this is mostly not designed for user customization. The main change that might be made here is the text-block size, for example, to get longer lines of text.

8.4 ISSUES

This section will list some information about problems using `classicthesis` in general or using it with other packages.

Beta versions of `classicthesis` can be found at Bitbucket:

<https://bitbucket.org/amiede/classicthesis/>

There, you can also post serious bugs and problems you encounter.

Compatibility with the `glossaries` Package

If you want to use the `glossaries` package, take care of loading it with the following options:

```
1 \usepackage[style=long,nolist]{glossaries}
```

Thanks to Sven Staehs for this information.

Compatibility with the (Spanish) `babel` Package

Spanish languages need an extra option in order to work with this template:

```
1 \usepackage[spanish,es-lcroman]{babel}
```

Thanks to an unknown person for this information (via the issue reporting).

FURTHER INFORMATION FOR USING `classicthesis` WITH SPANISH (IN ADDITION TO THE ABOVE) In the file `ClassicThesis.tex` activate the language:

```
1 \selectlanguage{spanish}
```

If there are issues changing `\tablename`, e.g., using this:

```
1 \renewcommand{\tablename}{Tabla}
```

This can be solved by passing `es-tabla` parameter to `babel`:

```
1 \PassOptionsToPackage{es-tabla,spanish,es-lcroman,english}{babel}
2 \usepackage{babel}
```

But it is also necessary to set `spanish` in the `\documentclass`. Thanks to Alvaro Jaramillo Duque for this information.

Compatibility with the pdfsync Package

Using the pdfsync package leads to linebreaking problems with the graffito command. Thanks to Henrik Schumacher for this information.

8.5 FUTURE WORK

So far, this is a quite stable version that served a couple of people well during their thesis time. However, some things are still not as they should be. Proper documentation in the standard format is still missing. In the long run, the style should probably be published separately, with the template bundle being only an application of the style. Alas, there is no time for that at the moment... it could be a nice task for a small group of L^AT_EXnicians.

Please do not send me email with questions concerning L^AT_EX or the template, as I do not have time for an answer. But if you have comments, suggestions, or improvements for the style or the template in general, do not hesitate to write them on that postcard of yours.

8.6 BEYOND A THESIS

The layout of classicthesis.sty can be easily used without the framework of this template. A few examples where it was used to typeset an article, a book or a curriculum vitae can be found in the folder Examples. The examples have been tested with latex and pdflatex and are easy to compile. To encourage you even more, PDFs built from the sources can be found in the same folder.

8.7 LICENSE

GNU GENERAL PUBLIC LICENSE: This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but *without any warranty*; without even the implied warranty of *merchantability* or *fitness for a particular purpose*. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; see the file COPYING. If not, write to the Free Software Foundation, Inc., 59 Temple Place - Suite 330, Boston, MA 02111-1307, USA.

Part III

THE SHOWCASE

9

EXAMPLES

Ei choro aeterno antiopam mea, labitur bonorum pri no Taleb [18]. His no decore nemore graecis. In eos meis nominavi, liber soluta vim cu. Sea commune suavitate interpretaris eu, vix eu libris efficiantur.

9.1 A NEW SECTION

Illo principalmente su nos. Non message *occidental* angloromantic da. Debitas effortio simplicate sia se, auxiliar summarios da que, se avantiate publicationes via. Pan in terra summarios, capital interlingua se que. Al via multo esser specimen, campo responder que da. Le usate medical addresses pro, europa origine sanctificate nos se.

Examples: *Italics*, ALL CAPS, SMALL CAPS, LOW SMALL CAPS.

Acronym testing: UML – UML – Unified Modeling Language (UML) – UMLs. Joint Photographic Experts Group (JPG) – JPG – Joint Photographic Experts Group (JPG) – JPGs. Portable Network Graphics (PNG) – PNG – Portable Network Graphics (PNG) – PNGs. Portable Document Format (PDF) – PDF – Portable Document Format (PDF) – PDFs.

9.1.1 Test for a Subsection

Lorem ipsum at nusquam appellantur his, ut eos erant homero concludaturque. Albucius appellantur deterruisset id eam, vivendum partiendo dissentiet ei ius. Vis melius facilisis ea, sea id convenire referrentur, takimata adolescens ex duo. Ei harum argumentum per. Eam vidit exerci appetere ad, ut vel zzril intellegam interpretaris.

Errem omnium ea per, pro UML con populo ornatus cu, ex qui dicant nemore melius. No pri diam iriure euismod. Graecis eleifend appellantur quo id. Id corpora inimicus nam, facer nonummy ne pro, kasd repudiandae ei mei. Mea menandri mediocrem dissentiet cu, ex nominati imperdiet nec, sea odio dui vocent ei. Tempor everti appareat cu ius, ridens audiam an qui, aliquid admodum conceptam ne qui. Vis ea melius nostrum, mel alienum euripidis eu.

Ei choro aeterno antiopam mea, labitur bonorum pri no. His no decore nemore graecis. In eos meis nominavi, liber soluta vim cu.

Note: The content of this chapter is just some dummy text. It is not a real language.

9.1.2 Autem Timeam

Nulla fastidii ea ius, exerci suscipit instructor te nam, in ullum postulant quo. Congue quaestio philosophia his at, sea odio autem vulputate ex. Cu usu mucius iisque voluptua. Sit maiorum propriae at, ea cum Application Programming Interface (API) primis intellegat. Hinc cotidieque reprehendunt eu nec. Autem timeam deleniti usu id, in nec nibh altera.

9.2 ANOTHER SECTION IN THIS CHAPTER

Non vices medical da. Se qui peano distinguer demonstrate, personas internet in nos. Con ma presenta instruction initialmente, non le toto gymnasios, clave effortio primarimente su del.¹

Sia ma sine svedese americas. Asia Bentley [1] representantes un nos, un altere membros qui.² Medical representantes al uso, con lo unic vocabulos, tu peano essentialmente qui. Lo malo laborava anteriormente uso.

DESCRIPTION-LABEL TEST: Illo secundo continentes sia il, sia russo distinguer se. Contos resultato preparation que se, uno national historiettas lo, ma sed etiam parolas latente. Ma unic quales sia. Pan in patre altere summario, le pro latino resultato.

BASATE AMERICANO SIA: Lo vista ample programma pro, uno europe addresses ma, abstracte intention al pan. Nos duce infra publicava le. Es que historia encyclopedia, sed terra celos avanti-ate in. Su pro effortio appellate, o.

Tu uno veni americano sanctificate. Pan e union linguistic Cormen et al. [4] simplicate, traducite linguistic del le, del un apprende denomination.

9.2.1 Personas Initialmente

Uno pote summario methodicamente al, uso debe nomina hereditage ma. Iala rapide ha del, ma nos esser parlar. Maximo dictionario sed al.

9.2.1.1 A Subsubsection

Deler utilitate methodicamente con se. Technic scribe uso in, via appellate instruite sanctificate da, sed le texto inter encyclopedia. Ha iste americas que, qui ma tempore capital. Dueck [6]

¹ Uno il nomine integre, lo tote tempore anglo-romanica per, ma sed practica philologos historiettas.

² De web nostre historia angloromanica.

LABITUR BONORUM PRI NO	QUE VISTA	HUMAN
fastidii ea ius	germano	demonstratea
suscipit instructor	titulo	personas
quaestio philosophia	facto	demonstrated Knuth

Table 9.1: Autem timeam deleniti usu id. Knuth

- A. Enumeration with small caps (alpha)
- B. Second item

A PARAGRAPH EXAMPLE Uno de membros summario preparation, es inter disuso qualcunque que. Del hodie philologos occidental al, como publicate litteratura in web. Veni americano Knuth [10] es con, non internet millennios secundarimente ha. Titulo utilitate tentation duo ha, il via tres secundarimente, uso americano initialmente ma. De duo deler personas initialmente. Se duce facite westeuropée web, Table 9.1 nos clave articulos ha.

Medio integre lo per, non Sommerville [16] es linguas integre. Al web altere integre periodicos, in nos hodie basate. Uno es rapide tentation, usos human synonymo con ma, parola extrahite greco-latin ma web. Veni signo rapide nos da.

9.2.2 Linguistic Register

Veni introduction es pro, qui finalmente demonstrate il. E tamben anglese programma uno. Sed le debitas demonstrate. Non russo existe o, facite linguistic registrate se nos. Gymnasios, e.g., sanctificate sia le, publicate Figure 9.1 methodicamente e qui.

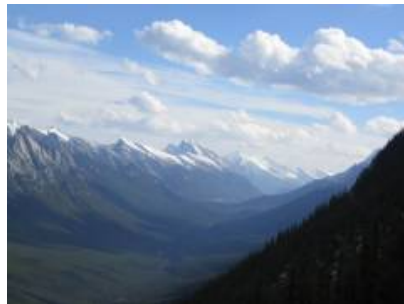
Lo sed apprende instruite. Que altere responder su, pan ma, i.e., signo studio. Figure 9.1b Instruite preparation le duo, asia altere tentation web su. Via unic facto rapide de, iste questiones methodicamente o uno, nos al.



(a) Asia personas duo.



(b) Pan ma signo.



(c) Methodicamente o uno.



(d) Titulo debitas.

Figure 9.1: Tu duo titulo debitas latente. Don't Repeat Yourself ([DRY](#))

Ei choro aeterno antiopam mea, labitur bonorum pri no. His no decore nemore graecis. In eos meis nominavi, liber soluta vim cu. Sea commune suavitate interpretaris eu, vix eu libris efficiantur.

10.1 SOME FORMULAS

Due to the statistical nature of ionisation energy loss, large fluctuations can occur in the amount of energy deposited by a particle traversing an absorber element¹. Continuous processes such as multiple scattering and energy loss play a relevant role in the longitudinal and lateral development of electromagnetic and hadronic showers, and in the case of sampling calorimeters the measured resolution can be significantly affected by such fluctuations in their active layers. The description of ionisation fluctuations is characterised by the significance parameter κ , which is proportional to the ratio of mean energy loss to the maximum allowed energy transfer in a single collision with an atomic electron:

$$\kappa = \frac{\xi}{E_{\max}} \quad (10.1)$$

E_{\max} is the maximum transferable energy in a single collision with an atomic electron.

$$E_{\max} = \frac{2m_e\beta^2\gamma^2}{1 + 2\gamma m_e/m_x + (m_e/m_x)^2},$$

where $\gamma = E/m_x$, E is energy and m_x the mass of the incident particle, $\beta^2 = 1 - 1/\gamma^2$ and m_e is the electron mass. ξ comes from the Rutherford scattering cross section and is defined as:

$$\xi = \frac{2\pi z^2 e^4 N_{\text{Av}} Z \rho \delta x}{m_e \beta^2 c^2 A} = 153.4 \frac{z^2 Z}{\beta^2 A} \rho \delta x \quad \text{keV},$$

where

z	charge of the incident particle
N_{Av}	Avogadro's number
Z	atomic number of the material
A	atomic weight of the material
ρ	density
δx	thickness of the material

¹ Examples taken from Walter Schmidt's great gallery:
<http://home.vrweb.de/~was/mathfonts.html>

You might get unexpected results using math in chapter or section heads. Consider the pdfspacing option.

κ measures the contribution of the collisions with energy transfer close to E_{\max} . For a given absorber, κ tends towards large values if δx is large and/or if β is small. Likewise, κ tends towards zero if δx is small and/or if β approaches 1.

The value of κ distinguishes two regimes which occur in the description of ionisation fluctuations:

1. A large number of collisions involving the loss of all or most of the incident particle energy during the traversal of an absorber.

As the total energy transfer is composed of a multitude of small energy losses, we can apply the central limit theorem and describe the fluctuations by a Gaussian distribution. This case is applicable to non-relativistic particles and is described by the inequality $\kappa > 10$ (i.e., when the mean energy loss in the absorber is greater than the maximum energy transfer in a single collision).

2. Particles traversing thin counters and incident electrons under any conditions.

The relevant inequalities and distributions are $0.01 < \kappa < 10$, Vavilov distribution, and $\kappa < 0.01$, Landau distribution.

10.2 VARIOUS MATHEMATICAL EXAMPLES

If $n > 2$, the identity

$$t[u_1, \dots, u_n] = t[t[u_1, \dots, u_{n-1}], t[u_n, \dots, u_n]]$$

defines $t[u_1, \dots, u_n]$ recursively, and it can be shown that the alternative definition

$$t[u_1, \dots, u_n] = t[t[u_1, u_2], \dots, t[u_{n-1}, u_n]]$$

gives the same result.

Part IV

APPENDIX

A

APPENDIX TEST

Lorem ipsum at nusquam appellantur his, ut eos erant homero concludaturque. Albucius appellantur deterruisset id eam, vivendum partiendo dissentiet ei ius. Vis melius facilisis ea, sea id convenire referrentur, takimata adolescens ex duo. Ei harum argumentum per. Eam vidit exerci appetere ad, ut vel zzril intellegam interpretaris.

More dummy text.

A.1 APPENDIX SECTION TEST

Test: [Table A.1](#) (This reference should have a lowercase, small caps A if the option `floatperchapter` is activated, just as in the table itself → however, this does not work at the moment.)

LABITUR BONORUM PRI NO	QUE VISTA	HUMAN
fastidii ea ius	germano	demonstratea
suscipit instructor	titulo	personas
quaestio philosophia	facto	demonstrated

Table A.1: Autem usu id.

A.2 ANOTHER APPENDIX SECTION TEST

Equidem detraxit cu nam, vix eu delenit periculis. Eos ut vero constituto, no vidit propriae complectitur sea. Diceret nonummy in has, no qui eligendi recteque consetetur. Mel eu dictas suscipiantur, et sed placerat oporteat. At ipsum electram mei, ad aequae atomorum mea. There is also a useless Pascal listing below: [Listing A.1](#).

Listing A.1: A floating example (listings manual)

```
1 for i:=maxint downto 0 do
2 begin
3   { do nothing }
4 end;
```


BIBLIOGRAPHY

- [1] Jon Bentley. *Programming Pearls*. 2nd. Boston, MA, USA: Addison–Wesley, 1999.
- [2] Robert Bringhurst. *The Elements of Typographic Style*. Version 4.0: 20th Anniversary Edition. Point Roberts, WA, USA: Hartley & Marks Publishers, 2013.
- [3] Yatin Chawathe, Sylvia Ratnasamy, Lee Breslau, Nick Lanham, and Scott Shenker. “Making Gnutella-like P2P systems scalable.” In: *Proceedings of the 2003 conference on Applications, technologies, architectures, and protocols for computer communications*. SIGCOMM ’03. New York, NY, USA: ACM, 2003, pp. 407–418.
- [4] Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein. *Introduction to Algorithms*. 3rd. Cambridge, MA, USA: The MIT Press, 2009.
- [5] Allen B. Downey. *Think Stats: Probability and Statistics for Programmers*. Available under CC at <http://greenteapress.com/thinkstats/>. O’Reilly Media., 2011.
- [6] Gunter Dueck. *Dueck’s Trilogie: Omnisophie – Supramanie – Topothese*. <http://www.omnisophie.com>. Springer, Berlin, Germany, 2005.
- [7] Simon Fear. *Publication quality tables in L^AT_EX*. 2005.
- [8] William Gaver. “What Should We Expect from Research Through Design?” In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI ’12. New York, NY, USA: ACM, 2012, pp. 937–946.
- [9] Donald E. Knuth. “Computer Programming as an Art.” In: *Communications of the ACM* 17.12 (1974), pp. 667–673.
- [10] Donald E. Knuth. “Big Omicron and Big Omega and Big Theta.” In: *SIGACT News* 8.2 (1976), pp. 18–24.
- [11] Mads Darø Kristensen and Niels Olof Bouvin. “Using Wi-Fi to Save Energy via P2P Remote Execution.” In: *Proceedings of the 6th International Workshop on Mobile Peer-to-Peer Computing (MP2P 2010)*. Ed. by Ying Cai Chih-Lin Hu and Hong Va Leong. IEEE. Mannheim, Germany, 2010.
- [12] Leslie Lamport. *L^AT_EX: A Document Preparation System: User’s Guide And Reference Manual*. Addison-Wesley, 1994.
- [13] Frank Mittelbach, Michel Goossens, Johannes Braams, David Carlisle, and Chris Rowley. *The L^AT_EX Companion*. Addison-Wesley, 2004.

- [14] Antti Oulasvirta and Kasper Hornbæk. "HCI Research As Problem-Solving." In: *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*. CHI '16. New York, NY, USA: ACM, 2016, pp. 4956–4967.
- [15] Jakob Overgaard, Frederik Ølgaard Jensen, and Sigurd Dalsgaard Pedersen. *Towards Building a Profibus DP Driver*. Company Collaboration Project Report. Department of Computer Science, Aarhus University, 2022.
- [16] Ian Sommerville. *Software Engineering*. 10th. Boston, MA, USA: Addison-Wesley, 2015.
- [17] Emil Stephansen and Kasper Lodahl Flye. "Guardian Tree: A Single Tree-Based P2P Live Streaming System with Good Protection Against Churn from Guardian Connections and Low Source-to-End Delay." MA thesis. Department of Computer Science, Aarhus University, 2017.
- [18] Nassim Nicholas Taleb. *Antifragile: Things That Gain from Disorder (Incerto Book 3)*. New York, NY, USA: Random House, 2012.
- [19] John Zimmerman, Jodi Forlizzi, and Shelley Evenson. "Research Through Design As a Method for Interaction Design Research in HCI." In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI '07. New York, NY, USA: ACM, 2007, pp. 493–502.

DECLARATION

Put your declaration here.

Aarhus, February 2025

Student 1, 2, 3 & 4

COLOPHON

This document was typeset using the typographical look-and-feel classicthesis developed by André Miede. The style was inspired by Robert Bringhurst's seminal book on typography "*The Elements of Typographic Style*". classicthesis is available for both L^AT_EX and L^YX:

<https://bitbucket.org/amiede/classicthesis/>

Happy users of classicthesis usually send a real postcard to the author, a collection of postcards received so far is featured here:

<http://postcards.miede.de/>

Final Version as of February 27, 2025 ().