Template for reports on Milestones

Henry-Norbert Cocos, M.Sc.

Frankfurt University of Applied Sciences
(1971-2014: Fachhochschule Frankfurt am Main)
Nibelungenplatz 1
D-60318 Frankfurt am Main
cocos@fb2.fra-uas.de

Abstract This is where the abstract comes in. The abstract should describe the content of the document in a few sentences (less than 10 lines). Wikipedia writes: An abstract is a concise statement of the contents, an outline without interpretation or evaluation of a scientific paper. The abstract should be as short as possible. All essential facts should be explicitly included. The abstract should arouse the reader's interest in the content of the document.

This is where the introduction (the prologue or foreword) comes in. The introduction should also be short and concise. The reader should be prepared for the text that follows. Of course, the introduction should also be formulated in an interesting way.

1 General information on the document

This document is a template for the written documentation of the Milestones, which must be created as part of the examination of the lecture Cloud Computing in WiSe 2024 at FRA-UAS. It is intended to give beginners in LaTeX [3] a few basics and some hints on layout, language and correct working. Lecture participants can use this template, but they do not have to.

The document uses the macro package `llncs` (Lecture Notes in Computer Science) [1] for LaTeX2e from Springer-Verlag, which is often used for scientific papers in computer science.

A few words about the length: For scientific publications, a maximum length of 10 pages has been established. In general, the length of a paper is never a criterion for quality.

2 Structure of the documentation

After the abstract and the introduction, you describe the problem. Then your own contribution begins and thus the main part of your work. Describe possible ways to solve the problem and develop your solution. This also includes arguing why you have developed your solution in this way and not another. Next, describe your implementation. Go into the important parts of your own work. Longer (more than 1 page) source texts belong in the appendix. Then evaluate the results of your work. Failures and their reasons are also results and you should describe these.

Finally, there is a summary/conclusion/outlook and possibly the appendices.

3 Basics in LaTeX

The following pages provide a few basics on how to use LaTeX. Particularly recommended books are the LaTeX books by Helmut Kopka [4] [5] [6], as well as Frank Mittelbach and Michel Goossens [7]. The books by Joachim Schlosser [8] and Roland Willms [9] are also interesting.

3.1 Comments in source code

Comments always begin with the percent sign `%` This causes LaTeX to ignore all commands, text or other information up to the end of the line. Attention: The percent sign only applies up to the end of the line.

3.2 Written pleadings and textual highlights

There are different fonts: **Bold Face** (`\textbf{...}`), Roman (`\textrm{...}`), *Italic* (`\textit{...}`), `Typewriter` (`\texttt{...}`), Sans Serif (`\textsf{...}`), *Slanted* (`\textsl{...}`) and Small Caps (`\textsc{...}`). However, this should be used very sparingly.

Only really important points should be highlighted. You can use `\emph{}` for this. This command can be used to simply *highlight* text.

Text Underline is also possible, but not recommended. The Underline comes from the time of typewriters. Back then, there was no other way to highlight text. Nowadays, this is unusual in high-quality publications because it simply doesn't look good.

3.3 Unformatted text

The ability of LaTeXto display unformatted text with a fixed character width is particularly helpful when integrating source code.

The `verbatim` environment is available for setting unformatted text. The content of the environment is set by LaTeX in the font `TypeWriter` (typewriter font) and is not interpreted. It can therefore also contain special characters from LaTeXṠpaces and line breaks are simply adopted and printed. No more commands can be executed until `\end{verbatim}` The environments `verbatim` also ensures that the unformatted text is set off. The following lines are an example of

```
This is a test with the environment verbatim.
You can simply enter special characters here.
§ $ % & / | \ ~ * # - -- ---
```

There is a possibility to output text up to one line long, unformatted. This is the command `\verb<character>Text<character>`

The text in question is enclosed by an (almost) arbitrary character. For example, it can be a `!`, `§`, `|` or `+` The character must be specified directly after the `\verb` command. The first occurrence starts the `verbatim` environment and the next occurrence of the character ends the environment. Therefore, the character must not be within the environment (of the text to be displayed).

3.4 Special characters and continuation points

A few special characters: \, $, &, €, %, #, _, ~, ^, |, {, }

Other special characters: ©, ®, ™, §, ¶, £, †, ‡, •

Continuation points are made by the command `\dots` Result: ...

3.5 hyphen, dash, longer dash

LaTeX distinguishes between three different types of dashes in printed text.

There is the *separating stroke*, which is used as a separator when separating words, or as a *binding stroke* in compound terms. It is written like an ordinary dash (`-`).

The second variant is the *thought stroke* –. This is often used to indicate distance or time. It is realized with two dashes (--). There should be spaces before and after this dash.

In the English-speaking world, the *longer dash* —. It is realized with three dashes (---). It is usually connected to the preceding and following text passages without spaces. This stylistic device is rather unusual in German.

3.6 Source code

The lstlisting environment is recommended for setting source code. Beforehand, the author uses the command \lstset to define the type of source code and how it should be formatted. Listing 1.1 contains a shell script with an example of source code. Each source code should have a signature (description). For syntax highlighting to work correctly, the author must specify the programming language with the keyword language

```bash
#!/bin/bash
#
# Script: operands2.bat
#
# Include function libraries
. function.bib

echo "Please type in an operator."
echo "Possible values are: + - * /"
read OPERATOR
echo "Please type in the first operand:"
read OPERAND1
echo "Please type in the second operand:"
read OPERAND2

# Process input
case $OPERATOR in
  +)  add $OPERAND1 $OPERAND2 ;;
  -)  sub $OPERAND1 $OPERAND2 ;;
  \*) mul $OPERAND1 $OPERAND2 ;;
  /)  div $OPERAND1 $OPERAND2 ;;
  *)  echo "Wrong input: $OPERATOR" >&2
      exit 1
      ;;
esac

# Output the result
echo "$OPERAND1 $OPERATOR $OPERAND2 = $RESULT"
```

Listing 1.1. This is the signature of the source code

3.7 Lists and enumerations

- Lists are often very helpful for structuring text.
- Lists can also be used to summarize text.
- Lists are made with the `itemize` environment.
    - Nested listings are also no problem.
        - You can nest up to four levels.
            - But more than two levels almost never looks good.

1. There are of course also enumerations in LaTeX.
2. Enumerations are made with the `enumerate` environment.
    (a) These can also be nested as desired.
3. You can of course also use...
    - lists and
    - Enumerations can be nested as desired.

3.8 Colors in texts

colors should be used here not or only in appropriate (rare!) cases. The reasons are:

- colored texts heavy that reading significant.
- A color printer is not available everywhere.
- Not all color printers produce the same result.

3.9 Citation

Quotations can be helpful. If you copy texts word-for-word, make this clear with indentations and indicate the source. The `quotation` environment is available for quotations. Here is a suitable quote from Donald E. Knuth, the developer of TeX:

> Science is knowledge which we understand so well that we can teach it to a computer; and if we don't fully understand something, it is an art to deal with it. [2]

In scientific theses, however, the need for such citations should be rather low, as the design and subsequent implementation of a solution is usually the core of the work. Even if you use images or tables, you must cite the source in the caption or table heading. It's a shame if you forget to cite the source at the end. It is therefore recommended that you check your own work with tools such as PlagScan[1] to be on the safe side.

[1] http://www.plagscan.com

3.10 Footnotes

Footnotes can be helpful to include additional information in the document. In general, however, footnotes should be used sparingly.

In LaTeX footnotes[2] are created with the command `\footnote{Text of footnote}` You do not need to worry about the numbering or positioning of the footnotes.[3]

3.11 Figures

If possible, insert illustrations as vector graphics. This is particularly easy with self-created diagrams. In contrast to raster graphics, vector graphics can be scaled continuously and without loss of quality. File formats for vector graphics include `eps`, `ps`, `pdf` and `svg` File formats for raster graphics include `bmp`, `gif`, `jpg` and `png`

Each figure requires a caption with a unique number and must be referenced in the text (see figure 1.

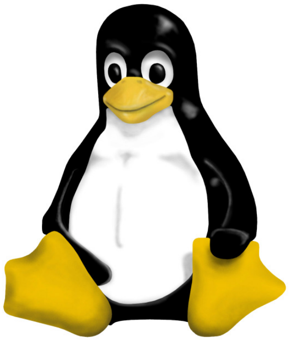

Figure 1. Caption of figure 1

Rotating images is easy with the parameter `angle=<angle>`. Figure 2 has been rotated by 270 degrees.

If an image should be as wide as the text field, this can be easily achieved with `width=\textwidth`

[2] A footnote
[3] Another footnote

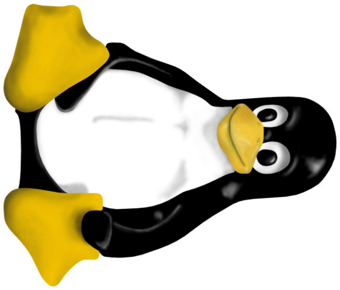

Figure 2. caption of figure 2

3.12 Tables

LaTeX offers many environments for creating tables. A simple environment is `tabular`. Springer recommends writing the table heading above the table and not using vertical separators. Another recommendation from Springer is to only draw horizontal separators to delimit the actual table and the table header. An example of such a table is Table 1.

Each table requires a table heading with a unique number and must be referenced in the text.

Table 1. A simple table

line	**left-aligned**	**centered**	**right-aligned**
1	line 1	line 1	line 1
2	line 2	line 2	line 2
3	line 3	line 3	line 3

Classically, tables look more like Table 2, with vertical and horizontal separators at all field boundaries. The result is of course correct, but does not look as elegant as Table 1.

Table 2. A table in classic layout

line	**Left-aligned**	**Centered**	**Right-aligned**
1	line 1	line 1	line 1
2	line 2	line 2	line 2
3	line 3	line 3	line 3

3.13 Working correctly with numbers and currencies

Avoid evaluations that you cannot substantiate and quantify. Formulations such as „simple“, „difficult“, „slow“, „fast“, „favorable price“ or „high data throughput“ are not clear and have no place in scientific documentation. Ask yourself: How much is a lot? How much money is cheap? How many MB/s is fast? Such formulations always depend on the context and are not unambiguous. If you evaluate something, it must be based on sources or your own measurement results.

In German-language documents, currency symbols are written after the amount (example: 10 €) with a protected space between the currency symbol and the amount. Such a space must not be wrapped. In English-language documents, currency symbols are placed before the amount (example: $10). There is no space between the currency symbol and the amount.

The decimal separator marks the boundary between the integer part and the fractional part of a number. In German-language documents, a comma is used as the decimal separator (decimal point). In English-language documents, a point is used as the decimal separator (decimal point).

For large numbers, it is recommended to use thousands separators for better readability. In German-language documents, a period (example: 10,000 €) or a protected space (example: 10 000 €) is used as a thousands separator. In English-language documents, a comma is used as the thousands separator (example: $10,000).

3.14 Quality of the language

Quality comes from agony and it’s not the reader who should agonize, but the author!

Formulate factually and precisely. Your text should be free of long-winded, prosaic and cumbersome formulations.

Avoid sentences that are too long and run over several lines. The text should be easy to read.

Formulate actively. You can achieve this by avoiding words such as „wird“, „wurde“ and „werden“.

Avoid unnecessary Anglicisms in a German-language thesis. Words such as website, email, server and browser are in the Duden dictionary and are virtually Germanized. There have been established German terms for many other words in IT for decades. It makes sense to use these as well.

Write in the third person. This means you avoid the first person form (e.g. „I/we measured the data throughput with xyz...") and addressing the reader directly (e.g. „in my/our work you see..."). Keep your distance from the reader (e.g. „The measurement of data throughput with xyz resulted in...").

Do not get carried away with advertising language („blah-blah language") by euphorically praising certain products or companies. If you translate and include sections with advertising texts from American product pages, the reader will notice.

With your own work, as an author you become blinded by time and overlook poor wording. When you have finished the content of your work, read it out loud to yourself from start to finish. You will recognize bad formulations and repetitions of words immediately.

Avoid the dork apostrophe[4].

Make sure you use the hyphen correctly. Note: If you combine an English word with a German word, you must use a hyphen. Examples are „cloud service", „grid resource" and „client-server application".

3.15 Layout

A text belongs under each heading. For example, the heading of a section should not follow directly under a chapter heading.

Paragraphs should not be too long, but if possible should not consist of a single sentence.

Make sure the margins are large enough. This makes it easier to proofread your work and enhances its appearance. It is not the aim of your work to squeeze as much text as possible onto one page. LaTeX and the `llncs` template do this automatically.

Use justified text and not flattened text. LaTeX and the `llncs` template do this automatically.

3.16 References

The references (and the reference to them!) are of crucial importance, because they show that the author is well-read and familiar with the subject matter. Referring to recognized sources is a cornerstone of scientific work. The use of BibTeXis recommended for large documents. The literature sources are collected

[4] http://www.deppenapostroph.info

centrally in a `.bib` file, which is imported once in the document. Only the referenced sources appear in the finished document and BibTEXcontrols the layout of the sources.

BibTEX is often not necessary for smaller documents. The `thebibliography` environment is sufficient here. Each new entry begins with the command `\bibitem{mark}` In continuous text, it is possible to refer to this marker via `\ref{marker}` One advantage of `thebibliography` is that it is very easy to use. The disadvantage is that all changes are made manually.

4 Closing words

The conclusion contains a summary of the document. A kind of conclusion. Here, the most important findings and results are summarized again concisely and precisely in a few sentences. Many authors find working on the conclusion and abstract a tedious task. Nevertheless, authors should make a special effort here, as many readers only decide whether they want to read the entire document or will refer to it in the future based on the content and quality of these two parts.

References

1. Information for Authors of Springer Computer Science Proceedings. Springer. 2017 http://www.springer.com/gp/computer-science/lncs/conference-proceedings-guidelines
2. Knuth, Donald E. *Computer Programming as an Art*. Communications of the ACM 17 (12). December 1974. S.667-673 https://dl.acm.org/doi/pdf/10.1145/1283920.1283929
3. LATEX – A document preparation system. http://www.latex-project.org
4. Kopka, Helmut. *LATEX, Band 1: Einführung*. Pearson. 2005
5. Kopka, Helmut. *LATEX, Band 2: Ergänzungen*. Pearson. 2002
6. Kopka, Helmut. *LATEX, Band 3: Erweiterungen*. Pearson. 2002
7. Mittelbach, Frank und Goossens, Michel. *Der LaTeX-Begleiter*. Pearson. 2005
8. Schlosser, Joachim. *Wissenschaftliche Arbeiten schreiben mit LaTeX: Leitfaden für Einsteiger*. Mitp-Verlag. 2008
9. Willms, Roland. *LATEX: Für Schnelleinsteiger*. Franzis. 2006