Hasso Plattner Institute for Digital Engineering

Bachelor's Thesis

English Title

Your Name

Your Email

Supervised by Prof. Dr. Holger Giese
System Analysis and Modeling Group

Potsdam, May 9, 2019

Contents

1 | Latex Crash Course

This is a template for writing a bachelor thesis. Feel free to change the presets to your desire. The template will walk you through the most common tasks. You might not want to change it directly but instead create a copy, so that you always have a backup.

1.1 Tooling

There are lots of ways to use latex. This template can be used with your personal setup so you don't need to do what I am doing.

1.1.1 Onlinetools

The most comfortable and easy way to use latex are online tools like Overleaf. These simplify the process of writing and compiling Latex documents. They have decent git integration and are pretty much just "plug and play". If you've never used Latex before, start with this if you just want to write!
These tools do have a few problems though. First and foremost they only work online. So if you are on the go or want to turn of internet so you don't get distracted while writing, you can't do that! Secondly compiling is slow. Once the document gets a little longer and you click on compile the time you wait can become multiple minutes. The compiling is done server side for overleaf and since it's a free tool they wont have limitless ressources for each and every user.

1.1.2 Offline

There are lots of ways to setup Latex locally on your pc. I'm personally using vscode + Latex Workshop with a simple texlive installation. If you want to use a different editor you can also do that. Christian Zöllner recommends sublime text with Latextools. For distributions I highly recommend that you stick to Tex Live on Linux. If you work on windows you can install something like MikTex instead but I don't guarantee that the template is going to work very well.
With a local installation you avoid the problems mentioned above. You can work offline just fine and performance of compiling will be limited to your machine. But obviously setup will be a little bit more troublesome. Although it went very smoothly for me personally.
Just as a hint, there are also Latex Plugins for Vim and Emacs which the most harcore users will recommend for obvious reasons. So knock yourself out, GL!

1.2 Citing and Referencing

1.2.1 Bibliography

To cite any source in your thesis, you will need to add that source to your .bib file. In this template you can find the .bib file in `/content_extra/literature.bib`. I have added a few sample entries so you can use those to add your own. Or to make the handling of the bibtex file a little easier you can use a reference manager like Mendeley [Mend]. If you prefer something a little bit more oldschool try JabRef.
To then actually add the reference you simply use `\cite{your_reference}`.
For more information on citing checkout: `https://en.wikibooks.org/wiki/LaTeX/Bibliography_Management`

1.2.2 Crossreferencing

To create a crossreference you need to create a label with `\label{your_label}` at the point that you want to crossreference. Afterwards you can use `\autoref{your_label}` to crossreference that point. It looks like this: chapter 1

1.3 Glossary

If you want to use a glossary uncomment the line `\makeglossaries` in the template. Regardless of if you want to have a glossary (you probably wont need it), it is always a good idea to define your acronyms in there. Because these might change later on. You can do that with the command `\newacronym{label}{Acronym}{Full name}` And then use it via: `\gls{label}`. It will look like this on first use: Hasso-Plattner-Institute (HPI). And like this for all uses afterwards: HPI.

For a more information on glossaries checkout: `https://www.overleaf.com/learn/latex/Glossaries`

1.4 Code

For displaying code I recommend the environment `\begin{lstlisting}[language=Python]` as it comes with highlighting and looks decently nice.

```python
def a_very_cool_method(a, b):
    """ yes since python doesn't support multiline lambdas
        and I want to save the result in an extra variable.
        I need a method for this! It's very pythonic! """
    c = a * b
    return c;
```

If you aren't happy with the styling, somebody should make the effort to customize it as suggested here: `https://www.overleaf.com/learn/latex/Code_listing`

1.5 Figures, Graphics and Tables

1.5.1 Tables

Tables are pretty straight forward and look like this:

1st column	2nd column	3rd column
a	b	c

The code for this table looks like this:

```
\begin{tabular}{r|cl}
    1st column & 2nd column & 3rd column\\
    \hline
    a & b & c
\end{tabular}
```

For a very thorough guide on tables check this page from overleaf again: `https://de.overleaf.com/learn/latex/Tables`

1.5.2 Graphics and Figures

Including a graphic is very easily done by using `\includegraphics{path_to_graphic}` Wrapping a graphic in a `begin{figure} \end{figure}` environment allows you to configure the graphic as you wish. For example you can add a caption with `\caption{your_caption}`.

Figure 1.1 – This is the HPI logo yay!

It makes a lot of sense to give your images a `\label` so that you can reference it like this: Figure 1.1.

For a thorough guide on graphics and figures checkout: `https://de.overleaf.com/learn/latex/Inserting_Images`

Bibliography

[Mend] *Download Mendeley Desktop Software*. URL: `https://www.mendeley.com/download-desktop/` (visited on 03/27/2019).