University of Bamberg

Distributed Systems Group

Seminar paper

As part of the Bachelor|Master seminar

Seminar Topic

With the subject:

LaTeX only

Submitted by:

Author

Supervisor: Prof. Dr. Guido Wirtz

Bamberg, Winterterm 2018/2019

Contents

List of Figures

List of Tables

Listings

Abbreviations

DSG Distributed Systems Group

1 Introduction

For writing theses, the Distributed Systems Group provides LaTeX templates. This document illustrates how the specific environments and commands of the template can be used for creating theses. However, this is no general guide on creating documents with LaTeX. As an introductory literature, the book by Kopka [1] is recommended. There are other older books which provide a good introduction as well. A good overview is also provided by the wikibook "LaTeX" [2]. Furthermore, a lot of information can be found online using various search engines.

A LaTeX-document can be created by running `pdflatex seminar` or by using an appropriate IDE, like *TeXnicCenter*[1] or *TeXlipse*[2]. In addition, the tool SumatraPDF[3] can be used which enables a forward and backward search inside the documents.

2 General structure

The central file of the template is `seminar.tex` which needs to be adjusted by the specific author for a specific thesis. In the file, settings for the title page can be changed. Furthermore, the files for the individual sections need to be included, as well as the *.bib*-file containing *bibtex*-entries.

2.1 Creation of a title page

A title page can be created using the command `\maketitle`. Depending on whether it is a seminar paper or a Bachelor/Masterthesis, the command has a different number of parameters. To create a title page for a seminar paper, six parameters need to be passed to `\maketitle`.

```
\maketitle{Seminar topic}{Paper topic}%
{Author}{Supervisor}{Semester}{Bachelor|Master}
```

The individual parameters are self-explanatory. If the seminar paper is written by more than one author, the names should be separated by `,\\`. In the case of three authors, the third parameter would therefore be `Hans Meier,\\ Peter Müller and\\ Hans Müller` as an example. For the semester, an abbreviation should be used, such as `SoSe19`.

To create a title page for a Bachelor/Masterthesis, the command `\maketitle` has to be called with three parameters.

```
\maketitle{Bachelor-/Masterthesis}{Course of studying}{Thesis topic}%
{Author}{Submission date}
```

Also in this case, the parameters are self-explanatory.

---

[1]`http://www.texniccenter.org`
[2]`http://texlipse.sourceforge.net`
[3]`https://www.sumatrapdfreader.org/free-pdf-reader.html`

2.2 Including sections

To keep the work well-arranged, it is reasonable to create a *.tex*-file for each section. To assemble the files to one single document, the sections need to be included using the command `\input` inside the file `seminar.tex`. If, for example, a section should be included which is inside the file `section-1.tex`, then the following command would be used:

```
\input{section-1}
```

2.3 Including the bibliography

The literature sources should be listed in a *.bib*–file. The *bibtex*–file `example.bib` contains some examples of different sources, like books, articles, etc. Each entry needs a unique label, which is formed from the first three letters of the lastname of the author and the last two digits of the year when it was published. If there is more than one author for a source, the label is formed from the beginning letters of the lastnames of the first three authors and the last two digits of the year when it was published. If an author has published more than one work within a year, the following labels should be appended with small letters, starting with *b*.

A *.bib*-file should be included using `\bibliography` within the file `seminar.tex`. The exemplary file `references.bib` was therefore included using:

```
\bibliography{references}
```

3 Including graphics

Generally, graphics are included between paragraphs. Another option is to surround graphics with text. Graphics should be provided as or *pdf* or *eps*-files. This depends on whether *latex* or *pdflatex* is used to create the document. If *latex* is used, only *eps*-files can be included. If the document is created with *pdflatex*, *pdf*-files should be used.

Graphics should be inside the *images*-directory, to easily include them. Newly created graphics should therefore be copied to the *images*-directory.

3.1 Graphics between paragraphs

Graphics can be included between paragraphs using `\asfigure`.

```
\asfigure{picture1}{vs-logo}{DSG-Logo}{5}
```

The first parameter `picture1` serves as a label to reference the graphic. The second parameter is the name of the graphics file without its file ending. As a third parameter, the caption has to be provided and the last parameter defines the width of the graphic in

centimeters. The width has to be provided as a numeric value. If a graphic is wider than the provided width, it is shrinked. Smaller graphics are scaled up. The above command leads to figure 1.
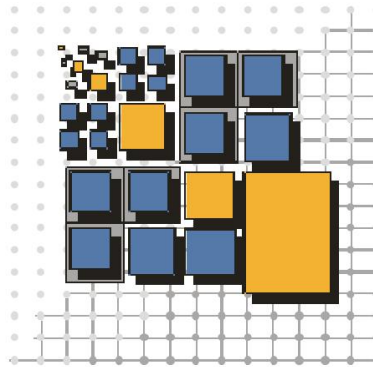

Figure 1: DSG-Logo

3.2   Graphics surrounded by text

If a graphic should be surrounded by text, the command \textflow has to be used. This command is based on the package *wrapfig* which might have to be installed. This package is, similar to all other environments with this functionality, limited and partially erroneous. Therefore it should be used only with caution. A graphic can be included with:


```
\textflow{r}{3}{vs-logo}{DSG-Logo}{picture2}
```
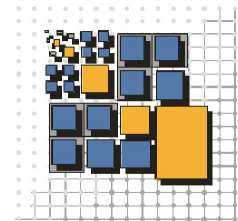
After this command, the text follows which should surround the figure. For the first parameter, only the small letters l and r are possible. They lead to the graphic being arranged on the left or right side. The second parameter can be used to specify the desired width in centimeters. The graphic is scaled accordingly. The third parameter is the name of the graphics file without its file ending. The fourth parameter defines the caption for the figure and with the fifth parameter, a label can be defined to reference the figure.

Figure 2: DSG-Logo

A peculiarity of this package is that graphics are always placed at the beginning of a paragraph. If a graphic should be enclosed completely by a paragraph, as it is the case with figure 2, the command \fliesstext should not be placed before the paragraph but within it. The above command is therefore placed after the word "'installed"'.

4   Including Java source code

To include Java source code, the package *listings* is required, which might have to be installed. This package enables the inclusion of source code from various programming languages into the document. There are two possibilities to include source code in a

document. One possibility is to put the source code directly in the document and the other is to read it from a file.

To include source code directly, a special environment is required which is shown in the following example:

```
\begin{javacode}{This is the first listing}{listing1}
public class Test{

  public Test(){
    // do something
  }

}
\end{javacode}
```

First, an environment called `javacode` is created. Required parameters are a caption and a label. Then follows the actual source code and the environment is closed again. The above example leads to listing 1.

Listing 1: This is the first listing

```
1 public class Test {
2
3   public Test(){
4     // do something
5   }
6
7 }
```

As an alternative, the source code can also be read from a file. This is beneficial, because it increases the readability. To include source code from a file, the command `\javafile` is used. For a successful inclusion, the source code file has to be located in the `src`-directory.

Example usage:

```
\javafile{src.java}{This is the second listing}{listing2}
```

The first parameter has to be the name of the file (with file ending) inside the *src*–directory. The second parameter defines the caption and the third parameter is the label for referencing the listing.

The above example leads to listing 2 as a result.

Listing 2: This is the second listing

```
1 public class Test{
2
3   public Test(){
4     // do something
5   }
6
7 }
```

5   Using the list of abbreviations

To include the list of abbreviations, the package `acronym`[4] is used. Used abbreviations are added to the `abbreviations.tex`–file, which can for example have the following content:

```
\begin{acronym}[LSPI]
 \acro{DSG}{Distributed Systems Group}
 \acro{LSPI}{Lehrstuhl für Praktische Informatik}
\end{acronym}
```

The abbreviation in square brackets `[LSPI]` should be the longest of all used abbreviations. An abbreviation can be created with the command `\acro` inside the `acronym`–environment. As a default, only the abbreviations which are actually used in the text are included in the list of abbreviations. To use an abbreviation inside the text, the following expression is sufficient:

`\ac{`*<acronym>*`}`

When an abbreviation is used for the first time, this expression leads to the abbreviation being written out followed by the abbreviation in brackets. Afterwards, only the abbreviation is printed. The command `\ac{DSG}` therefore creates the written out version when it is used for the first time: ”‘Distributed Systems Group (DSG)”’. And for every following usage, only the abbreviation is printed: ”‘DSG”’.

---

[4]`http://www.ctan.org/tex-archive/macros/latex/contrib/acronym`

References

[1] H. Kopka and P. W. Daly, *Guide to LaTeX*, 4th ed. Addison-Wesley Professional, 2003.

[2] "LaTeX - Wikibooks," http://en.wikibooks.org/wiki/LaTeX, 2012.