Typesetting Your Outline, Annotated Bibliography, and Full Paper for the Course Student Conference in Computing Science*

Frank Drewes

Department of Computing Science
Umeå University, Sweden
drewes@cs.umu.se

1 Introduction

An outline is a draft of a research paper. It will help you to organize your thoughts, ideas and materials in a logical form. It should be detailed enough to make the following clear to yourself and to the reader.

Purpose What exactly is (will be) the goal of your paper. You should therefore develop some kind of *thesis statement*, i.e., a specific statement, research question or hypothesis you want to answer or investigate in more detail.

Rationale/Motivation Make clear why your thesis statement is interesting or important. A general motivation like "I want to learn more about topic X" definitely is not good enough.

Contents What your final paper approximately will contain.

Resources The resources you intend to use in your work. This will be specified in more detail in the annotated bibliography (reference section).

The main sections and subsections should be drafted and contain some actual text or information about the planned contents, like the following.

- Meaningful section and subsection headers.
- Some (preliminary) text in all or most of the sections and subsections that help you organize your thoughts, ideas and materials. This will also give the reader (and yourself) a feeling for the planned contents.
- Citations of literature you intend to use to support your reasoning, like [1] or [4, 8, 10].
- Drafts of figures (see Figure 1 and Figure 2), tables (see Table 1), or examples.

2 Annotated Bibliography

The annotated bibliography is a list of literature references, where each reference is followed by an "annotation". Please do not confuse this annotation with an

* based on an earlier version by Jürgen Börstler

abstract. An annotation should describe and comment the reference *in your own words*, and it should focus on the aspects that are *important to you*.

The actual data of the annotated bibliography is stored in a separate text file, a BIBTeX database such as the file `demo.bib`. Each entry describes a reference and consists of a number of fields recording title, author, etc. Annotations and other information can be stored in additional fields. Please note that you do not need to delete any annotations from your BIBTeX file for later deliverables. The annotations will be disregarded by BIBTeX because you will use a different BIBTeX formatting style for the references in Deliverables 3 and 4a/b (see the commented section at the end of `demo.bib`).

3 Formatting Guidelines

Please check the lecture overheads and the submission guidelines for details on formatting and other requirements. The outline should consist of about two pages plus the annotated bibliography.

4 Subsections, Lists, Tables, and Figures

Sections and subsections on the first and second levels are numbered automatically. Subsections on levels three or more do not get section numbers in the formatting style we use here.

4.1 Lists

Lists can be itemized, like in the list right before Section 2, or enumerated as below. Of course, lists can be nested.

1. This is an example of a nested enumerated (numbered) list.
   (a) First subitem.
   (b) Second subitem.
2. Second item.
3. And so on.

4.2 Tables

Tables are constructed using the `tabular` environment. For an example, please check the LaTeX source code for Table 1. There are several useful extensions of the `tabular` environment. One of them is `tabularx`, which lets you explicitly define the table width (normally `\linewidth`) and has a column type `X` whose width is automatically calculated from this. If you want to use it, load the package `tabularx` and have a look at its documentation.

left aligned	formatted as a paragraph	center aligned	right aligned
row 2	it is possible to have long texts in tables if they are formatted as a paragraph	this does not work well otherwise	123
row 3	45	6	7

Table 1. A simple table with 4 columns and 3 rows.

4.3 Floating Tables and Figures

Tables and figures should normally be put into the environments `table` and `figure`, respectively. In this way, LaTeX will automatically attempt to move the table or figure to a place where it fits. Moreover, you can (and should) give an informative caption and define a symbolic label used to refer to them from the text. The labelling mechanism works as follows. In or after the caption of the float, you place a the command `\label{my label}`. Now LaTeX will produce the numer of that float in all places where you use the command `\ref{my label}`. You can also refer to the page number on which the float appears by using the command `\pageref{my label}`. For example, in the caption of Table 1 the command `\label{tab:table}` was used, so that this sentence could refer to it by `Table~\ref{tab:table}`. Note that a tilde is used instead of a space character between the word Table and the referencing command. This creates a nonbreaking space, which prevents LaTeX from inserting a line break between the two. The probably obvious purpose is to avoid ugly linebreaks such as Table 1.[1]

The contents of a "float" can be any LaTeX material, but tables often consist of `tabular` material and figures often contain graphics imported via the command `\includegraphics`.

In plain LaTeX (*latex*), imported graphics must be in *PS/EPS* format. If you use *pdflatex*, which we strongly recommend, you can include a range of different formats, like for example *PDF* (see Figure 1) or *JPG* (see Figure 2).

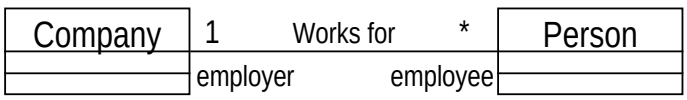

Fig. 1. A figure from a PDF file (must be PDF 1.4 compatible).

As mentioned, LaTeX automatically moves floats to a "good" place, normally near the place where the environment is found in the source file. Thus, as a

[1] The same kind of nonbreaking space should be used in front of `\pageref` and `\cite`, for the same reasons.

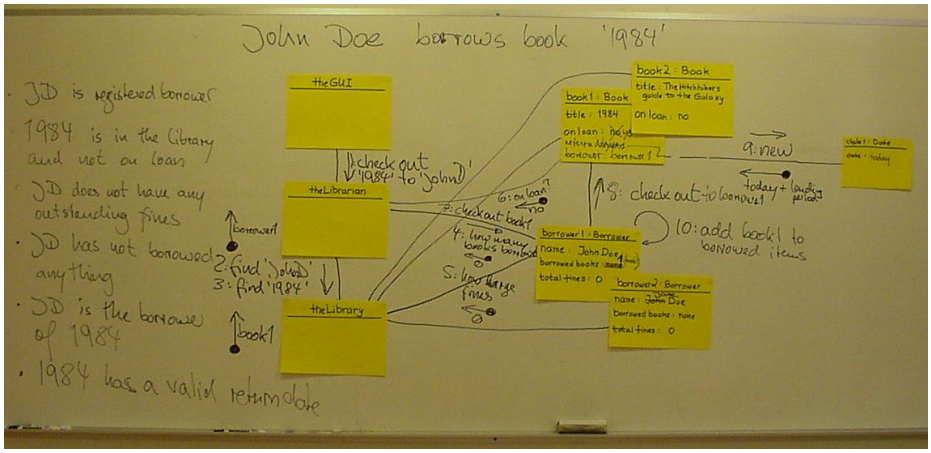

Fig. 2. Another figure; this time from a JPG file.

rule of thumb, the environment should be placed near the position where you refer to the table or figure for the first time, even though it could in principle be added anywhere. Many authors have quite distinct ideas about the optimal placement of a float in the compiled paper. In fact, this placement can be affected by optional arguments to the floating environments. *Please do not make use of these possibilities!* One reason is that the automatic placement defined by the style (in this case the LNCS style) is intended to create a uniform "look and feel" which is destroyed by manual formatting. The other, more important reason is that including the paper in the proceedings may lead to slight changes in, e.g., page breaks. In such cases, manually placed tables and figures can end up in very unsatisfactory places.

5 Further Remarks

Below, some more hints are collected that you should use when preparing your full paper and the final version. In fact, you should follows these recommendations from the very beginning, because it saves time to do it right from the start.

5.1 More about Manual Formatting

In principle, LaTeX gives you
all possibilities in the world
for manual typesetting
in all kinds of weird ways.[2] Please refrain from using them, including manual linebreaks (except in places where these are intended to be used, e.g.,

[2] In fact, this statement is literally true to the extent possible, because LaTeX is Turing complete. Thus, every computable layout can be implemented in LaTeX.

to end a row in a `tabular` environment), `\noindent`, `\bigskip`, and the like, the reasons basically being the same as those for avoiding manual placement of floats. Of course, you should even more strictly refrain from changing general layout parameters such as `\parskip` and `\parindent`.

5.2 Referring to Internet Resources

People use different ways to refer to homepages, forums, and other types of Internet resources. Some treat them as ordinary references and include them in their reference section, such as [3]. However, since such material is quite different in character from real scientific publications, a cleaner solution is to use the reference section only for ordinary references and add footnotes to point to other material, e.g., when writing about BlueJ[3]. Note that, nowadays, even scientific publications sometimes lack a printed version and are available only electronically (such as, e.g., [7]). Of course, such publications should nevertheless be included in the list of references. The important point is not whether material is printed or not, but which role it plays.

6 A Checklist for the Full Version

When your full version is complete and ready for submission, please spend a few minutes to check the following, using the search facility of your favorite editor:

- You have taken care of the adjustments necessary to be made when switching from the outline and annotated bibliography to the actual style of the article. In particular, this means that you do not use the `oribibl` option in the `\documentclass` command, the BIBTeX style used is `splncs`, and the command `\nocite{*}` has been removed or commented out.
- Words in the title of the paper and in section and subsection headings are correctly capitalized.
- Spaces in front of `\ref`, `\pageref`, and `\cite` are nonbreaking. (Also, make sure that there actually *are* spaces in front of them!) Again, note that there might be small changes in the layout when the proceedings are produced. Therefore, it is not guaranteed that all line breaks will stay the same.
- Check the list of references carefully. If something is strange, correct the corresponding record in your BIBTeX file. Note that names of authors and editors in the `author` and `editor` fields should preferably be formatted according to `last name(s), first name(s)`, and multiple authors/editors must be separated by the keyword `and`. In titles, abbreviations such as UML should be enclosed in curly brackets (i.e., `{UML}`) to prevent BIBTeX from turning them into lowercase.
- Check whether you use ordinary double quotation marks `"` in your LaTeX source. If so, replace each by two single quotation marks: ` `` ` on the left and `''` on the right, thus turning, e.g., "fishy" into “fishy”.
- Typeset URLs by means of the command `\url`.

[3] `http://www.buej.org`, BlueJ tool homepage, accessed 2005-01-12

References

[1] Kent Beck and Ward Cunningham. A laboratory for teaching object-oriented thinking. In *Proceedings OOPSLA'89*, pages 1–6, 1989.

The original (classical) paper on CRC-cards. A must read for everyone interested in CRC-cards, although the paper is quite superficial in many places. The authors highlight that they used the approach with experienced programmers. Their experience might therefore not easily transfer to novices.

[2] Joseph Bergin. The object game—an exercise for studying objects. `Http://csis.pace.edu/~bergin/patterns/objectgame.html`, accessed 2008-01-22, 2006.

[3] BlueJ. Tool homepage, 2005. `Http://www.buej.org`, accessed 2005-01-12.

[4] Grady Booch, James Rumbaugh, and Ivar Jacobson. *The Unified Modeling Language User Guide*. Addison-Wesley, Reading, MA, 1999.

The definite source for UML by the original developers. Somewhat out of date, since it refers to UML 1.x. There is a second edition covering UML2.0.

[5] Jürgen Börstler. Object-oriented analysis and design through scenario role-play. Technical Report UMINF-04.04, Dept. of Computing Science, Umeå University, Umeå, Sweden, 2004.

Describes the usage of CRC cards (see [1]) for teaching novices in great detail. Introduces a new kind of diagram (roleplay diagrams) to track and document scenarios. Comes with a detailed example.

[6] CACM Forum. 'Hello, World' gets mixed greetings. *Communications of the ACM*, 45(2):11–15, 2002.

[7] Andrea Corradini and Frank Drewes. Term graph rewriting and parallel term rewriting. In *Proc. 6th Intl. Workshop on Computing with Terms and Graphs (TERMGRAPH 2011)*, Electronic Proceedings in Theoretical Computer Science, pages 3–18, 2011.

[8] Ann E. Fleury. Programming in Java: Student-constructed rules. In *Proceedings SIGCSE'00*, pages 197–201, 2000.

Describes the rules that students constructed to explain to themselves how certain things work. These rules do often not explain things correctly and/or completely.

This can lead to *misconceptions* (i.e., erroneous "understandings" of concepts) that will likely lead to misunderstandings and errors. Such misconceptions are fully consistent with a constructivistic view of learning. The students learn by integrating new things with already known things, based on their current (incomplete) knowledge about the subject and related topics.

[9] R. Granerud, J. Kaasbøll, R. Borge, C. Holmboe, and O. Smørdal. Children's understanding of object-orientation. In Annita Fjuk, Amela Karahasanovic, and Jens Kaasbøll, editors, *Comprehensive Object-Oriented Learning: The Learner's Perspective*, pages 27–47. Informing Science Press, Santa Rosa, CA, USA, 2006.

[10] H. Thimbleby. A critique of Java. *Software—Practice and Experience*, 29(5):457–478, 1999.

A very interesting paper discussing a range of problems in Java. In particular the discussion about Strings is very interesting. The author makes a convincing case that Strings are no "proper" objects, since they cannot be changed. The ability of an object to change its (internal) state depending

on the messages it receives is otherwise considered as an inherent property of objects in the object-oriented paradigm.