

Scientific Writing

“Hints and Tips for writing Scientific Documents”

Des Traynor 09/02/05
Minds Society

A dark blue rounded square containing the word "minds" in a white, lowercase, serif font.

minds

<http://www.minds.may.ie/>

Talk Overview

- Preparing to write up.
- Outlining your final year thesis.
- Writing and re-writing.
- Stuck?
- Final Thoughts

Preparing for the write-up

- Its going to be difficult, if you find it easy you're doing something wrong.
- Allow 3 weeks, allow for things that will go wrong. Don't throw days away.
- Last minute heroes are rare, typically last minute submissions are littered with typos, grammar errors, and incomplete/missing sections.

Personal Preparation

- When you pick a day and start you must put other things on hold. e.g. dossing, telly, lie ins etc.
- If you get hangovers stop drinking. On a 21 day schedule, spending a day hugging the toilet is not advised.
- Stop development, ultimately your thesis is more important than your code. People **will read all** of your thesis, don't make them suffer.

Getting a structure for the thesis

- This is the most important stage. It helps create the “bones” of your thesis, and if you do this stage well enough you will simply be filling in the blanks.
- Divide and Conquer approach
- Typically your thesis will have around 6 chapters, usually along the following lines.

Chapter 1: Introduction

- What problem are you solving?
- Why?
- What other solutions have been used before?
- What makes yours any different?
- Overview of the thesis.

Chapter 2: Requirements

- What did you have to build/research etc?
- What features were required?
- What attributes were required? (portability, maintainability)
- Requirements analysis: What do these restrictions mean? How do they affect the project?

Chapter 3: Design

- How did you design your software?
- Did you use any well known paradigms e.g Model View Controller (for GUIs)
- Design is essentially "*How did you go about solving this problem?*"
- You should not mention specifics (e.g. Java, Perl, Apache) They are implementation issues.

Chapter 4: Implementation

- What tools could you choose for the job?
- Which ones did you choose?
- Why?
- Did you control development in a structured manner? Did you use a versioning system? What issues arose during development?
- How did you build your system?

Chapter 5: Testing

- Usually your implementation choice affects your testing methodology. Did you black-box or white-box or both (make sure understand terms before using them)
- Test cases: What tests did you *actually* perform?
- Did it work? What changes did you make as a result? Are these results repeatable?
- You will be demoing your software, lying is not the best idea here.

Chapter 6: Conclusions

- Link this chapter with chapter 1 & 2. You were given a problem, with requirements. You solved it.
- This is usually the pat-yourself-on-the-back chapter.
- Future Work: You should all have some. What more could be done, if you don't know ask someone, there is always extra things that you can do.
- There is always room for improvement.

Some thoughts

- Whilst that chapter layout is typical enough, you may wish to use more chapters to further divide your thesis.
- For example if implementation required solving a (non trivial) problem then, you may wish to give that area a stand alone chapter. If you did something “good”, definitely separate it, so it can be re-used.
- Thats all I will be saying about chapters/structure etc.

Scientific Writing

- Most of you probably don't do much of it. It is a slightly different language to normal English.
- Everything in your thesis will be either
 - **true.** (e.g. *The system **is** written in PHP...*)
 - **A logical conclusion** (e.g. *"Writing systems in C reduces their cross platform capabilities, therefore Java was a more suitable language in this case"*)
 - **Cited** (e.g. *"PHP has been criticized as a poor language for large scale web development [Crane,02]"*)

Do or do not, there is no try.

- As you can now see, there is no room for ambiguity, or weak sentences.
- Do not use words like should/might/maybe. For the most part, things are or aren't in science, when you say "it could be because", it reads as "I don't know why"
- Do not use colloquial or "relative" language
 - "Using PHP was less hassle because I know it"
 - It was easier to re use the original configuration. (easier than what?)
 - It took long enough to compile. (long with regard to what?)

Walking the tightrope

- You must be precise, while still maintaining readability. This is often difficult.
- If you are struggling, read a (good) thesis from previous years, or read a (good) paper that is in the area.

Avoid Rambling

- No one likes reading BS, you aren't impressing anyone with it.
- Example
 - “The evidence collected thus far in the project clearly states ...” should be **replaced** with “Our data shows...”
 - “It would seem to be clear that ...” should be replaced with “Apparently...”
- There are more (and better) examples on your handout.

Thats all for Scientific Writing

- Read these three links thoroughly, they are excellent.
 - 1) <http://www.ag.iastate.edu/aginfo/checklist.html>
Word Usage
 - 2) <http://www.sportsci.org/jour/9901/wghstyle.html>
Grammar and Punctuation (you *will* need this)
 - 3) <http://tinyurl.com/4c55j>
Excellent all round guide.

Stuck?

- You will all get stuck at some point, similar to writers block.
- It inevitably happens when you are in a bad humor.
- First step is to lighten up, things could be worse.
- On a 3 week schedule, you can't afford to lose days over silly tantrums.

Three suggestions

1. Put your last effort away, and forget it. Go for a coffee/smoke/walk/ or play cards. Do something to forget it. Then re-write it from a different perspective. Finally combine the two, and you'll have a decent chapter.
2. Throw it away (somewhere safe), find someone and rant at them about what you are trying to say. Try to remember what you say. Often its the process of "scientizing" that is causing you the difficulty.
3. If you can't even start, write the first paragraph in a joke stupid fashion, accept that you are throwing it away. Slowly you will get into the style you need. Then re-write the first.

The end

- That's all I know.
- Questions (now or after, or dtraynor@cs.may.ie)