2011-12

Recommended outline for M.S. theses

Rowe, Neil

Monterey, California. Naval Postgraduate School

Recommended outline for M.S. theses

Prof. Neil Rowe, Version 12/11

Abstract: a summary of the main points of the thesis. You should try to have a sentence corresponding to ach chapter of the thesis except the first and last. Mention your most important accomplishments, and minimize background information.

Chapter 1: Introduction (2 to 6 pages)

- Describe the need (problem) your thesis tries to address. Convince the reader that it is a serious need.
- Explain why a computer program or methodology can help this need.
- Explain why previous computer programs or methodologies (if any) haven't solved this need, or else haven't solved it completely.
- Explain what new ideas you had for solving this need. You only need describe these ideas in general terms here.
- Briefly describe the remaining chapters of the thesis.

Chapter 2: Previous attempts to solve this problem, and other problems like it, with computer programs or methodologies (4 to 12 pages)

- Describe the programs or methodologies most similar to yours, with formal citations. Don't give all their details, but point out their key features, especially things that are significantly different in your program or methodology.
- Describe any mathematics necessary for your program or methodology that was not done by you.
- Be sure to have references of each of two types: attempts to solve similar problems to those you are trying to solve, and attempts to use similar solution methods or tools to those that you are trying to use for a different problem.

Chapter 3: Description of your application (2 to 10 pages) (may be interchanged with Chapter 2)

- Describe the problems in the real world your program or methodology will try to address. Describe the setting of those problems, and note things that aren't done as well as they could be, either because they aren't using computers and software or they aren't using the right things. Focus on the problems, not the computer or software.
- Describe the specific assumptions you made in writing your program or formulating your methodology, as for instance the situations the program tries to model, the sort of people that will use the program, applicable orders and regulations that define the task environment, and so on.

Chapter 4: Description of your program or methodology (8 pages minimum)

- If you have a program, say what computer your program runs on, what programming language it is written in, and what special facilities it requires.
- Describe the input and output of your program or methodology in general terms.
- Give a block diagram of the main parts for your program or methodology, and describe the main interactions between the parts. Try to avoid procedure names, variable names, etc.; explain in a high-level way.
- Mention any programs or subroutines or submethodologies that you use that were written by other people, and exactly where they go in the block diagram.
- Describe the data structures used in your program.
- Describe the components of your program or methodology, each in a separate section (this may take many pages if it is complicated).
- Describe the error checking and other user-friendliness features of your program or methodology.
- If you experimented with more than one algorithm, program, or methodology, do the same thing above for the others.

Chapter 5: Discussion of results (2 to 8 pages, not counting tables and figures)

- Summarize the performance of your program or methodology in whatever way is appropriate. Be sure to say how much space it took, how long it took to run (usually in CPU time) or do, and how accurate its results were. Use tables and figures wherever you can to simplify the discussion. (Put actual computer runs in appendices.)

Chapter 6: Conclusion (1 to 8 pages)

- State the major achievements of your program or methodology. Try to compare your program to previous similar programs. Don't just repeat statements in Chapter 1, but put things in a new perspective.
- Honestly admit the major weaknesses of your program or methodology, and how they might be fixed by future students or researchers.

Appendix A: Test runs if interesting to the reader (if there are several, make this several different appendices)

Appendix B: Text of programs if interesting to the reader (should not exceed 50 pages)

****

Some general guidelines for thesis writing

Thesis writing is a form of technical writing. Technical writing tries to convey difficult concepts and so it needs to be especially clear compared to other kinds of writing. That means sentences should be kept short and should use the minimum number possible of specialized words.

1. Always use the same term for the same thing; don't try to be poetic. You want to make things easy as possible for your readers.

2. Use your own words to describe things; use quotations from other people sparingly. If you do use quotations, enclose them in quotations marks and give a formal citation immediately afterwards. This is important because universities treat plagiarism very seriously.

3. The purpose of a thesis is to describe interesting things you did. So don't spend a lot of space on other peoples' work unless it's unique or you need it to make a point about it. So don't describe a well-known algorithm in detail just because you use it in your program.

4. Use "we", "us", "our", and "ours" to describe what you did. And use active voice for verbs, not passive. Some examples:

- Replace "Results in path planning were investigated" by "We investigated path planning".
- Replace "the code that was developed" by "our code".
- Replace "Some examples are shown in Figure 3-1" by "Figure 3-1 shows some examples".

Be aware that some conferences and journals disagree with this guideline and want you to use third person to sound more scientific. But in a thesis it is more important to distinguish what you did and what someone else did.

5. Be careful to use hyphens properly to group multiple modifiers of the same noun. This is particularly important when you use words as modifiers that are normally used as nouns. For instance, in "large artificial-intelligence program" you should hyphenate the two middle words since "intelligence" is a noun and the reader who is not familiar with your topic might think there is something called an "intelligence program" of which there are particular subtypes that are "large" and "artificial". Some more examples of correct hyphenation:

- "intelligent computer-aided instruction"
- "ten-procedure fire-fighting program"
- "problem-dependent forward-chaining Prolog code"
- "large integrated nonbacktracking Java program"

Notice you shouldn't hyphenate an expression unless the whole cluster of words functions as an adjective; for instance, you should write "I work on applied artificial intelligence" but also "I work on applied artificial-intelligence programs".

6. Avoid one-sentence paragraphs except for numbered items of a list.

7. Minimize mention of names of specific files or exact names of procedures. These things are usually too hard for the reader to

remember.

8. Avoid acronyms as much as possible, e.g. write "NPS" as "Naval Postgraduate School" or by its principal noun as "School". Acronyms are used by bureaucracies to make it harder for readers outside a small group to understand what is described, but your goal in a thesis is to communicate. If you must use one, write out its words in parentheses the first time you use the acronym.

9. "Internet" and "Web" (when referring to the World Wide Web) should be capitalized since they are "proper nouns", nouns referring to only one thing in the world. On the other hand, "intrusion-detection system" should not be capitalized since it refers to many things in the world, despite frequently being seens as a capitalized acronym "IDS" (which is another reason to avoid acronyms).

10. Standard citation formats in computer science are either in the form of parethesized author-year pairs (e.g. "(Rowe, 2008)") or numbers in brackets (e.g. "[7]") where the number is that of an item in your reference list. Do not use footnotes for references. In both cases you should have a reference list at the back with references in alphabetical order of the first author's last name. A citation should be given in the first sentence making a reference.

11. Do not repeat something you said earlier unless you have something new to say; cross-reference instead. Technical papers are not usually read in order.

12. In your reference list, conference papers are more desirable than Web pages. Journal papers, book chapters, and books (if not too general) are more desirable than conference papers. But usually you need to include a few conference papers to cover recent work on your topic.

13. Some common phrases to avoid: "utilize" (use "use" instead), "in order to", (use "to" instead), "one of the" (use "one" instead), "some of the" (use "some" instead), "in fact" (delete), "actually" (delete), "a kind of" (use "a" instead), "a type of" (use "a" instead).