A Senior Thesis Proposal Template for CS Students at Allegheny

Stu Dent
Possible thesis advisor: I. M. Firstreader

March 9, 2011

Abstract

A LaTeX template for Senior Project proposals would be of benefit to students who find the organizational aspects of such a project difficult or unclear. This paper proposes itself as just such a template. Using this template as a reference guide, students should be able to understand the form and function of the different parts of a formal thesis proposal; they may, of course, choose to deviate from it as necessary or desirable in the context of their own project proposals. Future work will provide another template for the formal written thesis itself, complete with a `gatorthesis.sty` style-file.

1 Introduction

Creation of a formal research proposal is a daunting task; while authors such as Zobel [**?**] discuss the basic mechanics of technical writing, and writers such as Griffith et al. [**?**] explain the basics of using LaTeX for such writing, there is little guidance on the actual structure of a research proposal. This structure may vary considerably depending on the nature of the research topic—whether it consists of a coding project, a theoretical study, a literature survey or other acceptable topics. Despite their differences, all proposals share common features, and this paper provides a LaTeX template suitable for most such projects. It is expected that a user will modify this template according to the specifics of the proposed research; in particular, boiler-plate section headings and subheadings should be replaced by informative ones specific to the topic.

1.1 Components of a Proposal

The introduction of a proposal typically acquaints the reader with the nature of problem being addressed and the basic idea of the project being proposed to address it. It should

hit many of the points of the abstract while giving more of the motivation and need for the work being proposed. It should then lay out the structure of the paper, telling the reader what will be found in the sections which follow. Thus, the next sections should address the following topics:

- the thesis statement and research objectives;
- prior work and current art—other approaches to the problem and their drawbacks;
- methodology and experiment—including background research to be done, necessary equipment and algorithmic techniques, and general plan-of-attack;
- analysis and validation—how results will be judged and interpreted;
- a time-table for achieving key objectives, and one or more fallback plans in cases where goals are not met by specific target dates;
- concluding remarks.

The proposal should also include a bibiography/reading-list. In this case, it is permissible to include references which are not directly cited in the text. (You can use the `\nocite{*}` command before the bibliography section, as below, to include all BibTEX references from a database file in the bibliography.) You should still cite references in the text as appropriate.

2 Prior Work

Depending on the nature of the topic, prior work should either precede or follow the thesis statement. Some topics will require the background information to put the thesis proposal in context. Others are best served by giving the thesis statement first and then contrasting and comparing it with prior work in the field.

2.1 TEX

If this were a real proposal dealing with how to write proposals, it might be appropriate to say a few words about Knuth, inventor of the TEX typesetting language [**?**]. Description of Knuth's contribution goes here.

2.2 LaTeX (Lamport, 1984)

Description of Lamport's contribution [**?**] goes here.

2.3 Other Work

Description of other contributions leading up to this thesis goes here, e.g., Zobel's book on writing for computer science [**?**].

3 Thesis

It is difficult for students to begin a proposal without an adequate guide; my project, therefore, will demonstrate the following thesis:

> It is both simple and useful to provide students with a LaTeX template for their formal thesis proposals. This will result in far fewer questions from students uncertain as to what belongs in their proposals. Such a template will be of benefit to students in Computer Science/Applied Computing as well as to any others willing and able to use LaTeX for their work.

A formal thesis statement should be a *falsifiable* statement about the goal you will attempt to achieve with your research project. For a purely scientific project, this is the hypothesis you are testing with your research. For an applied programming project, it is usually a statement about the feasibility and correctness of your approach and the advantages it has over other approaches. For a survey or study, it is usually a statement regarding the need or usefulness of such a study, its intended audience, and so on.

Often, you may want to include an itemized list of goals you plan to achieve. Thus, this paper has the goals of

- helping students, by ridding their minds of confusion;
- helping faculty, too, by eliminating their need to say "No, no, no! You need to include a section on *bleh* and *blah*! I told *everybody* that. I thought I told everybody that. Arrgghh!!!"

Figure 1: Include images in a proposal as appropriate

In summary, this section gives the main idea of what you propose, with the goals and contributions it will make to the field; the details of proposed implementation and methodology will be given later in Sec. **??**.

If you didn't already discuss prior work, this is a good place to do so.

4 Implementation and Methodology

Here, you should lay out the details of how you propose to solve the problem and otherwise conduct the research necessary to support your thesis. Include details regarding hardware and software you will use, resources you will draw on, algorithms you will implement, and other ideas about how you will accomplish your task. It is inevitable that your final work will deviate from earlier plans, as you research your topic, learn new methods, and discover what works as expected.

4.1 Using Tables and Figures

Use tables and figures as appropriate; a picture can explain a lot in very compact form, and can keep readers interested. Avoid things that are merely flash and do not add any relevant information. (See Fig. **??**.)

```
#define D define
#D Y return
#D R for
#D e while
#D I printf
#D l int
#D W if
#D C y=v+111;H(x,v)*y++= *x
#D H(a,b)R(a=b+11;a<b+89;a++)
#D s(a)t=scanf("%d",&a)
#D U Z I
#D Z I("123\
45678\n");H(x,V){putchar(".XO"[*x]);W((x-V)%10==8){x+=2;I("%d\n",(x-V)/10-1);}}
l V[1600],u,r[]={-1,-11,-10,-9,1,11,10,9},h[]={11,18,81,88},ih[]={22,27,72,77},
bz,lv=60,*x,*y,m,t;S(d,v,f,_,a,b)l*v;{l c=0,*n=v+100,j=d<u-1?a:-9000,w,z,i,g,q=
3-f;W(d>u){R(w=i=0;i<4;i++)w+=(m=v[h[i]])==f?300:m==q?-300:(t=v[ih[i]])==f?-50:
t==q?50:0;Y w;}H(z,0){W(E(v,z,f,100)){c++;w= -S(d+1,n,q,0,-b,-j);W(w>j){g=bz=z;
j=w;W(w>=b||w>=8003)Y w;}}}W(!c){g=0;W(_){H(x,v)c+= *x==f?1:*x==3-f?-1:0;Y c>0?
8000+c:c-8000;}C;j= -S(d+1,n,q,1,-b,-j);}bz=g;Y d>=u-1?j+(c<<3):j;}main(){R(;t<
1600;t+=100)R(m=0;m<100;m++)V[t+m]=m<11||m>88||(m+1)%10<2?3:0;I("Level:");V[44]
=V[55]=1;V[45]=V[54]=2;s(u);e(lv>0){Z do{I("You:");s(m);}e(!E(V,m,2,0)&&m!=99);
W(m!=99)lv--;W(lv<15&&u<10)u+=2;U("Wait\n");I("Value:%d\n",S(0,V,1,0,-9000,9000
));I("move: %d\n",(lv-=E(V,bz,1,0),bz));}}E(v,z,f,o)l*v;{l*j,q=3-f,g=0,i,w,*k=v
+z;W(*k==0)R(i=7;i>=0;i--){j=k+(w=r[i]);e(*j==q)j+=w;W(*j==f&&j-w!=k){W(!g){g=1
;C;}e(j!=k)*((j-=w)+o)=f;}}Y g;}
```

Figure 2: This mystery code (©1987 Roemer B. Lievaart) was included from a source file. The LaTeX file also shows how to change the font-size for a code-listing.

4.2 Working with Code and Pseudocode

If you have source code to include, you can do so using the *listings* package, which will format short inline code fragments such as

```
for (int i = 0; i < n; i++)
    cout << "It's easy to add source code to LaTeX documents";
```

or simply using the *verbatim* environment, which gives a Courier font to literal text:

```
for (int i = 0; i < n; i++)
    cout << "It's easy to add source code to LaTeX documents";
```

The listing environment is good for longer code examples and for use in figures, such as in Fig. **??**.

Another frequent need is to include algorithms written in pseudocode. The *algo* package can be used to format algorithms presentably in your documents; Fig. **??** on Pg. **??** shows an example of this. For more options on these packages, consult online resources.

Algorithm *CalculateMutationAdequacy*(T, P, M_o)
$(*$ Calculation of Strong Mutation Adequacy $*)$
Input: Test Suite T;
Program Under Test P;
Set of Mutation Operators; M_o
Output: Mutation Adequacy Score; $MS(P, T, M_o)$
1. $\mathcal{D} \leftarrow \mathcal{Z}_{n \times s}$
2. $\mathcal{E} \leftarrow \mathcal{Z}_s$
3. **for** $l \in$ *ComputeMutationLocations*(P)
4. **do** $\Phi_P \leftarrow$ *GenerateMutants*(l, P, M_o)
5. **for** $\phi_r \in \Phi_P$
6. **do for** $T_f \in \langle T_1, \ldots, T_e \rangle$
7. **do** $R_f^P \leftarrow$ *ExecuteTest*(T_f, P)
8. $R_f^{\phi_r} \leftarrow$ *ExecuteTest*(T_f, ϕ_r)
9. **if** $R_f^P \neq R_f^{\phi_r}$
10. **do** $\mathcal{D}[f][r] \leftarrow 1$
11. **else if** *IsEquivalentMutant*(P, ϕ_r)
12. **do** $\mathcal{E}[r] \leftarrow 1$
13. $D_{num} \leftarrow \sum_{r=1}^{s} pos(\sum_{f=1}^{n} \mathcal{D}[f][r])$
14. $E_{num} \leftarrow \sum_{r=1}^{s} \mathcal{E}[r]$
15. $MS(P, T, M_o) \leftarrow \frac{D_{num}}{(|\Phi_P| - E_{num})}$
16. **return** $MS(P, T, M_o)$

Figure 3: Algorithm for the Computation of Mutation Adequacy. This example pseudocode is by Gregory M. Kapfhammer.

5 Research and Writing Timetable

Give a brief overview of how you will proceed to accomplish the project, including a rough schedule for accomplishing the following tasks:

- background research,
- final proposal completion,
- proposal defense,
- research intermediate and final goals, and
- writing of key chapters, leading to final written thesis.

The timetable should take into account the actual schedule followed by the department, CS600 in the fall is devoted to the background research, final proposal and its defense,

beginnings of primary project work and writing of first two chapters. CS601 in the spring is spent finishing the research and the writing and preparing for the thesis defense.

The timetable section may also include contingency plans—see below in Sec. **??**.

6 Conclusion

Concluding remarks may discuss future research directions that will be made possible when the work succeeds, and possible places where the work might be cut short (due to difficulties) while still achieving some of the significant objectives. The latter might alternatively be discussed above in Sec. **??**.

The conclusion may also discuss future applications of and extensions to the thesis work. After completing this thesis proposal template, a logical next step is to create a template for the written thesis itself. Since writing a 50+ page document requires even more discipline and organization, this will entail creating a LaTeX style file, tentatively named `gatorthesis.sty`, which will be called by a `\usepackage{gatorthesis}` command. This will provide automatic formatting for title pages, abstract and acknowledgement pages, Tables of Contents and Figures, chapter headings, and so forth. For an experienced LaTeX user, with a BibTeX database already assembled for the proposal, this will make thesis writing go a lot smoother.

This paper has shown only one of many different ways you might structure your project proposal. Individual proposals will vary depending on the nature of the project, but most of them will have the same essential components. You will have many other occasions to write proposals—whether for graduate projects and theses, grant proposals, or industry-related projects. The important thing is to express your ideas in a concise and effective fashion, so that a reader is neither confused nor bored nor irritated. *Too long and wordy* is as bad as *too short and lacking in detail*; grammatical and spelling errors are likewise unacceptable. Write fast, rewrite thoroughly, and proofread religiously.