Super Outlining

John C. Lusth

Revision Date: August 8, 2012

Introduction

Outlining, like flowcharts for programming, is becoming a lost art. Unlike flowcharts, however, outlining has not been replaced by superior approaches. Through proper outlining, a document essentially writes itself, resulting in a concise, logical, and compelling narrative.

The method described herein was invented independently by me, but I have found subsequently that bits and pieces of my approach have been suggested before. My take was that writing a paper ought to be much like writing a program using *stepwise refinement*[1]. With stepwise refinement, one starts out by defining a *main* function; the *main* simply calls a series of subfunctions which do not yet exist. The *step* in stepwise refinement is to define these subfunctions, which, in turn, may call other functions that may or may not yet exist. The process ends when all functions are defined.

The *main* function of a program looks much like the outline of a paper, with the sections of the paper corresponding to the subfunctions the *main* function calls. Thus, it should be possible to apply the technique of stepwise refinement to writing a paper, should one start with an outline. When stepwise refinement is applied to an outline, each section heading of the outline is expanded to a short description. The description is then expanded to a series of paragraph topics. The topics are, in turn, expanded to partial paragraphs. Finally, the partial paragraphs are expanded to complete paragraphs. I call the combination of outlining and stepwise refinement, *super outlining*. Because of the individual steps taken during the process, super outlining leads to coherent documents that flow well.

Phase I

The initial phase of the super-outlining method is to map out the overall structure of the document, down to the subsection level. For a research paper documenting an experiment, the initial outline might look like this:

- Introduction
- Background
- Experimental Procedures
- Results and Analysis
- Conclusions

Of course, the section names will vary from discipline to discipline and whether the paper is experimental, theoretical, or descriptive. For example, a paper performing a literature review, which is essentially a *Background* section expanded into a full paper, might have the following top-level sections:

[1]The original idea, by Niklaus Wirth, can be read here: http://sunnyday.mit.edu/16.355/wirth-refinement.html

- Introduction
- Early work
- Seminal discoveries
- Refinements
- Remaining Challenges

Many times, a section is involved enough that subsections are warranted. For example, an *Experimental Procedures* section might be further divided:

- Experimental Procedures
  - Design
  - Fabrication
  - Test

Typically, a traditional outline, as shown in the examples above, is as far as most people get in the outlining process. Unfortunately, while this level of outlining guides the overall structure of the document, it gives no insight on how the individual sections and subsections might be written. So to address this problem, each section/subsection heading is annotated with a three-sentence description. The prose of these paragraphs will not generally appear in the final document, so it can be casual or instruction-based. For example, the paragraph describing the *Introduction* might look like:

- Introduction

  I'll start by stating that inheritance becomes much more complex within the context of non-local references, supplying references from Touretzky, Ungar/Smith, and some other references concerning the D programming language. I'm going to use the examples of Meyer and Taivalsaari to motivate the need for solving the problem. Then I'm going to give a hint about a possible inheritance approach using scope, rather than self-reference, in an attempt to simply and clarify the semantics of inheritance.

The example here uses ideas from Computer Science, but don't let that get in the way of this guiding principle that applies to all documents: each section is annotated with a three-sentence paragraph describing the ideas that will be expressed within the section.

A big draw of super-outlining depends on the theory that spending more time up front organizing your thoughts will reduce the overall time writing the full paper. By writing the section descriptions, it then becomes a simple matter to expand the outline further in Phase II.

Phase II

In Phase II, one envisions the paragraphs needed to flesh out the section descriptions. After each description, one places a list of one-sentence paragraph topics; if you will need five paragraphs to succinctly implement the section, then you should place a list of five sentences following the section description. Deciding on the number and topics of the paragraphs is usually a back-and-forth process, with the number of paragraphs growing and shrinking as you decide on the paragraph topics. Be sure to reserve the last paragraph in a section as an introduction to the next section, if appropriate.

For example, a paragraph topic list for *Introduction* might look like:

- Introduction

  I'll start by stating that inheritance becomes much more complex within the context of non-local references, supplying references from Touretzky, Ungar/Smith, and some other references concerning the D programming language. I'm going to use the examples of Meyer and Taivalsaari to motivate the need for solving the problem. Then I'm going to give a hint about a possible inheritance approach using scope, rather than self-reference, in an attempt to simply and clarify the semantics of inheritance.

  1. What is inheritance and what are the formal approaches to describing inheritance?
  2. How are non-local references in an ancestor class handled under the formal descriptions?
  3. Should non-locals be handled differently and, if so, what are the advantages and disadvantages of doing so?
  4. Posit the theory that manipulating scope is simpler than self-reference and handles non-local references correctly.
  5. Describe the rest of the paper, indicating the existence and purpose of each of the following sections.

Although you may not know much about CS inheritance, you can see from the example the beginnings of flow. Notice how the topic sentences lead from the description of the problem to its solution.

During Phase II, it is common to return to Phase I and modify the section descriptions. In fact, it would be rather unusual *not* to do so.

Phase III

In Phase III, each paragraph topic is augmented with the introductory and closing sentences of the paragraph. This will further cement the 'flow' of the document; care should be taken that the closing sentence of a paragraph should serve as a transition to the next paragraph, if appropriate. Indeed, after adding these bracketing sentences, one should be able to read the entire document as a coherent whole.

As an example, a Phase III 'paragraph' might look like this:

> If an object truly inherits the *functionality* of its ancestor, then non-local references in ancestor methods should reflect the point of view of the actual object, not the view of an ancestor object which does not even exist. *...Should non-locals be handled differently and, if so, what are the advantages and disadvantages of doing so?...* The traditional handling of non-locals likely stems from an ease-of-implementation issue; is there an equally simple implementation that handles non-locals properly?

Writing the paper

At this point, the full paper can be attempted. While filling out the middles of the partial paragraphs of Phase III, one should move each paragraph description to precede the paragraph it describes. One should also set off the section descriptions and the moved paragraph descriptions in a different font. For the final edit, the section and paragraph descriptions are not deleted, but set off as non-printing notes within the document.

Other considerations

Any document that is intended to be widely read should tell a story. That is to say, an *Abstract*, if required, should give a synopsis of the entire story, while the *Introduction* should set up, in a compelling way, the

problem that needs to be solved. Concrete examples of how the problem manifests itself should also be included in the *Introduction*. The *Background*, could give a chronological history of how the problem arose, if appropriate. The middle sections describe the meat-and-potatos version of the problem solution. Finally, the *Conclusions* section should recap the problem, its solution, and the strength and weaknesses of the solution.

Another rule the successful writer should use, if possible, is the *Rule of Three*. The *Rule of Three* states that any complex idea, thought, or approach, should be broken down into three components. If any of the components is still too complex to be described succinctly, it too should be broken down into three subcomponents, and so. Resist the attempt to use four or more subcomponents and *never* use two. The *Rule of Three* also states that the most important aspects of the document, typically the problem and its solution, should be repeated three times within the document. Given that the *Abstract* and the *Conclusions* both summarize the work described in the document, the desired repetition of the key points should happen naturally.

Lastly, in the final editing of the document, a critical eye is placed on each paragraph, each sentence and on each word of the document. Each item should only be present if it contributes in an original way to the overall story told by the document; otherwise, it should be removed. Typically, via super-outlining, each paragraph is serving a purpose, but often sentences repeat information or are awkward and inelegant. Questions to ask are:

- Does this sentence repeat information that has already been stated?
- If not, is the sentence absolutely necessary for purpose of the paragraph?
- Is each word absolutely necessary for the purpose of the sentence? In other words, can the sentence be shortened but still retain its meaning?

By looking critically at the word and sentence level, a document can both be shortened and become more elegantly written.