Sample Structure for Final Project Report

ECE 695 –Fault-Tolerant Computer System Design
School of Electrical and Computer Engineering
Purdue University
Fall 2013

This document lays out the suggested format and structure of the final project report. The format is strongly suggested, and the structure is only suggested, which means you can use any other structure that suits your specific project better.

For the format, the simple rules are:

1. 11 pt font
2. Double line spacing
3. 1” margin on all sides
4. Page numbers on each page

For those using Latex for creating the report, you can find the IEEE templates at the following URL:

http://www.ieee.org/conferences_events/conferences/publishing/templates.html

Use the IEEEtran class file and the following options: 11pt, draftclsnofoot, onecolumn.

Thus, at the beginning of your Latex file, you will have:

\documentclass[11pt, draftclsnofoot, onecolumn]{IEEEtran}

Now, here is the structure of the report. The page numbers may vary significantly from what I have in here, so these you can easily disregard.

1 Introduction [2-3 pages]

1.1 What is the problem?

1.2 How has it been solved till now?

1.3 What was your main solution idea?

1.4 What are the key technical details of your solution?

1.5 How did you evaluate your solution (2-3 key results)?

1.6 A high level figure of your solution, or evaluation method

1.7 A list of contributions (3-4) that you can claim from this work.

2 Background [2-3 pages]

2.1 Provide a background of the problem, in easy-to-understand terms. This should not be tied to your solution. Here you can provide some context about the problem – why it is important, where it is used, etc.

2.2 Give a high level view of the different approaches used to solve the problem till now. Within each approach, there may be multiple papers that fall within the same approach.

3 Design Overview [1-2 pages]

3.1 High-level conceptual figure of how your solution works

3.2 Workflow of how your solution works – the detailed pieces will come in the next section – you can give forward pointers to the details

3.3 What kinds of failures or attacks is your solution meant to handle

3.4 One or two common use cases – how an end user will use your solution

4 Solution Details [Longest single section]

Here you describe the detailed techniques in your solution. For each part of the solution, put it in the context of the overall system or solution – where does it fit, what is its functionality. Do not just give pseudo code, but explain in words what is the design behind the technique. If there are alternate ways of doing this, describe them and say why one is better than the others. If your technique expands some prior technique, refer to that, and point out the addition that you have done.

5 Implementation [As much as you need]

What language, IT infrastructure? What are the pragmatic trade-offs that you had to make? What is the complexity of the implementation – LOC, other metrics? What are the dependencies of your implementation?

6 Experiments and Results [As much as you need]

For each result, explain: what is the goal of the experiment, what you did, then comes the plot, then interpret the plot. Try and have some comparative result, with prior work.

7 Discussion

Here talk of things that your solution does not address straight away, but can be tweaked to handle. Point out weaknesses of your solution and how you would address them.

8 Conclusion and Future Work

Summarize the main contributions of the work and what further work someone should do to make the solution better.