3 Design

Design is quite important for any application. Thoughtful consideration turns out a better program, and a better product. Sadly enough, I was under time pressure and had to develop my document class in parallel with my second work report. This could have led to some crippling architecture flaws, but fortunately it seems that the system has evaded them.

3.1 LaTeX as a platform

Although I was quite familiar with the WordPerfect word processor, I became quite convinced that it would not fare well as a basis for the macro system after an initial prototype. Although the user interface was optimised for entering text, it was also optimised for adjusting the formatting of the text. This rubbed against the grain of my requirements, which would be to remove the need for hand-formatting.

I had been introduced to LaTeX by Prof. Victor Quintana, and came to the conclusion that it may be appropriate. It is a **mark-up language**, such that text has commands embedded within which commands the computer to perform a certain tasks. For instance, the words ``mark-up language'' were boldfaced with the following command: `\textbf{mark-up language}`. The `\textbf` command indicated that boldface text should be used, and the braces contained the text to be boldfaced. LaTeX is a complete programming language for manipulating documents, so it seemed like a good choice for implementing a set of formatting rules, such as work report guidelines.

The decision to use LaTeX was one of the best, and worst design choices I made. It was a great decision since the platform was well-defined and stable. TeX has been stable since 1995, MetaFont has been stable since 1998 and LaTeX has been stable since 2001. LaTeX is a well-respected system used throughout academia, and there was a plethora of resources available. As well, the syntax was relatively clean, and it facilitated documents that expressed the semantics as opposed to the raw typesetting.

However, it has one major disadvantage: LaTeX is not a word processor. Word processors are transparent to use, when the author types text, it appears much like it would in the final output. A text-based programming language is used to instruct LaTeX, which means that the author does not receive immediate feedback on the text input. In addition, syntax errors are inevitably made in LaTeX input, something that does not occur with word processors.

Still, the tradeoff for rapid development over ease-of-use has been rather profitable. See Section 7 for feedback