

MathML Formatting with \TeX Rules and \TeX Fonts

Luca Padovani
University of Bologna
lpadovan@cs.unibo.it

March 11, 2003

Abstract The Mathematical Markup Language (MathML, <http://www.w3.org/Math>) is an XML application for the encoding of mathematical expressions. Its presentational fragment allows the encoding of expressions for rendering purposes and in this respect it resembles \TeX and \LaTeX more familiar commands and macros for math typesetting. Being designed for publishing mathematics on the Web, MathML markup does not make any assumption on the graphic capabilities of the environment in which it is rendered: its formatting semantics are necessarily specified at a rather high level, leaving the details to individual implementations. This is exactly the opposite of what happens in \TeX , which has very precise, basically fixed formatting rules, and is guaranteed to achieve excellent results.

Computer Modern fonts, as well as other fonts designed for use with \TeX , provide high-quality glyphs for many mathematical symbols. As such, they are appealing candidates to be used for MathML formatting, especially when a formatting quality comparable to that of \TeX is desired. The problem is that these fonts have been designed with the explicit purpose of being used in conjunction with \TeX : they provide a number of *parameters* affecting the way \TeX uses and places the glyphs; at the same time, \TeX makes *assumptions* on the glyphs and their metrics, and on the arrangement and relationships of the glyphs in those fonts (for instance, the existence of a `skewchar` for the correct positioning of accents).

Because \TeX and \TeX fonts are so mutually dependent, designing a formatting engine for MathML that uses \TeX fonts can eventually tie the process of formatting to \TeX formatting rules. This is happening, for instance, in the Mozilla (www.mozilla.org) Web browser. For a MathML formatting engine this constitutes a severe drawback, for it limits the portability of the engine and prevents its use with differently designed fonts. In the paper we investigate the architecture of a formatting engine for MathML based on an abstraction of \TeX box primitives which is capable of achieving a \TeX -comparable formatting quality when used in conjunction with \TeX fonts. The engine is carefully designed so that the font-dependent formatting rules are isolated from the context independent ones. The use of a model based on \TeX -like boxes is crucial in the separation of the two modules. We also show how the formatting rules presented in Appendix G of the \TeX book can be easily adapted for MathML formatting, and how the semantically-rich MathML markup simplifies the rules themselves.

The outcome of the project is not just a software engineering exercise. We clarify the relationship among \TeX , \LaTeX , and MathML, and we give an effective way for broadening the basin of potential users of \TeX fonts. There is the somehow widespread belief that MathML and \TeX are competing technologies. We show that this is not always the case, as \TeX , \TeX fonts, and MathML can fruitfully coexist in the years to come.