MathBook XML

Robert A. Beezer

Abstract

MathBook XML is an XML application to describe the structure of a technical document, such as a mathematics research article or textbook. This application is designed so an author can easily and clearly separate presentation from content. This then allows for direct XSL conversion to many formats, such as print, PDF, HTML, and EPUB.

LaTeX syntax is used to represent mathematics and clean LaTeX source is the result of one of the conversions, which is the precursor to print and PDF outputs. HTML output allows for variable granularity of the resulting pages and provides presentation and navigation interfaces appropriate for both small and large screens.

1 Introduction

There is a need for a source format that explicitly expresses the structure of a technical document, while making no assumptions whatsoever about presentation. This format should enhance an author's ability to specify this structure and free the author from eventual decisions about presentation. With such a format it is possible to then create versions for readers in a variety of formats, such as print, PDF, HTML, and EPUB, in addition to new formats not yet imagined.

2 The case for XML

eXtensible Markup Language (XML) is an extremely simple specification, with few reserved characters and a handful of rules about syntax. It suffers from a reputation of being verbose and overly-complicated. But this is primarily an artifact of its employment as a data-interchange language written by programs, not people. It is possible to design an XML application (the set of elements and attributes) which is natural for authoring and editing. MathBook XML is a case in point.

Technical documents such as mathematics research papers and mathematics textbooks are very structured. Chapters break into sections and sections may break into subsections, all with a hierarchical numbering scheme. There are numbered definitions, theorems, lemmas and corollaries. Figures, tables and references are all numbered. Then there are extensive cross-references to these items. XML naturally allows for an easy specification of the tree-like structure of such a document and cross-referencing is well-supported.

The eXtensible Stylesheet Language (XSL) provides a powerful declarative language for transforming XML into text, HTML or XML formats. In our case, it is possible to write out LaTeX source (resulting in PDF or print) and to write out HTML (resulting in web pages or as the basis of other formats such as EPUB). So the ease of authoring in XML with a carefully designed set of elements, along with the transformational power of XSL, make an XML application a natural choice.

3 MathBook XML source format

The elements in MathBook XML will feel natural to an author. To begin with, `<chapter>`, `<section>`, `<title>`, `<definition>`, `<theorem>`, and `<example>` are all exactly what you would expect them to be. At the paragraph level, frequently-used items have short abbreviations, familiar to anyone who knows HTML: `<p>`, `<em>`, `<q>`.

LaTeX includes at least four systems for cross-references: theorems (and similar environments), references, figures and tables, and equations. In MathBook XML, there is only one system, since a cross-reference is able to inspect the *type* of object it is pointing to and format the pointer in the way a reader expects (bare number, in brackets, or in parentheses). Optionally, you can choose to have prefixes on cross-references, such as "Theorem" or "Corollary" added automatically, with support for different languages, similar to what the `cleveref` package does.

3.1 Mathematics

While one of the principal intents of MathBook XML is semantic markup, we have not chosen to go into this rathole for mathematics. LaTeX syntax works very well between the dollar signs and is understood by many authors. Indeed, it is the MathJax JavaScript library that makes this project possible. MathJax is able to do a very good job of expressing a wide range of LaTeX syntax (e.g. the `amsmath` package) in a web browser. So we can have quality output of displayed equations in both print/PDF and in HTML. MathBook XML is configured so an author need only specify macros once, and they can be employed in both the LaTeX and HTML output. Furthermore, they can be employed in the source code for diagrams described by the Ti*k*Z and Asymptote languages.

Inline math, single displayed equations (numbered or not) and multiline displays (numbered lines or not) are all supported. As an example of how authoring in XML is different, it is worth noting that in a multiline display, each line needs to be delimited by an XML element. This then supports switching numbering on and off on a per-line basis, in addition to marking individual lines as targets of cross-references, all in a manner consistent with the syntax used for cross-references elsewhere in the document.

3.2 Graphics

For documents tracked by version control, or published with open licenses, or where the consistency of notation is controlled by simple macros, it is desirable to specify diagrams with a graphics language, rather than employing opaque raster formats. MathBook XML supports diagrams specified in Ti*k*Z, Asymptote or Sage, with full support for macros in the first two (and may be supported soon within Sage graphics). Diagrams are specified directly in the source, and then it is easy with XSL to isolate each one, in order to wrap it in the appropriate syntax so that it can be transformed to a standalone image. This step, and the subsequent processing, are all controlled by a single Python script, `mbx`. Whenever possible, the result is in Scalable Vector Graphics (SVG) format, so for example, the image scales fluidly on a web page when the reader zooms in or out.

3.3 Widgets on the web

There is a wide variety of JavaScript and Java applets which may be embedded on a web page. For example, GeoGebra is a sort of playground for exploring concepts and results from Euclidean geometry. It can provide a prepared interactive demonstration or a blank canvas with tools. One of our primary motivations has been to support the inclusion of the Sage Cell Server to allow readers the use of this extensive open source system for mathematics directly in a textbook. The Sage Cell is a text area, typically pre-loaded with Sage code from the author, which can be evaluated by pressing a button that sends the code to a publicly available server that then returns the results back into the reader's page. The reader is then generally free to edit the code to examine new situations, learning new mathematics or new Sage commands in the process. WeBWorK homework exercises are another conceivable addition for textbooks.

But a question remains: what to do with a dynamic element (such as a GeoGebra demonstration) in an output format that is inherently static, such as print. PDF readers can generally accommodate hyperlinks, but typically it is only Adobe's own Acrobat Reader that can display more complicated items such as animations.

4 LaTeX output

A principal feature of the LaTeX output from MathBook XML is the isolation of style parameters in the preamble, and a body that is generic enough that it can be styled differently through style files or adjustments to the preamble. Parameters affecting style, such as margins, font size, and numbering depth can be controlled by parameters to the XSL processor and are not part of the MathBook XML specification of the source. With limited exceptions, the numbering of items is accomplished in the usual way, by letting the LaTeX processing do that work. In other words, numbers of theorems, sections, figures, are not hard-coded into the LaTeX created by the XSL transformation.

5 HTML output

HTML output from MathBook XML source allows for the creation of web pages at variable granularity ("chunks"). So a book can be broken up into one web page per subsection, if desired. Then CSS and JavaScript are employed to provide natural navigation interface elements, such as a clickable Table of Contents, plus "Previous", "Up" and "Next" buttons. On smaller screens (such as phones) the interface elements adjust to the limited space. The HTML output is also meant to be divorced from presentation, though the limited capabilities of CSS mean more information must be hard-coded, such as hierarchical numbering. But there is a good separation between the HTML and the CSS, which allows for different presentations.

The various numbering schemes that result from LaTeX processing are identical to the numbering hard-coded into the HTML. For example, equation numbers in multi-line mathematics displays are identical in a PDF created via LaTeX to the numbers that appear in a web page version rendered by MathJax. References to equations look and behave the same for both types of output.

6 Other output formats

Sage worksheets, iPython notebooks and Sage Math Cloud worksheets are all formats for web applications that allow a user to execute commands relevant to scientific computing, while maintaining a written record of input and output, with rich tools for annotating the results. Underneath, these documents are primarily HTML, adorned with CSS or JSON. We have experience with conversions of MathBook XML source to each of these formats, and have a usable conversion to Sage worksheets publicly available. With a new MathJax tool for rendering mathematics as embedded SVG images, it appears that it is now possible to manufacture EPUB output that is as capable as PDF.

7 Philosophy

The primary purpose of MathBook XML is to make it easy for authors to capture their writing in a structured format. Then processing tools can be applied to create various output formats, while also insulating the author from a variety of technical details. While the supplied XSL conversions are meant to be usable and of high-quality, they are primarily a demonstration of the utility of the XML specification of the source. The conversions are modular enough that others can create new conversions by making small additions or changes, or by starting at a lower level and converting to some entirely different format.

8 Development

Code for MathBook XML is open-source (GPL license) and lives in a GitHub repository [1]. The latest changes are on the `dev` branch. There is a sample article in the examples directory of the distribution, which is heavily annotated and tries to contain at least one example of every feature possible, and so serves as accurate documentation. Support questions and feature requests take place on a Google Groups forum. Pointers to these resources, and others, can be found at the website [2]. The Gallery at the website contains links to large examples of MathBook XML in production use.

9 Acknowledgements

This is joint work with David Farmer, Steve Blood, and Michael DuBois. Financial support has come from the UTMOST project (National Science Foundation DUE-1022574), a Shuttleworth Foundation Flash Grant, and the University of Puget Sound.

References

[1] Robert A. Beezer. GitHub repository: `rbeezer/mathbook`. Available at `https://github.com/rbeezer/mathbook`.

[2] Robert A. Beezer. MathBook XML. Available at `http://mathbook.pugetsound.edu`.

◇ Robert A. Beezer
Dept. of Mathematics and Computer Science
University of Puget Sound
Tacoma, Washington, USA
beezer (at) pugetsound dot edu
http://buzzard.pugetsound.edu