BookML: automated LaTeX to bookdown-style HTML and SCORM, powered by LaTeXML

Vincenzo Mantova

15th May 2023

Abstract

BookML is a small wrapper around LaTeXML for the production of accessible HTML content straight from LaTeX files, and for packaging it as SCORM. Created by and maintained for maths lecturers at the University of Leeds.

Sources available on GitHub.

Formats: GitBook (html), plain with Latin Modern (html), PDF, LaTeX source.

For a more beginner friendly guide see the Leeds BookML guide.

Contents

1 Getting started

1.1 Prerequisites

- LaTeXML (minimum 0.8.5, recommended 0.8.6 or later)
- for any image handling: the Perl module `Image::Magick`
- for handling EPS, PDF images: Ghostscript
- for BookML images (for Ti*k*Z and similar packages): Ghostscript, latexmk, `preview.sty`, dvisvgm (minimum 1.6, recommended 2.7 or later)
- for automatic PDF, HTML, zip, SCORM packaging: GNU make, latexmk, zip, optionally texfot

1.2 Installing BookML

1. Install the prerequisites.
2. **Install/upgrade:** unpack the latest BookML release and put the `bookml` folder next to your `.tex` files.
3. **First install only:** copy `bookml/GNUmakefile` next to your `.tex` files.

Or you can unpack the template to start with a working minimal example.

The Leeds BookML guide has further examples and tips for lecturers and detailed installation instructions (some specific to the University of Leeds), including for instance how to compile exercises with and without solutions, or how to produce various alternative PDFs from the same file.

1.3 Compilation

Run `make` in the folder containing `GNUmakefile` and the `.tex` files. Each `.tex` file containing the string `\documentclass` will be compiled to PDF, HTML, then a zip package and a SCORM package.

If compilation does not succeed, you may have to run `make` a second time. If it still fails, run `make clean-aux` or delete the `auxdir` folder to reset the state.

2 Options

The `bookml` package accepts a few options (for instance use `\usepackage[nomathjax]{bookml/bookml}` to avoid including MathJax). The options have no effect on the PDF output.

style=gitbook Use the GitBook style (the default behaviour). When using the GitBook style, you must call `latexmlc` (or `latexmlpost`) with the option `--navigationtoc=context`. Any PDF or EPUB file with the same name as the source will be detected and added to the download menu.

style=plain Use the LaTeXML style with a few slightly opinionated tweaks.

style=none Use the LaTeXML style with no tweaks (except for backported styles and some fixes).

nomathjax Do not include MathJax in the output.

mathjax=2 Use MathJax version 2 instead of version 3.

imagescale=X.XXX Rescale the images generated via LaTeX (§ 4.5) by the desired factor. The scaling factor is adjusted internally based on the options `8pt`, `9pt`, ..., `12pt` being passed to the document class.

nohtmlsyntax Disable the HTML syntax that makes `\<` into the BookML command for introducing HTML tags. Required if you already define `\<` for something else.

3 Customisation

3.1 css and fonts

The CSS files in the folder `bmluser`, if one exists, are automatically included at the end of the `<head>` tag and will override the previous styles.

If the file name ends with `.style1,style2.css`, then that file will be used only when `style=style1` or `style=style2` is passed. You can use `._all.css` to ensure that the file is included in every style.

The plain version of this manual has been compiled with `latin-modern.plain.css` that sets the font to Latin Modern.

3.2 html output

You can copy the file `bookml/LaTeXML-html5.xsl` one level up and modify it to customise the HTML output. Knowledge of XSLT is required!

3.3 Splitting

By default, files are split into separate pages for each section. You can change this by running `make SPLITAT=chapter` or `make SPLITAT=`.

To make the change permanent, add the line `SPLITAT=chapter` to `GNUmakefile`. You can also specify different values per file, for instance `docs/index.html: SPLITAT=`.

The `latexml` command line can be further customize by setting the variables `LATEXMLEXTRAFLAGS` and `LATEXMLPOSTEXTRAFLAGS`. See the file `bookml/bookml.mk` for further options.

4 Commands

4.1 Conditional execution

Call `\iflatexml ... \else ... \fi` to write code that is executed only by LaTeXML, or only (pdf)LaTeX respectively. `bookml` will try to use the `latexml` package, if available, or use its own embedded copy. See Figure 1.

```
\caption{Example of
  \iflatexml\ltxinline|\xymatrix|\else\ltxinline|\\xymatrix|\fi{}
  from the \ltxinline|xypic| documentation.}
```

Figure 1: Example of `\iflatexml`, used in Figure 4 to work around a subtle difference between LaTeXML and LaTeX.

4.2 Alternative text for images

Call `\bmlDescription{textual description}` *right after* an image to populate its `alt` attribute (or `aria-label` if appropriate). Inspect the HTML source of Figure 3 or use a screen reader to check its text description.

LaTeXML version 0.8.7 and later support the new LaTeX syntax which (eventually) will also embed the alternative text in the PDF output: `\includegraphics[alt={textual description}]`.

4.3 Disable MathJax for some equations

Call `\bmlDisableMathJax{}` inside an equation to stop MathJax from processing the equation. Useful if you want to use MathJax, but you have equations that are better handled by the browser. When used inside an environment creating multiple equations, it applies only to the ones containing the command. See Figure 2.

```
\begin{align*}
  \int_0^{+\infty} x^2 \mathrm{d}x \quad \text{rendered by \href{https://www.mathjax.org
  \bmlDisableMathJax \int_0^{+\infty} x^2 \mathrm{d}x \quad \text{rendered by \emph{the
\end{align*}
```

$$\int_0^{+\infty} x^2 \mathrm{d}x \quad \text{rendered by MathJax}$$
$$\int_0^{+\infty} x^2 \mathrm{d}x \quad \text{rendered by } \textit{the browser}$$

Figure 2: How to disable MathJax for a single equation.

4.4 Add custom CSS classes

Call `\bmlPlusClass{class}` *right after* some piece of content to add a CSS class. If done within text, its effect may be unpredictable. Its main use is to call `\bmlPlusClass{bml_no_invert}` after an image to prevent the picture from getting inverted in night mode. Compare how Figure 3 (with `bml_no_invert`) and Figure 4 (no additional classes) change in night mode to see the difference.

Note that the package `latexml` also offers `\lxAddClass` and `\lxWithClass` for the same effect but different behaviour regarding which element gets the class.

4.5 Generate pictures with LaTeX

LaTeXML supports the `picture` environment as well as *some* Ti*k*Z pictures and some XY-matrices. For various reasons, the output is usually mangled, and compilation times blow up. Some common packages are not supported altogether (for instance `tikzcd`, `animate`).

BookML offers a simple automated way of generating SVG images using LaTeX, bypassing LaTeXML entirely. In your preamble, after `\usepackage{bookml/bookml}`, write

```
\bmlImageEnvironment{tikzpicture,tikzcd}
\bmlImageEnvironment{animateinline}

% optional, but strongly recommended:
% do not load tikz when running in LaTeXML
\iflatexml\else
\usepackage{tikz}
\usetikzlibrary{cd}
\usepackage{animate}
\fi
```

All environments passed to `\bmlImageEnvironment`, in this case `tikzpicture`, `tikzcd`, `animate`, will be compiled with LaTeX using `latexmk` and converted to SVG images via `dvisvgm`. Note that in this example LaTeXML will not load Ti*k*Z at all. Figure 3 demonstrates this approach.

If you only need this mechanism in a pinch, you can simply wrap the desired content between `\begin{bmlimage}` and `\end{bmlimage}` as exemplified in Figure 4.

4.6 Alternative formats

`\bmlAltFormat{docs.large.pdf}{PDF (large print)}` instructs BookML to compile `docs.large.tex` to PDF and include the result in the HTML output. The file will appear in the 'Downloads' menu of the GitBook style.

Please note that `docs.large.tex` must *not* contain the string `\documentclass` or it will be compiled by itself into HTML, zip, and SCORM.

See `template.tex`, `template-sans.tex`, `template-sans-large.tex` for an example.

4.7 Direct html input

You can insert arbitrary HTML code using `\bmlRawHTML{html code}`.

Warning: the HTML code needs to be written in 'XML syntax', so you have to close all the tags (for instance, write `<br/>` instead of `<br>`, close the `<p>` tags, and so on) and empty attributes

```
\begin{animateinline}[
  alttext=none,loop,controls,nomouse,poster=20,autopause,
  begin={\begin{tikzpicture}
    \useasboundingbox (0,0) rectangle (5,3);},
    end={\end{tikzpicture}}]{30}
  \multiframe{160}{dShift=50mm+-0.5mm}
    {\duck[tophat,xshift=\dShift]}
\end{animateinline}
\bmlDescription{A stylised rubber duck, yellow
  and wearing a black top hat, enters from the right
  and slides until it exits from the left. The animation
  repeats every six seconds.}
\bmlPlusClass{bml_no_invert} % preserve colours in night mode
```

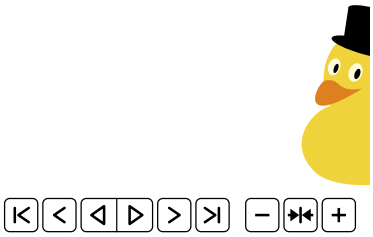

Figure 3: A fancy duck. Click on the play button to start the animation (for the PDF, it requires a compatible software such as Acrobat Reader).

```
\[ \begin{bmlimage}
  \xymatrix{
    U \ar@/_/[ddr]_y \ar@/^/[drr]^x \ar@{.>}[dr]|-{(x,y)} \\
    & X \times_Z Y \ar[d]^q \ar[r]_p & X \ar[d]_f \\
    & Y \ar[r]^g & Z}
  \end{bmlimage} \]
```

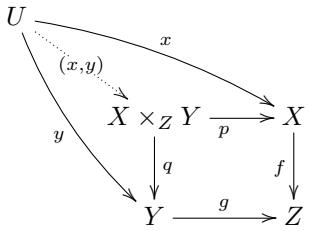

Figure 4: Example of `\xymatrix` from the `xypic` documentation.

```
\newcommand{\youtube}[2]{\bmlRawHTML{
  <div style="max-width: 1920px; width: 100\%">
    <div style="position: relative;
        padding-bottom: 56.25\%; height: 0; overflow: hidden;">
      <iframe width="1920" height="1080"
        src="https://www.youtube-nocookie.com/embed/#1"
        title="YouTube: #2" allowfullscreen=""
        style="border:none; position: absolute; top: 0; left: 0;
          right: 0; bottom: 0; height: 100\%; max-width: 100\%;"
        allow="accelerometer; autoplay; clipboard-write;
          encrypted-media; gyroscope; picture-in-picture"/>
    </div>
  </div>}
\iflatexml\else
\begin{center}
  Watch \href{https://www.youtube.com/watch?v=#1}{#2}.
\end{center}
\fi}
\youtube{mH0oCDa74tE}
  {Group theory, abstraction, and the 196,883-dimensional monster}
```

Watch Group theory, abstraction, and the 196,883-dimensional monster.

Figure 5: Demonstration of `\bmlRawHTML` within `\newcommand` with a video from 3Blue1Brown.

must be given the value `""` (see this old W3C guide for some indications). Moreover, you must remember to escape your `%&_^${}`, and replace `\` with `\textbackslash`.

`\bmlRawHTML` is robust, i.e. it does not change the category codes, so it can be used inside `\newcommand` to create custom macros. See for instance Figure 5 for a generic YouTube embedding macro. Note that the video will not be visible in the PDF, so a link should always be provided (possibly PDF only, as in the example).

4.8 Interspersing LaTeX and html (beta)

You may write arbitrary HTML using the syntax `\<html-tag>`. Just as for `\bmlRawHTML`, you need to use the XML syntax.

If you need the command sequence `\<` for other purposes, load the BookML package with the option `nohtmlsyntax`. You may still intersperse the HTML tags by declaring them first with `\bmlHTMLEnvironment{tag}`, and then using `\begin{h:tag} ... \end{h:tag}`. Attributes can be passed as optional arguments `\begin{h:tag}[attr1=val1,attr2=val2]`. You can specify multiple tags by separating them with commas, as in `\bmlHTMLEnvironment{tag1,tag2}`.

Use `\bmlHTMLInlineEnvironment{tag}` for tags that can only contain 'phrasing' content, for instance they should not contain `<p>` paragraphs.

See figures 6, 7 for an example in both styles.

Code for `<details>`.
Completing the quine is left as an exercise for the reader.

```
\<details style="text-align: left; width: 100\%" open="">
  \<summary>
    \textbf{Code for \lstinline[language=html,frame=none]|<details>|.}
  \</summary>

  Completing the quine is left as an exercise for the reader.
\</details>
```

Figure 6: Implementation of the `<details>` tag.

Code for `<details>`.
Completing the quine is left as an exercise for the reader.

```
\bmlHTMLEnvironment{details}
\bmlHTMLInlineEnvironment{summary}
\begin{h:details}[style={text-align: left; width: 100\%},open]
  \begin{h:summary}
    \textbf{Code for \lstinline[language=html,frame=none]|<details>|.}
  \end{h:summary}

  Completing the quine is left as an exercise for the reader.
\end{h:details}
```

Figure 7: Implementation of the `<details>` tag if using `nohtmlsyntax`.

4.9 ...and everything from `latexml.sty`

By using `\usepackage{bookml/bookml}`, you will also import `latexml.sty`, making several LaTeXML-related commands available, for instance `\lxBeginTableHead` for marking table headers. Please read the source of `latexml.sty` to learn what is included.